Virtual University	CS411-Visual Basics (Solved Macq's) LECTURE FROM (22 to 45)	VU LMS HELP vulmshelp.com
Junaidfazal08@gmail.com Bc190202640@vu.edu.pk	FOR MORE VISIT VULMSHELP.COME	JUNAID MALIK 0304-1659294

AL-JUNAID TECH INSTITUTE

Question # 1: In Xaml browser application, how much isolated memory we can use?

- 256 kb
- 128 kb
- **512 kb PG # 132**
- 1024 kb

Question # 2: Which property will be used if we want to restrict the user to don"t increase thewidth of a WPF element from a specific amount?

- **MaxWidth Page # 81**
- MaximumWidth
- WidthMaximum
- None of given options

Question # 3: Which one is the correct syntax for declaring attribute?

- **[Obsolete] page # 42**
- {Obsolete}
- (Obsolete)
- *Obsolete

Question # 4: In C#, comments are written using_____.

- # and */ /*
- **// and /* */ page # 14**
- # and /*
- // and */ /*

Question # 5: An event that is raised whenever the value of "CanExecute" changes is knownas_____.

- CanExecute

- **CanExecuteChanged**
- Command
- Execute

Question # 6:_____is an event that is introduced into an event processing system byan event producer.

- Event Producer
- Event Consumer
- **Raw Event Page # 11**
- Event Stream

Question # 7: Whenever an attribute value is enclosed in curly braces "{}", the XAMLcompiler/parser treats it as a/an__rather than a/an_____.

- Markup event, property
- Markup property, Event
- Literal string, Markup extension value
- **Markup extension value, literal string page # 65**

Question # 8: HTML is called_____layer.

- Physical
- Presentation
- Behavioral
- **Structural PG # 195**

Question # 9: CSS is called_____layer.

- Physical
- **Presentation PG # 179**
- Behavioral

- Structural

Question # 10: JS (JavaScript) is called_____layer.

- Physical
- Presentation
- **Behavioral PG # 179**
- Structural

Question # 11: Shift key is true if the shift key is_____when the event occurs.

- **Down PG # 187**
- Up
- None of the given
- Move

Question # 12: Which of the following response show internal server error?

- 404
- **500**
- 200
- 304

Question # 13: The original name of JavaScript was_____.

- JavaScript
- **LiveScript PG # 179**
- wireScript

- none of the given

Question # 14: One of the ways to create an instance of a class is____.

- **System array PG # 14**
- Factory method
- Sequential heap
- XAML compiler

Question # 15: Which "transform" property can help us to flip the element from its center?

- ScaleY
- None of given options
- ScaleX
- **RenderTransformOrigin PG # 85**

Question # 16: If we want that "Stretch" property of child element takes the available "heightor width of Parent" and shape of child elements doesn't change, then we should write —Stretch =_____||.

- 1
- **Fill PG # 100**
- UniformToFill
- Uniform

Question # 17: When a button makes itself disabled, then the value of "Focusable" property is _____?

- **False**
- Null
- True
- None of the given options

Question # 18: ManipulationCompleted gets raised after_____ is raised for all fingers.

- TouchMove
- **TouchUp PG # 116**
- TouchDown
- TouchRight

Question # 19: CSS is a_____language.

- Object Oriented
- Structuring
- **Formatting PG # 180**
- Non

Question # 20: JavaScript code is written inside file having extension____.

- JSC
- **JS**
- Javasript
- JVS

Question # 21: Which of the following is TRUE about Object data providers?

- This is useful for binding XML data
- This is useful for binding objects which are designed for binding

- **This is useful for binding objects which are not designed for binding PG # 151**
- None of the given

Question # 22: Which of the following is TRUE about **IsAsync** in context of data binding?

- This is used to access the property of an object in background
- This is used to access the property of an object in foreground
- **This is used to create data source object in background**
- This is used to create data source object in foreground

Question # 23: Refresh" is a predefined building command classified as:

- **Navigation command PG # 121**
- Media command
- Application command
- Editing command

Question # 24: Which of the following is NOT true about threads?

- Threads are useful to handle simultaneous requests.
- **A program can have more than one thread**
- Data cannot be share among the threads.
- Each thread can proceed independently of other threads.

Question # 25: Which of the following is correct way to start execution of a thread objectnamed as myThread?

- myThread.Go();
- **myThread.Start(); PG # 156**
- myThread.Begin();

- myThread.initialize ();

Question # 26: The best way to implement simple threading in Windows Forms programs is touse the class.

- **BackgroundWorker**
- Windowsform
- Simplethreadworker
- None of the given

Question # 27: A child Window is just like any other top-level window: but it automatically gets closed when the parent is closed and minimized when the parent is minimized. Such aWindow is sometimes called a_____.

- **Modeless Dialog PG # 124**
- Inherited Window
- Super Window
- Sibling Window

Question # 28: In .Net, a task that does not return a value is represented by the
_____.

- System.Threading.Tasks.Task<TResult>
- System.Threading.Tasks.Task
- System.Threarding.Tasks.
- **None of the given options**

Question # 29: In TAP, progress is handled through an_interface, which ispassed to the asynchronous method a parameter.

- **IProgress<T>**
- Progress<T> (Action<T>)

- Iprogress (T)
- Progress<T>

Question # 30: In navigation Windows, the frames are more like a/an___.

- **HTML frame Page #128**
- .Net frame
- Java frame
- Image frame

Question # 31:_____keeps the navigation history in navigation based applications.

- **Journal**
- Frame
- Web browser
- Web page

Question # 32: URI stands for:

- Universal Resource Identifier
- **Uniform Resource Identifier**
- Uniform Resource Identification
- Universal Resource Identification

Question # 33: In Objective-C, which of the following statements can be used to displaysomething in the "answer" field?

- [answerField:answer];
- [setTextanswer];
- **[answerFieldsetText:answer]; PG # 208**
- [setText answeranswerField]

Question # 34: Which one of the following is used, if we want to add resources in Window?

- <windowResources></windowResources>
- <windowsResources></windowsResources>
- **<window.Resources></window.Resources> PG # 136**

Question # 36: Which of the following is used to set the view to default?

- NSCoder

- clearAll
- **IB** **PG # 217**
- Redraw

Question # 37: In context of data binding, when we want to apply both sorting and groupingtogether then the rule is that_.

- Grouping will be applied before sorting
- **Sorting will be applied before Grouping PG # 150**
- Both shall be applied different properties
- None of the given

Question # 38: What is the word —Color‖ in given below code? UIColor" (^) (Line*,int anotherArg)

- Return Type of Block
- Notation to specify that it is a block
- Name of Variable
- **Block Variable PG # 222**
- <windows.Resources></windows.Resources>

Question # 35: If we assign an NSString to a possession for its possessionName, and then werelease the string, it will be.

- Allocated
- Decreased
- Instance created
- **Destroyed**

Question # 39:_____helps us in provisioning ease of access to data source for databinding.

- **Data Provider**
- Interfaces
- Source
- Data Organizer

Question # 40: What is the purpose of given below code? [self.viewaddSubview:imageView];

- To make sure that image is not scaled incorrectly
- To create the image View
- To set the image
- **Add the image to this View controllers PG # 226**

Question # 41: What is the purpose of given below code? NSArray *folders NSSearchPathForDirectoriesInDomains(NSD ocumentDirectory,NSUserDomaianIask, YES);

- **Get the document folder(s) PG # 226**
- Create the folder

- Get the first folder
- Add the image to first folder

Question # 42: Which JQuery object is used with bind() function to pass data to an eventhandling function?

- Target
- Bind
- **Data** **PG # 187**
- Jobj

Question # 43: Which of the following technologies is not used in AJAX?

- DOM
- DHTML
- **Flash**
- Css

Question # 44: Which function is used to create cancel button event of dialog box_____.

- OnInitCancel()
- OnClickCancel()
- CancelClick()
- **OnCancel()**

Question # 45: Consider the following C# code segment: button.Background = (Brush) new BrushConverter().ConvertFrom (—SystemColors.WindowBrush); Which one of the following isequivalent XAML code?

- <Button Background= —SystemColors.WindowsBrush />
- Button Background= —SystemColor.WindowsBrush />
- <Button Background= —SystemColor.WindowBrush />
- **<Button Background= "SystemColors.WindowBrush"/>** **PG # 132**

Question # 46: Multi touch events are categorized into_____and_____.

- Touch events, touchup events
- Touchdown events, touchup events
- Advanced touch events, low-level manipulation events
- Basic touch events, higher-level manipulation events

Question # 47: Which of the following components generates an input Event?

- Printer
- **Multi Touch**

- Plotter
- Speaker

Question # 48: Using_____requires more overhead than_____because of theextra tracking.

- StaticResource, DynamicResource
- **DynamicResource, StaticResource PG # 137**
- PermanentResource, TemporaryResource
- TemporaryResource, PermanentResource

Question # 49: We use "UIElement" property ClipToBounds= —false when _____.

- We want area child element don't cross the boundary of parent
- None of given
- **We want area of child element is allowed to cross the boundary of parentPG # 97**
- Parent area also increases on increasing of area of child element

Question # 50:__is used to perform navigation.

- Session
- **Hyperlink PG # 127**
- Hypertext markup language code
- Internet protocol

Question # 51: To design a complex Interface, which of the following technique is best toachieve the required results?

- Panels composed in three panels
- **Panels composed within panels**
- Panels composed in two panels
- Panels composed in single panel

Question # 52: Which of the following is not a string format property throughout WPF?

- ContentStringFormat
- ItemStringFormat
- **RowHeaderStringFormat**
- StringFormat

Question # 53: In multi touch events, a/an_____id is assigned to each individualevent.

- Device
- Processor
- CPU

- **Event**

Question # 54: Commonly we have two types of custom command bindings named as___.

- StartBindings and StopBindings
- EventBindings and InstanceBindings
- ButtonBindings and TextboxBindings
- **KeyBindings and MouseBindings PG # 122**

Question # 55: Which of the following is an example of tunnel key event?

- Key up
- Preview key entered
- **Preview key down**
- Key down

Question # 56: In context of data binding,_____contains the current item to get itsynchronized with data Source.

- String
- Template
- View
- **List**

Question # 57:_____and_____are the two important properties of Bindingobject.

- Start, Destination
- Items, Selected
- **Source, Path**
- BindTo, BindFrom

Question # 58: Which one of the following panels is most powerful, versatile andcustomizable?

- **Grid Panel PG # 92**
- Canvas panel

- Doc panel
- Stack panel

Question # 59: In order to resize the row and column in Grid panel,__is used.

- Grid divider
- Grid compiler
- Grid converter
- **Grid splitter PG # 95**

Question # 60: Data binding is about tying together arbitrary .NET__.

- **Objects PG # 140**
- Functions
- Properties
- Variables

Question # 61:_____supports the creation of applications that run directly in a webbrowser.

- **WPF**
- C/C++
- C#
- HTML

Question # 62: Which of the following operation cannot be performed by using a View?

- Grouping
- Filtering
- Sorting
- **Searching PG # 146**

Question # 63: To check that the user either pressed left Alt key or right

Alt key, the____is used.

- IsKeyDown
- IsDown
- **keyboardDevice.IsKeyDown PG # 113**
- KeyStates.IsKeyDown

Question # 64: In _____browser app, you change"<TargetZone>Internet</TargetZone>"to "<TargetZone>Custom</TargetZone>"

- Partial-trust
- Full-reject
- Partial-reject
- Full-trust

Question # 65:_____provides the logic behind the "Back" and "Forward" buttons.

- Hyperlink
- Navigation Window
- IFrame
- **Journal PG # 138**

Question # 66: In multi touch events, when multiple fingers are touching simultaneously,these events get raised for each finger.

- **Independently**
- Completely
- Partially
- Dependently

Question # 67: In a code segment, if the Visibility=||Collapsed|| then which of the following events are not generated?

- Key down Events

- **Mouse Events PG # 113**
- Routed Events
- Keyboard Events

Question # 68: A Window can spawn_____number of additional Windows byinstantiating a Window-derived Class and calling Show (...).

- Four
- **Any PG # 124**
- Six
- Five

Question # 69: Frame has a_____property used in enabling or disabling the bar.

- ShowsNavigationUI
- BlocknavigationUI
- **NavigationUIVisibility PG # 129**
- NavigationUIBlock

Question # 70: If we insert some objects in a single cell in Grid panel, then these objects areplaced on _

- **One on Top of the other**
- One after the other
- One on Bottom of the other
- One on Side of the other

Question # 71: A _____receives keyboard events only if it has keyboard focus.

- Mouse
- Keyboard Controls

- Keyboard
- **UIElement**

Question # 72: To remove data binding between the source and the target, we can use _____function.

- **ClearBinding PG # 151**
- ResetBinding
- StopBinding
- RemoveBinding

Question # 73: A/An_____screen is an image that appears while a game or program isloading.

- Opening
- Initial
- **Splash**
- Startup

Question # 74: We can perform sorting through View by using an object of_____class

- ViewOrder
- ViewDescription
- SortOrder
- **SortDescription PG # 158**

Question # 75: "Binding" technique binds two properties together and keeps a _____open.

- Memory location
- Selection

- Choice
- **Communication channel PG # 140**

Question # 76: Which one of the following can be used to set dependency property values?

- Temporary resource
- Permanent resource
- **Dynamic resource PG # 137**
- Static resource

Question # 77: Which one of the following is used, if we want to add a window resource inprocedural code?

- Window.Resource.Add ();
- <Widows.Resources></Widows.Resources>
- Window.Resoruces.Add();
- NONE

Question # 78: We use "GetDefaultView" method of_____to get the default View.

- **CollectionViewSource PG # 146**
- DataSource
- ViewDescription
- ViewCollection

Question # 79: In case of HTML page navigation, you must use the_____of"Navigate" that_____a URI.

- Overload, rejects
- IFrame, rejects
- IFrame, accepts

- **Overload, accepts**

Question # 80: The Java programming language has a specific class for creating splashscreens, called_.

- Java.awt.start
- Java.lang.splash
- Java.start.screen
- **Java.awt.SplashScreen**

Question # 81: In case of integrating Navigation to XBAP, "ShowNavigationUI" should be setto___in order to bypass integration.

- **False**
- True
- Null
- 0

Question # 82: Which of the followings are keyboard events?

- Key entered, key exist
- Key strike, key release
- Key pressed, key released
- **Key down, key up PG # 113**

Question # 83: To plug custom logic, you need to add a___to the element thatwill execute the command or any parent element.

- Execute command
- Helpcanexecute
- **CommandBinding**
- **RoutedUICommand**

Question # 84: A class that implements the _lCommand` and supports _bubbling` just like arouted event is known as___.

- **RoutedUlCommand PG # 121**
- Canexecute
- Executed command
- CanexecuteChanged

Question # 85: Which of the following is not true about task completion source?

- Provide exception handling facility
- Support result return facility
- Provide Continuation facility
- **it is executable thing PG # 175**

Question # 86: "JavaScript"_____is available in web browser.

- **Interpreter**
- Compiler
- Linker
- Conjunction

Question # 87: The "Task Based Asynchronous pattern" method returns either a "Task" or a "Task<TResult>", based on whether the corresponding____method returns "void" ora type "TResult".

- Callback
- Return
- **Synchronous**
- Asynchronous

Question # 88: Which of the following C# Method reports a progress

change?

- Finalize
- **OnReport**
- GetType
- GetHashCode

Question # 89: Threadpool is used to_____.

- To make easy the process of thread creation
- To make easy debugging process
- **To save time of thread creation PG# 160**
- To avoid complexity

Question # 90: Which of the following is not a part of "cancellation" task?

- **The calling thread does forcibly end the task**
- Notice and respond to the cancellation request in your user delegate
- Pass a cancellation token to your user delegate and optionally to the task instance
- Create and start a cancelable task

Question # 91: DOM is a___representation of data.

- Tree
- Graphical
- Object
- **None of the given**

Question # 92: which of the following Object creates a cancellation token?

- OperationCanceledtoken
- OperationCanceled

- CancellationToken
- **CancellationTokenSource PG#172**

Question # 93: which of the following is not a "JQuery" filter?

- Even
- Odd
- Has
- **Write PG#184**

Question # 94: Which of the following -Event‖ property is used to show the —distance (in pixels)‖ of the mouse pointer from the left edge of the browser window?

- pageY
- **pageX PG # 187**
- ScreenX
- ScreenY

Question # 95: —async and_____keywords in C# are the heart of asynchronousprograming.

- Return
- **Await**
- Wait
- String

Question # 96:_____Mode of binding is opposite to OneWay mode ofbinding.

- **OneWay To source PG#153**
- TwoWay
- One Time

- Two Time

Question # 97:_____Can add/remove/change —CSS‖ properties based on input or mouse clicks.

- J#
- JQuery
- HTML
- **None of the given option PG#180**

Question # 98: —JavaScript‖ is_____scripting Language.

- Interpreter
- Browser
- Server
- **Client**

Question # 99:_____solves —JavaScript‖ complexity and browser incompatibilities.

- Jhtml
- RQuery
- **JQuery PG#179**
- J#

Question # 100:_____is more than one thing happening at the same time.

- Cohesion
- Congruence
- Coherence
- Concurrency

Question # 101: Which of the following —Event —properties is used with —key press event‖ to determine the numeric code for the key that was pressed? .

- Page X
- **Which PG#187**
- Shift Key
- Target

Question # 102: Which of the following is correct way to temporarily stop execution of athread object named as —myThread‖ for 500 milisec?

- myThread.Sleep(0.50);
- myThread.Stop(0.05);
- **myThread.Sleep(500);**
- myThread.Stop(500);

Question # 103: HTML is a_____helper function.

- RQuery
- Browser
- **jQuery PG#184**
- JS

Question # 104: Which of the following server(s) support -AJAX‖?

- Both SMTP and HTTP
- **HTTP**
- www
- SMTP

Question # 105: Why we use —Alloc message?

- To create a class
- **To create an instance**
- To extend functions
- To inherent methods

Question # 106: Objective – c introduces_____messaging in c?

- **Smalltalk style PG# 197**
- Short session
- Group
- New Style

Question # 107: Threadpool create and reduce real threads using hillclimbing algo to _____.

- **To maximize CPU usage Pg#160**
- To reduce the cost of thread creation
- To maximize memory utilization
- To minimize CPU usage

Question # 108: Which of the following is not handled by the —Task Parallel Library (TPL)**||**?

- Partitioning of the work
- **Progress report handling**
- Scheduling of threads
- Scales the degree of concurrency

Question # 109: "Callback" method is used to_____the operation.

- **Terminate**
- Pause

- Invoke
- Revoke

Question # 110: While dealing with "threads" in ".Net", one can avoid performance bottlenecks and enhance the overall responsiveness of his / her application by using _____programming.

- Multithreading programming
- Synchronous programming
- Parallel programming
- Asynchronous programming

Question # 111: Interact with web page layout

- Java script
- **HTML**
- XML
- DOM

Question # 112:_____are high order functions that compose, combine, or otherwisemodify functions in useful and interesting ways.

- **Combinators**
- None of the given
- Separators
- Modifiers

Question # 113: Which of these are defined as the model object?

- UILabel *questionField;
- NSMutableArray *questions@field;
- **NSMutableArray *questions; PG # 205**
- Question= 0;

Question # 114: Which of the following "Event" property is used to show the "distance (inpixels)" of the mouse pointer from top edge of the Monitor?

- **ScreenY PG#187**
- pageX
- pageY
- ScreenX

Question # 115: What is —nil| like?

- **Null Page # 197**
- zero
- terminate
- move to new line

Question # 116: Which of the following "Event" property is used to show the "distance (inpixels)" of the mouse pointer from top edge of the browser window?

- **pageY Page # 187**
- pageX
- screenX
- screenY

Question # 117: Observe the following piece of code taken out from Main() function: Thread myThread = new Thread (Go); myThread.Strat(); myThread.Join(); What will be the effect ofinstruction myThread.Join()?

- Execution of myThread will be joined with existing running threads
- myThread will be executed after Main thread is completed
- **Main thread execution will be suspended till completion of myThread**
- Both Main thread and myThread can now

share the data Question # 118: What message

is used to send to destroy the object?

- stop
- delete
- free
- **release**

Question # 119: What is NSMutableArray?

- Object
- **Class Reference**
- Method
- Group

Question # 120: AJAX stands for_____.

- Abstract JSON and XML
- Asynchronous JavaScript and XHtml
- **Asynchronous JavaScript and XML**
- Abstract Java and XML Library

Question # 121: What is the purpose of the following "JQuery" code?
$(_#navbar a‘)

- Find all the element whose name is navbar and then find a descendent a
- Find all the element whose name is navbar and then find a ancestor a
- **Find all the element whose ID is navbar and then find a**

descendent a

- Find all the element whose ID is navbar and then find a ancestor a

Question # 122: The pairing of labels and arguments is an important feature of_____.

- C++
- Scripts
- **Objective-C page # 197**
- PHP

Question # 123: Web servers receive request and responds as____.

- Html only
- Plan Text only
- JSON only
- **Html, plain text and JSON Page #192**

Question # 124: Which of the following class supports data parallelism in "Task ParallelLibrary"?

- System.Threading.Tasks
- System.Task.Parallel
- **System.threading.Task.Parallel**

System.Task.Threading.Parallel Question # 125: Which of the following "Event" property is used to show the "distance (inpixels)" of the mouse pointer from left edge of the Monitor?

- pageX
- pageY
- ScreenY
- **ScreenX PG# 187**

AL-JUNAID TECH INSTITUTE

Question # 126: which of the following response show file not found error?

- **404**
- 500
- 200
- 304
-

FinalTerm Papers Solved MCQS with Reference

Question # 1: In Xaml browser application, how much isolated memory we can use?

- 256 kb
- 128 kb
- **512 kb** **PG # 132**
- 1024 kb

Question # 2: Which property will be used if we want to restrict the user to don"t increase the width of a WPF element from a specific amount?

- **MaxWidth** **Page # 81**
- MaximumWidth
- WidthMaximum
- None of given options

Question # 3: Which one is the correct syntax for declaring attribute?

- **[Obsolete]** **page # 42**
- {Obsolete}
- (Obsolete)
- *Obsolete

Question # 4: In C#, comments are written using _____.

- # and */ /*
- **// and /* */** **page # 14**
- # and /*
- // and */ /*

Question # 5: An event that is raised whenever the value of "CanExecute" changes is known as _____.

- CanExecute
- **CanExecuteChanged**
- Command
- Execute

Execute—The method that executes the command-specific logic
CanExecute—A method returning true if the command is enabled or false if it is disabled
CanExecuteChanged—An event that is raised whenever the value of CanExecute changes

Question # 6: _____ is an event that is introduced into an event processing system by an event producer.

- Event Producer
- Event Consumer
- **Raw Event** **Page # 11**
- Event Stream

Question # 7: Whenever an attribute value is enclosed in curly braces "{}", the XAML compiler/parser treats it as a/an _____ rather than a/an _____.

- Markup event, property
- Markup property, Event
- Literal string, Markup extension value
- **Markup extension value, literal string** **page # 65**

Question # 8: HTML is called _____ layer.

- Physical
- Presentation
- Behavioral
- **Structural** **PG # 195**

HTML: structural layer and CSS: presentation layer and JS: behavioral layer.

Question # 9: CSS is called _____ layer.

- Physical
- **Presentation** **PG # 179**
- Behavioral
- Structural

HTML: structural layer and CSS: presentation layer and JS: behavioral layer.

Question # 10: JS (JavaScript) is called _____ layer.

- Physical
- Presentation
- **Behavioral** **PG # 179**
- Structural

HTML: structural layer and CSS: presentation layer and JS: behavioral layer.

Question # 11: Shift key is true if the shift key is _____ when the event occurs.

- **Down** **PG # 187**
- Up
- None of the given
- Move

shiftKey	Is *true* if the shift key is down when the event occurs.

Question # 12: Which of the following response show internal server error?

- 404
- **500** **Click here for more detail**
- 200
- 304

Question # 13: The original name of JavaScript was _____.

- JavaScript
- **LiveScript** **PG # 179**
- wireScript
- none of the given

JS has nothing to do with Java, originally named LiveScript but renamed to associate with the then hot Java.

Question # 14: One of the ways to create an instance of a class is_____.

- **System array** **PG # 14**
- Factory method
- Sequential heap
- XAML compiler

Question # 15: Which "transform" property can help us to flip the element from its center?

- ScaleY
- None of given options
- ScaleX
- **RenderTransformOrigin** **PG # 85**

Question # 16: If we want that "Stretch" property of child element takes the available "height or width of Parent" and shape of child elements doesn't change, then we should write "Stretch = _____".

- 1
- **Fill** **PG # 100**
- UniformToFill
- Uniform

Question # 17: When a button makes itself disabled, then the value of "Focusable" property is _____?

- **False**
- Null
- True
- None of the given options

Question # 18: ManipulationCompleted gets raised after _____ is raised for all fingers.

- TouchMove
- **TouchUp** **PG # 116**
- TouchDown
- TouchRight

Question # 19: CSS is a _____ language.

- Object Oriented
- Structuring
- **Formatting** **PG # 180**
- None of the given options

CSS is a formatting language

Question # 20: JavaScript code is written inside file having extension_____.

- JSC
- **JS** **Click Here For More Detail**
- Javasript
- JVS

JavaScript files have the file extension **.js**.

Question # 21: Which of the following is TRUE about Object data providers?

- This is useful for binding XML data
- This is useful for binding objects which are designed for binding
- **This is useful for binding objects which are not designed for binding** **PG # 151**
- None of the given

Binding to a method is useful for classes that are not designed for data binding.

Question # 22: Which of the following is TRUE about **IsAsync** in context of data binding?

- This is used to access the property of an object in background
- This is used to access the property of an object in foreground
- **This is used to create data source object in background**
- This is used to create data source object in foreground

Question # 23: Refresh" is a predefined building command classified as:

- **Navigation command** **PG # 121**
- Media command
- Application command
- Editing command

NavigationCommands e.g: BrowseBack,BrowseForward, BrowseHome, BrowseStop, Favorites, FirstPage, GoToPage, LastPage, NextPage, PreviousPage, **Refresh,** Search, Zoom, and more.

Question # 24: Which of the following is NOT true about threads?

- Threads are useful to handle simultaneous requests.
- **A program can have more than one thread**
- Data cannot be share among the threads.
- Each thread can proceed independently of other threads.

Question # 25: Which of the following is correct way to start execution of a thread object named as myThread?

- myThread.Go();
- **myThread.Start();** **PG # 156**
- myThread.Begin();
- myThread.initialize ();

Question # 26: The best way to implement simple threading in Windows Forms programs is to use the class.

- **BackgroundWorker** **Click here for more detail**
- Windowsform
- Simplethreadworker
- None of the given

Question # 27: A child Window is just like any other top-level window: but it automatically gets closed when the parent is closed and minimized when the parent is minimized. Such a Window is sometimes called a _____.

- **Modeless Dialog** **PG # 124**
- Inherited Window
- Super Window
- Sibling Window

Any number of child windows can be made by instantiating a Window dervied class and calling Show. Child window like parent window but gets closed when parent and similarly minimized, also called modeless dialog.

Question # 28: In .Net, a task that does not return a value is represented by the _____.

- System.Threading.Tasks.Task<TResult>
- System.Threading.Tasks.Task
- System.Threarding.Tasks.
- **None of the given options**

Question # 29: In TAP, progress is handled through an_____ interface, which is passed to the asynchronous method a parameter.

- **IProgress<T>** **Click Here For More Detail**
- Progress<T> (Action<T>)
- Iprogress (T)
- Progress<T>

In the TAP, progress is handled through an IProgress<T> interface (described later in this document) passed into the asynchronous method as a parameter named “progress”.

Question # 30: In navigation Windows, the frames are more like a/an_____.

- **HTML frame** **Page #128**
- .Net frame
- Java frame
- Image frame

NavigationWindow more like a top-level window whereas Frame more like an HTML frame

Question # 31: _____keeps the navigation history in navigation based applications.

- **Journal** **Click Here For More Detail**
- Frame
- Web browser
- Web page

Question # 32: URI stands for:

- Universal Resource Identifier
- **Uniform Resource Identifier** **Click here for more detail**
- Uniform Resource Identification
- Universal Resource Identification

URL: In computing, a Uniform Resource Locator (URL) is a subset of the **Uniform Resource Identifier** (URI) that specifies where an identified resource is available and the mechanism for retrieving it. The URL shows you where you can find the database on the internet and which protocol you should use.

Question # 33: In Objective-C, which of the following statements can be used to display something in the "answer" field?

- [answerField:answer];
- [setTextanswer];
- **[answerFieldsetText:answer];** **PG # 208**
- [setText answeranswerField]

// Display it in the answer field
[answerField setText:answer];

Question # 34: Which one of the following is used, if we want to add resources in Window?

- <windowResources></windowResources>
- <windowsResources></windowsResources>
- **<window.Resources></window.Resources>** **PG # 136**
- <windows.Resources></windows.Resources>

Question # 35: If we assign an NSString to a possession for its possessionName, and then we release the string, it will be_____.

- Allocated
- Decreased
- Instance created
- **Destroyed** Click here for more detail

Question # 36: Which of the following is used to set the view to default?

- NSCoder
- clearAll
- **IB** **PG # 217**
- Redraw

use IB to set the view to default

Question # 37: In context of data binding, when we want to apply both sorting and grouping together then the rule is that _____.

- Grouping will be applied before sorting
- **Sorting will be applied before Grouping** **PG # 150**
- Both shall be applied different properties
- None of the given

Question # 38: What is the word “Color” in given below code? UIColor" (^) (Line*,int anotherArg)

- Return Type of Block
- Notation to specify that it is a block
- Name of Variable
- **Block Variable** **PG # 222**

Question # 39: _____helps us in provisioning ease of access to data source for data binding.

- **Data Provider** **Click Here For More Detail**
- Interfaces
- Source
- Data Organizer

Question # 40: What is the purpose of given below code? [self.viewaddSubview:imageView];

- To make sure that image is not scaled incorrectly
- To create the image View
- To set the image
- **Add the image to this View controllers** **PG # 226**

/* Add the image to this view controller 's view */
[self.view addSubview:imageView];

Question # 41: What is the purpose of given below code?
NSArray *folders
NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomaianIask, YES);

- **Get the document folder(s)** **PG # 226**
- Create the folder
- Get the first folder
- Add the image to first folder

/* Get the document folder(s) */

Question # 42: Which JQuery object is used with bind() function to pass data to an event handling function?

- Target
- Bind
- **Data** **PG # 187**
- Jobj

data	A jQuery object used with the *bind()* function to pass data to an event handling function (see page 177).

Question # 43: Which of the following technologies is not used in AJAX?

- DOM
- DHTML
- **Flash** **Click here for more detail**
- Css

Ajax doesn't use flash technology. Technologies used by ajax: JavaScript Loosely typed scripting language JavaScript function is called when an event in a page occurs Glue for the whole AJAX operation DOM API for accessing and manipulating structured documents Represents the structure of XML and HTML documents CSS Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript XMLHttpRequest JavaScript object that performs asynchrous interaction with the server

CS411 Quiz # 3 2014 & 2015
From Lectures (22 TO 32)

Question # 44: Which function is used to create cancel button event of dialog box _____.

- OnInitCancel()
- OnClickCancel()
- CancelClick()
- **OnCancel()**

Question # 45: Consider the following C# code segment: button.Background = (Brush) new BrushConverter().ConvertFrom ("SystemColors.WindowBrush"); Which one of the following is equivalent XAML code?

- <Button Background= "SystemColors.WindowsBrush"/>
- <Button Background= "SystemColor.WindowsBrush"/>
- <Button Background= "SystemColor.WindowBrush"/>
- **<Button Background= "SystemColors.WindowBrush"/>** **PG # 132**

Question # 46: Multi touch events are categorized into _____ and _____.

- Touch events, touchup events
- Touchdown events, touchup events
- Advanced touch events, low-level manipulation events
- **Basic touch events, higher-level manipulation events** **PG # 114**

Question # 47: Which of the following components generates an input Event?

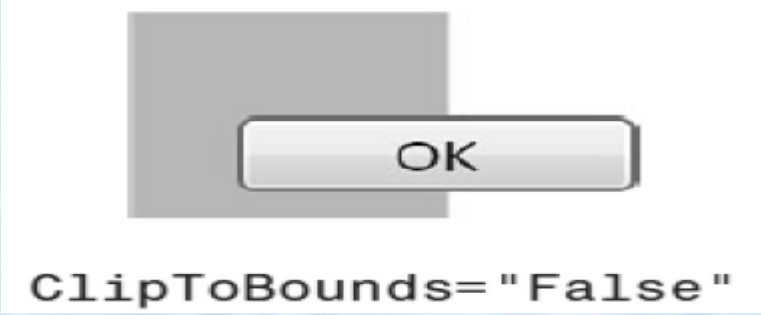
- Printer
- **Multi Touch**
- Plotter
- Speaker

Question # 48: Using _____ requires more overhead than _____ because of the extra tracking.

- StaticResource, DynamicResource
- **DynamicResource, StaticResource** **PG # 137**
- PermanentResource, TemporaryResource
- TemporaryResource, PermanentResource

Question # 49: We use "UIElement" property ClipToBounds= “false” when _____.

- We want area child element don’t cross the boundary of parent
- None of given
- **We want area of child element is allowed to cross the boundary of parent** **PG # 97**
- Parent area also increases on increasing of area of child element



Question # 50: _____ is used to perform navigation.

- Session
- **Hyperlink** **PG # 127**
- Hypertext markup language code
- Internet protocol

You can perform navigation in three main ways: Calling the Navigate method, **Using Hyperlinks**, Using the journal.

Question # 51: To design a complex Interface, which of the following technique is best to achieve the required results?

- Panels composed in three panels
- **Panels composed within panels**
- Panels composed in two panels
- Panels composed in single panel

Question # 52: Which of the following is not a string format property throughout WPF?

- ContentStringFormat
- ItemStringFormat
- **RowHeaderStringFormat**
- StringFormat

Question # 53: In multi touch events, a/an _____ id is assigned to each individual event.

- Device
- Processor
- CPU
- **Event**

Question # 54: Commonly we have two types of custom command bindings named as _____.

- StartBindings and StopBindings
- EventBindings and InstanceBindings
- ButtonBindings and TextboxBindings
- **KeyBindings and MouseBindings** **PG # 122**

Question # 55: Which of the following is an example of tunnel key event?

- Key up
- Preview key entered
- **Preview key down**
- Key down

Question # 56: In context of data binding, _____ contains the current item to get it synchronized with data Source.

- String
- Template
- View
- **List**

Question # 57: _____and _____ are the two important properties of Binding object.

- Start, Destination
- Items, Selected
- **Source, Path**
- BindTo, BindFrom

Question # 58: Which one of the following panels is most powerful, versatile and customizable?

- **Grid Panel** **PG # 92**
- Canvas panel
- Doc panel
- Stack panel

Question # 59: In order to resize the row and column in Grid panel, _____ is used.

- Grid divider
- Grid compiler
- Grid converter
- **Grid splitter** **PG # 95**

Question # 60: Data binding is about tying together arbitrary .NET _____.

- **Objects** **PG # 140**
- Functions
- Properties
- Variables

Question # 61: _____ supports the creation of applications that run directly in a web browser.

- **WPF** **Click Here For More Detail**
- C/C++
- C#
- HTML

Question # 62: Which of the following operation cannot be performed by using a View?

- Grouping
- Filtering
- Sorting
- **Searching** **PG # 146**

View Supported items, 1. Grouping, 2. Filtering, 3. Sorting, 4. Navigation

Question # 63: To check that the user either pressed left Alt key or right Alt key, the _____ is used.

- IsKeyDown
- IsDown
- **keyboardDevice.IsKeyDown** **PG # 113**
- KeyStates.IsKeyDown

You can use keyboarddevice.iskeydown to even check if left or right alt is down etc.

Question # 64: In _____ browser app, you change"<TargetZone>Internet</TargetZone>" to "<TargetZone>Custom</TargetZone>"

- Partial-trust
- Full-reject
- Partial-reject
- **Full-trust** **PG # 132**

Question # 65: _____ provides the logic behind the "Back" and "Forward" buttons.

- Hyperlink
- Navigation Window
- IFrame
- **Journal** **PG # 138**

Journal provides logic behind back and fwd.

Question # 66: In multi touch events, when multiple fingers are touching simultaneously, these events get raised for each finger _____.

- **Independently** **Click Here For More Detail**
- Completely
- Partially
- Dependently

Question # 67: In a code segment, if the Visibility=”Collapsed” then which of the following events are not generated?

- Key down Events
- **Mouse Events** **PG # 113**
- Routed Events
- Keyboard Events

If Visibility=Collapsed no mouse events are generated but opacity=0 generates all events.

Question # 68: A Window can spawn _____ number of additional Windows by instantiating a Window-derived Class and calling Show (...).

- Four
- **Any** **PG # 124**
- Six
- Five

Question # 69: Frame has a _____ property used in enabling or disabling the bar.

- ShowsNavigationUI
- BlocknavigationUI
- **NavigationUIVisibility** **PG # 129**
- NavigationUIBlock

Question # 70: If we insert some objects in a single cell in Grid panel, then these objects are placed on _____

- **One on Top of the other**
- One after the other
- One on Bottom of the other
- One on Side of the other

Question # 71: A _____ receives keyboard events only if it has keyboard focus.

- Mouse
- Keyboard Controls
- Keyboard
- **UIElement** **Click Here For More Detail**

Question # 72: To remove data binding between the source and the target, we can use _____ function.

- **ClearBinding** **PG # 151**
- ResetBinding
- StopBinding
- RemoveBinding

Question # 73: A/An _____ screen is an image that appears while a game or program is loading.

- Opening
- Initial
- **Splash**
- Startup

Question # 74: We can perform sorting through View by using an object of _____ class.

- ViewOrder
- ViewDescription
- SortOrder
- **SortDescription** **PG # 158**

Question # 75: "Binding" technique binds two properties together and keeps a _____ open.

- Memory location
- Selection
- Choice
- **Communication channel** **PG # 140**

Binding binds two properties together and keeps a communication channel open.

Question # 76: Which one of the following can be used to set dependency property values?

- Temporary resource
- Permanent resource
- **Dynamic resource** **PG # 137**
- Static resource

dynamic can only be used to set dep. prop. values.

Question # 77: Which one of the following is used, if we want to add a window resource in procedural code?

- Window.Resource.Add ();
- <Widows.Resources></Widows.Resources>
- Window.Resoruces.Add();
- **<Window.Resources></Window.Resources>** **PG # 136**

Question # 78: We use "GetDefaultView" method of _____ to get the default View.

- **CollectionViewSource** **PG # 146**
- DataSource
- ViewDescription
- ViewCollection

```
// Get the default view
ICollectionView view = CollectionViewSource.GetDefaultView(
this.FindResource (" photos "));
```

Question # 79: In case of HTML page navigation, you must use the _____ of "Navigate" that _____ a URI.

- Overload, rejects
- IFrame, rejects
- IFrame, accepts
- **Overload, accepts**

Question # 80: The Java programming language has a specific class for creating splash screens, called _____.

- Java.awt.start
- Java.lang.splash
- Java.start.screen
- **Java.awt.SplashScreen** **Click here for more detail**

Question # 81: In case of integrating Navigation to XBAP, "ShowNavigationUI" should be set to _____ in order to bypass integration.

- **False**
- True
- Null
- 0

Question # 82: Which of the followings are keyboard events?

- Key entered, key exist
- Key strike, key release
- Key pressed, key released
- **Key down, key up** **PG # 113**

Question # 83: To plug custom logic, you need to add a _____ to the element that will execute the command or any parent element.

- Execute command
- Helpcanexecute
- **CommandBinding** **Click here for more detail**
- RoutedUlCommand

To plug in custom logic, you need to add a CommandBinding to the element that will execute the command *or any parent element* (thanks to the bubbling behavior of routed commands).

Question # 84: A class that implements the 'ICommand' and supports 'bubbling' just like a routed event is known as _____.

- **RoutedUICommand** **PG # 121**
- Canexecute
- Executed command
- CanexecuteChanged

CS411 Quiz # 4 2014 & 2015 From Lectures (32 TO 42)

Question # 85: Which of the following is not true about task completion source?

- Provide exception handling facility
- Support result return facility
- Provide Continuation facility
- **it is executable thing** **PG # 175**

Question # 86: "JavaScript" _____is available in web browser.

- **Interpreter** **Click here for more Detail**
- Compiler
- Linker
- Conjunction

JavaScript is an interpreted language, not a compiled language

Question # 87: The "Task Based Asynchronous pattern" method returns either a "Task" or a "Task<TResult>", based on whether the corresponding _____ method returns "void" or a type "TResult".

- Callback
- Return
- **Synchronous** **Click here for more Detail**
- Asynchronous

The TAP method returns either a System.Threading.Tasks.Task or a System.Threading.Tasks.Task<TResult>, based on whether the corresponding synchronous method returns void or a type TResult.

Question # 88: Which of the following C# Method reports a progress change?

- Finalize
- **OnReport** **Click here for more Detail**
- GetType
- GetHashCode

OnReport Reports a progress change.

Question # 89: Threadpool is used to _____.

- To make easy the process of thread creation
- To make easy debugging process
- **To save time of thread creation** **PG# 160**
- To avoid complexity

Question # 90: Which of the following is not a part of "cancellation" task?

- **The calling thread does forcibly end the task** **Click here for more Detail**
- Notice and respond to the cancellation request in your user delegate
- Pass a cancellation token to your user delegate and optionally to the task instance
- Create and start a cancelable task

1. Create and start a cancelable task.
2. Pass a cancellation token to your user delegate and optionally to the task instance.
3. Notice and respond to the cancellation request in your user delegate.
4. Optionally notice on the calling thread that the task was canceled

Question # 91: DOM is a _____ representation of data.

- Tree
- Graphical
- Object
- **None of the given**

Question: DOM is a graphical or tree representation of data? Still confuse?

Instructor's Reply on MDB: No, It is not graphical and not a tree representation of any module. It is a logical representation of API or any library. In simple words, it is a way to call a library into your programming. For example you called a print function so this function has its own structure with different type of parameters and calling it. You should follow that structure for the utilization of this function. Same like the case the implementation of Document object model it has a structure that you should follow i.e. table tag should be used in html tag.

Question # 92: which of the following Object creates a cancellation token?

- OperationCanceledtoken
- OperationCanceled
- CancellationToken
- **CancellationTokenSource** **PG#172**

Cancellation. Cancel is on cancellationtokensource. Most have builtin cancellation support.

Question # 93: which of the following is not a "JQuery" filter?

- Even
- Odd
- Has
- **Write** **PG#184**

Jquery filters are :even :odd $('.striped tr:even') :first :last :not $('a:not(.navbutton)'); :has $('li:has(a)') — diff from descendent :contains $('a:contains(Click Me!)') :hidden :visible $('div:hidden').show();

Question # 94: Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from the left edge of the browser window?

- pageY
- **pageX** **PG # 187**
- ScreenX
- ScreenY

Event property	Description
pageX	The distance (in pixels) of the mouse pointer from the left edge of the browser window.

Question # 95: "async" and _____ keywords in C# are the heart of asynchronous programing.

- Return
- **Await** **Click here for more Detail**
- Wait
- String

The Async and Await keywords in Visual Basic and the async and await keywords in C# are the heart of async programming.

Question # 96: _____ Mode of binding is opposite to OneWay mode of binding.

- **OneWay To source** **PG#153**
- TwoWay
- One Time
- Two Time

Question # 97: _____Can add/remove/change "CSS" properties based on input or mouse clicks.

- J#
- JQuery
- HTML
- **None of the given option** **PG#180**

JS can add/remove/change CSS properties based on input or mouse clicks.

Question # 98: "JavaScript" is _____ scripting Language.

- Interpreter
- Browser
- Server
- **Client**

JavaScript is most commonly used as a client side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it.

Question # 99: _____ solves "JavaScript" complexity and browser incompatibilities.

- Jhtml
- RQuery
- **JQuery** **PG#179**
- J#

Jquery is a JS library intended to make JS programming easier. Jquery solves JS complexity and browser incompatibilities.

Question # 100: _____ is more than one thing happening at the same time.

- Cohesion
- Congruence
- Coherence
- **Concurrency** **PG#156**

Question # 101: Which of the following "Event "properties is used with "key press event" to determine the numeric code for the key that was pressed? .

- Page X
- **Which** **PG#187**
- Shift Key
- Target

pageX	The distance (in pixels) of the mouse pointer from the left edge of the browser window.
shiftKey	Is *true* if the shift key is down when the event occurs.
which	Use with the keypress event to determine the numeric code for the key that was pressed (see tip, next).
target	The object that was the "target" of the event—for example, for a *click()* event, the element that was clicked.

Question # 102: Which of the following is correct way to temporarily stop execution of a thread object named as "myThread" for 500 milisec?

- myThread.Sleep(0.50);
- myThread.Stop(0.05);
- **myThread.Sleep(500);**
- myThread.Stop(500);

Question # 103: HTML is a _____ helper function.

- RQuery
- Browser
- **jQuery** **PG#184**
- JS

Question # 104: Which of the following server(s) support "AJAX"?

- Both SMTP and HTTP
- **HTTP**
- www
- SMTP

Question # 105: Why we use "Alloc" message?

- To create a class
- **To create an instance**
- To extend functions
- To inherent methods

Question # 106: Objective – c introduces _____ messaging in c?

- **Smalltalk style** **PG# 197**
- Short session
- Group
- New Style

Question # 107: Threadpool create and reduce real threads using hillclimbing algo to _____.

- **To maximize CPU usage** **Pg#160**
- To reduce the cost of thread creation
- To maximize memory utilization
- To minimize CPU usage

threadpool creates or reduces real threads using a hillclimbing algo to maximize cpu usage and reduce slicing.

Question # 108: Which of the following is not handled by the "Task Parallel Library (TPL)"?

- Partitioning of the work
- **Progress report handling** **Click here for more detail**
- Scheduling of threads
- Scales the degree of concurrency

The TPL scales the degree of concurrency dynamically to most efficiently use all the processors that are available. In addition, the TPL handles the partitioning of the work, the scheduling of threads on the ThreadPool.

Question # 109: "Callback" method is used to _____ the operation.

- **Terminate**
- Pause
- Invoke
- Revoke

Question # 110: While dealing with "threads" in ".Net", one can avoid performance bottlenecks and enhance the overall responsiveness of his / her application by using _____ programming.

- Multithreading programming
- Synchronous programming
- Parallel programming
- **Asynchronous programming** **Click here for more detail**

You can avoid performance bottlenecks and enhance the overall responsiveness of your application by using **asynchronous programming.** However, traditional techniques for writing asynchronous applications can be complicated, making them difficult to write, debug, and maintain.

Question # 111: Interact with web page layout

- Java script
- **HTML**
- XML
- DOM

Question # 112: _____ are high order functions that compose, combine, or otherwise modify functions in useful and interesting ways.

- **Combinators**
- None of the given
- Separators
- Modifiers

Question # 113: Which of these are defined as the model object?

- UILabel *questionField;
- NSMutableArray *questions@field;
- **NSMutableArray *questions;** **PG # 205**
- Question= 0;

Question # 114: Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from top edge of the Monitor?

- **ScreenY** **PG#187**
- pageX
- pageY
- ScreenX

screenY	The distance (in pixels) of the mouse pointer from the top edge of the monitor.

Question # 115: What is "nil" like?

- **Null** **Page # 197**
- zero
- terminate
- move to new line

nil is like null.

Question # 116: Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from top edge of the browser window?

- **pageY** **Page # 187**
- pageX
- screenX
- screenY

pageY	The distance (in pixels) of the mouse pointer from the top edge of the browser window.

Question # 117: Observe the following piece of code taken out from Main() function: Thread myThread = new Thread (Go); myThread.Strat(); myThread.Join(); What will be the effect of instruction myThread.Join()?

- Execution of myThread will be joined with existing running threads
- myThread will be executed after Main thread is completed
- **Main thread execution will be suspended till completion of myThread**
- Both Main thread and myThread can now share the data

Question # 118: What message is used to send to destroy the object?

- stop
- delete
- free
- **release**

Question # 119: What is NSMutableArray?

- Object
- **Class Reference**
- Method
- Group

Question # 120: AJAX stands for _____.

- Abstract JSON and XML
- Asynchronous JavaScript and XHtml
- **Asynchronous JavaScript and XML** **Click here for more detail**
- Abstract Java and XML Library

Question # 121: What is the purpose of the following "JQuery" code? $(‘#navbar a’)

- Find all the element whose name is navbar and then find a descendent a
- Find all the element whose name is navbar and then find a ancestor a
- **Find all the element whose ID is navbar and then find a descendent a**
- Find all the element whose ID is navbar and then find a ancestor a

Question # 122: The pairing of labels and arguments is an important feature of _____.

- C++
- Scripts
- **Objective-C** **page # 197**
- PHP

Question # 123: Web servers receive request and responds as _____.

- Html only
- Plan Text only
- JSON only
- **Html, plain text and JSON** **Page #192**

web server: receives request and responds as **HTML, plain text, XML, JSON. or application server** for more complicated tasks. need web server for ajax examples.

Question # 124: Which of the following class supports data parallelism in "Task Parallel Library"?

- System.Threading.Tasks
- System.Task.Parallel
- **System.threading.Task.Parallel** **Click here for more detail**
- System.Task.Threading.Parallel

Task Parallel Library (TPL) supports data parallelism through the System.Threading.Tasks.Parallel class.

Question # 125: Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from left edge of the Monitor?

- pageX
- pageY
- ScreenY
- **ScreenX** **PG# 187**

screenX	The distance (in pixels) of the mouse pointer from the left edge of the monitor.

Question # 126: which of the following response show file not found error?

- **404** **Click here for more detail**
- 500
- 200
- 304

IMPORTANT EVENT PROPERTY AND DESCRIPTIONS:

Event property	Description
pageX	The distance (in pixels) of the mouse pointer from the left edge of the browser window.
pageY	The distance (in pixels) of the mouse pointer from the top edge of the browser window.
screenX	The distance (in pixels) of the mouse pointer from the left edge of the monitor.
screenY	The distance (in pixels) of the mouse pointer from the top edge of the monitor.
shiftKey	Is *true* if the shift key is down when the event occurs.
which	Use with the keypress event to determine the numeric code for the key that was pressed (see tip, next).
target	The object that was the "target" of the event—for example, for a *click()* event, the element that was clicked.
data	A jQuery object used with the *bind()* function to pass data to an event handling function (see page 177).

Note: Give me a feedback and your Suggestion also If you find any mistake in mcqz plz inform me Via Contact us Page on our Site. And tell me your answer with references.

For More Solved Papers By Arslan Visit Our Website:
Www.VirtualUstaad.blogspot.com

☺😐☹ **MUHAMMAD FAISAL** ☺😐☹

MIT 4th Semester

Al-Barq Campus (VGJW01) Gujranwala

faisalgrw123@gmail.com

CS411 Mega OBJECTIVE File for Final Term Papers

CS411 – VISUAL PROGRAMMING

Question No: 1 (Marks: 1) - Please choose one

A\An _____ displays the map and the labels for the recorded locations.

- **mkmapview** **(Page#213)**
- uiactivityindicatorview
- uitextfield
- mapmkview

Question No: 2 (Marks: 1) - Please choose one

_____ properties based on input or mouse clicks.

- HTML
- **CSS** **(Page#180)**
- C#
- XML

Question No: 3 (Marks: 1) - Please choose one

Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from left edge of the Monitor?

- pageX
- pageY
- ScreenY
- **ScreenX** **(Page#205)**

Question No: 4 (Marks: 1) - Please choose one

_____ was introduced in 95 by netscape.

- C ++
- . Net
- **Javascript** **(Page#179)**
- PHP

Question No: 5 (Marks: 1) - Please choose one

Web servers receive request and responds as _____.

- Html only
- Plan Text only
- JSON only
- **Html, plan text and JSON** **(Page#211)**

Question No: 6 (Marks: 1) - Please choose one

Concurrent collections may not useful when you want a thread-safe collection three static methods in the Parallel class.

True
False **(Page#175)**

Question No: 7 (Marks: 1) - Please choose one

We can return a _____from void function without explicitly return it.

- value
- event
- **task** **(Page#169)**
- function

Question No: 8 (Marks: 1) - Please choose one

The pairing of labels and arguments is an important feature of _____.

- C++
- Scripts
- **Objective-C** **(Page#217)**
- PHP

Question No: 9 (Marks: 1) - Please choose one

Thread pool save time of thread _____.

- insertion
- **creation** **(Page#160)**
- deletion
- updation

Question No: 10 (Marks: 1) - Please choose one

What is NSMutableArray?

- Object
- **Class Reference** **(Page#217)**
- Method
- Group

Question No: 11 (Marks: 1) - Please choose one

In the _____, the target is updated whenever the source changes.

- Binding Object
- **Binding Mode** **(Page#153)**
- Data Mode
- Data Object

Question No: 12 (Marks: 1) - Please choose one

Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from top edge of the Monitor?

- **ScreenY** **(Page#205)**
- pageX
- pageY
- ScreenX

Question No: 13 (Marks: 1) - Please choose one

Without binding you could not insert custom logic.

True
False **(Page#153)**

Question No: 14 (Marks: 1) - Please choose one

What is “nil” like?

- **Null** **(Page#217)**
- zero
- terminate
- move to new line

Question No: 15 (Marks: 1) - Please choose one

Sorting is applied _____ grouping.

- at initial stage
- anytime
- **before** **(Page#150)**
- after

Question No: 16 (Marks: 1) - Please choose one

Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from top edge of the browser window?

- **pageY** **(Page#205)**
- pageX
- screenX
- screenY

Question No: 17 (Marks: 1) - Please choose one

Prop. Paths in Bindings are useful for master/detail interfaces.

True **(Page#150)**
False

Question No: 18 (Marks: 1) - Please choose one

Objective-C introduces _____ messaging in C.

- **Smalltalk style** **(Page#217)**
- Short sessions
- Group
- New style

Question No: 19 (Marks: 1) - Please choose one

Template can be shared as _____.

1) data items
2) events
3) **resources** **(Page#144)**
4) files

Question No: 20 (Marks: 1) - Please choose one

Objective – c introduces _____ messaging in c?

1) **Smalltalk style** **(Page#217)**
2) Short session
3) Group
4) New Style

Question No: 21 (Marks: 1) - Please choose one

Navigation means managing the _____ item.

1) initial
2) last
3) **current** **(Page#150)**
4) any

Question No: 22 (Marks: 1) - Please choose one

Threadpool create and reduce real threads using hillclimbing algo to _____.

1) **To maximize CPU usage** **(Page#174)**
2) To reduce the cost of thread creation
3) To maximize memory utilization
4) To minimize CPU usage

Question No: 23 (Marks: 1) - Please choose one

_____ style is a rule telling web browser how to display an element.

- HTML
- Java
- C#
- **CSS** **(Page#180)**

Question No: 24 (Marks: 1) - Please choose one

HTML is a _____ helper function.

- RQuery
- Browser
- **jQuery** **(Page#201)**
- JS

Question No: 25 (Marks: 1) - Please choose one

Today websites reach interactivity of _____applications.

- web based
- **desktop** **(Page#179)**
- both of above
- none of given

Question No: 26 (Marks: 1) - Please choose one

_____ is more than one thing happening at the same time.

- Cohesion
- Congruence
- Coherence
- **Concurrency** **(Page#169)**

Question No: 27 (Marks: 1) - Please choose one

Long running operations make application _____.

- implicit
- responsive
- **unresponsive** **(Page#160)**
- explicit

Question No: 28 (Marks: 1) - Please choose one

Which of the following "Event "properties is used with "key press event" to determine the numeric code for the key that was pressed? .

- Page X
- **Which** **(Page#205)**
- Shift Key
- Target

Question No: 29 (Marks: 1) - Please choose one

Thread-safe collection guaranteed the thread safe code.

True
False **(Page#177)**

Question No: 30 (Marks: 1) - Please choose one

MS introduced jscript. Their version of JS for IE.

True **(Page#179)**
False

Question No: 31 (Marks: 1) - Please choose one

Jquery solves _____ complexity and browser incompatibilities.

i. . Net
ii. C
iii. **JS** **(Page#179)**
iv. C#

Question No: 32 (Marks: 1) - Please choose one

Difference between Break and Stop is that Break reaches at least the sequential point.

True **(Page#175)**
False

Question No: 33 (Marks: 1) - Please choose one

Sorting criteria should be same as grouping otherwise output make sense.

True
False **(Page#150)**

Question No: 34 (Marks: 1) - Please choose one

_____ is more than one thing happening at the same time.

i. Cohesion
ii. Congruence
iii. Coherence
iv. **Concurrency** **(Page#169)**

Question No: 35 (Marks: 1) - Please choose one

A\An _____ indicates that the device is working and not stalled.

- mkmapview
- **uiactivityindicatorview** **(Page#213)**
- uitextfield
- mapmkview

Question No: 36 (Marks: 1) - Please choose one

We can join and use shared data.

True **(Page#161)**
False

Question No: 37 (Marks: 1) - Please choose one

_____ solves "JavaScript" complexity and browser incompatibilities.

- Jhtml
- RQuery
- **JQuery** **(Page#195)**
- J#

Question No: 38 (Marks: 1) - Please choose one

Template can even be auto-applied to some type by setting its datatype prop.

True **(Page#144)**
False

Question No: 39 (Marks: 1) - Please choose one

We use _____ to avoid compiler errors.

- data
- **xdata** **(Page#150)**
- ydata
- none of the given

Question No: 40 (Marks: 1) - Please choose one

_____Can add/remove/change "CSS" properties based on input or mouse clicks.

- J#
- JQuery
- HTML
- **None of the given option** **(Page#196)**

Question No: 41 (Marks: 1) - Please choose one

_____ is typing together arbitrary objects.

- data frame
- data store
- **data binding** **(Page#140)**
- markup extension

Question No: 42 (Marks: 1) - Please choose one

_____ Mode of binding is opposite to OneWay mode of binding.

- **OneWay To source** **(Page#165)**
- TwoWay
- One Time
- Two Time

Question No: 43 (Marks: 1) - Please choose one

Collection view source can be used to create _____ views and applied to targets.

- various
- old
- **new** **(Page#150)**
- initial

Question No: 44 (Marks: 1) - Please choose one

Which of the following "Event" property is used to show the "distance (in pixels)" of the mouse pointer from the left edge of the browser window?

- pageY
- **pageX** **(Page#205)**
- ScreenX
- ScreenY

Question No: 45 (Marks: 1) - Please choose one

When an item with no header selected, then a _____ returned.

- key value
- prime value
- **default value** **(Page#140)**
- numeric value

Question No: 46 (Marks: 1) - Please choose one

Which of the following is not a "JQuery" filter?

- Even
- Odd
- Has
- **Write** **(Page#201)**

Question No: 47 (Marks: 1) - Please choose one

Binding has a _____ prop and a _____ prop.

- source, destination
- **source, target** **(Page#140)**
- target, destination
- none of the given

Question No: 48 (Marks: 1) - Please choose one

Which of the following Object creates a cancellation token?

- OperationCanceledtoken
- OperationCanceled
- CancellationToken
- **CancellationTokenSource** **(Page#186)**

Question No: 49 (Marks: 1) - Please choose one

We can not bind to a collection.

True
False **(Page#141)**

Question No: 50 (Marks: 1) - Please choose one

Threadpool is used to _____.

- To make easy the process of thread creation
- To make easy debugging process
- **To save time of thread creation** **(Page#174)**
- To avoid complexity

Question No: 51 (Marks: 1) - Please choose one

_____ and later merge the journal and provide a more streamlined interface.

- IE 6
- **IE 7** **(Page#132)**
- IE 8
- IE 9

Question No: 52 (Marks: 1) - Please choose one

Methods with async are called asynchronous.

True **(Page#166)**
False

Question No: 53 (Marks: 1) - Please choose one

Exception is raised when an item with no header selected, then a default value returned.

True
False **(Page#140)**

Question No: 54 (Marks: 1) - Please choose one

HTML is called _____ layer.

- Physical
- Presentation
- Behavioral
- **Structural** **(Page#195)**

Question No: 55 (Marks: 1) - Please choose one

Data can be:

- xml file
- web service
- db table
- **all of the given** **(Page#140)**

Question No: 56 (Marks: 1) - Please choose one

We can mix Items and item source.

True
False **(Page#142)**

Question No: 57 (Marks: 1) - Please choose one

_____helps apps remain oblivious to the visual tree.

- Routed Events
- **Input Events** **(Page#116)**
- Output Events
- Obvious Events

Question No: 58 (Marks: 1) - Please choose one

Frame work element and framework content element both have a Resources prop. Often are style or data providers.

True **(Page#135)**
False

Question No: 59 (Marks: 1) - Please choose one

Any assembly marked with allow partially trusted callers and placed in _____ can be called by partial-trust code

- PAC
- SAC
- **GAC** **(Page#132)**
- RAC

Question No: 60 (Marks: 1) - Please choose one

Additional events can be used for fine grained progress.

True **(Page#133)**
False

Question No: 61 (Marks: 1) - Please choose one

Many XBAP take advantage of navigation.

True
False **(Page#132)**

Question No: 62 (Marks: 1) - Please choose one

Routed events are like _____ on top of .Net properties.

- Event properties
- Layer
- Properties
- **Dependency properties** **(Page#116)**

Question No: 63 (Marks: 1) - Please choose one

You can still use rich text and media, isolated storage up to _____, arbitrary files on host web server, use browser file open dialog for local files.

- ♈ 312kb
- ♈ 412kb
- ♈ **512kb** **(Page#132)**
- ♈ 1000kb

Question No: 64 (Marks: 1) - Please choose one

One of the purposes of journal is we use this in photo gallery for undoable image rotation.

True **(Page#129)**
False

Question No: 65 (Marks: 1) - Please choose one

Use embedded resource to define a binary resource.

True
False **(Page#113)**

Question No: 66 (Marks: 1) - Please choose one

Which of the following is Not a routing strategy?

- ♈ Tunneling
- ♈ **Indirect** **(Page#117)**
- ♈ Bubbling
- ♈ Direct

Question No: 67 (Marks: 1) - Please choose one

Navigation can also happen between _____.

- xml files
- **html files** **(Page#128)**
- iso files
- http files

Question No: 68 (Marks: 1) - Please choose one

The _____ statement iterates over each element in an "enumerable" object.

- **Foreach** **(Page#191)**
- Dowhile
- While
- For

Question No: 69 (Marks: 1) - Please choose one

You can get an instance of navigation service by calling the static navigationservice.getnavigationservice method.

True **(Page#128)**

False

Question No: 70 (Marks: 1) - Please choose one

The more the standardization on builtin commands, the more seamless and declarative the interaction can be between controls.

True **(Page#123)**

False

Question No: 71 (Marks: 1) - Please choose one

Windows installer benefits include:

- showing custom setup UI
- arbitrary code at setup time
- register COM components and file associations
- **All of the given options** **(Page#127)**

Question No: 72 (Marks: 1) - Please choose one

Persisting and Restoring can use registry or file system.

True **(Page#127)**
False

Question No: 73 (Marks: 1) - Please choose one

The corresponding type converter class for "Brush" class will be _____.

- **BrushConverter** **(Page#149)**
- CastBrush
- BrushCast
- None of the given options

Question No: 74 (Marks: 1) - Please choose one

Button and textbox have direct knowledge of each other.

True
False **(Page#123)**

Question No: 75 (Marks: 1) - Please choose one

_____ provides logic behind back and fwd.

- Hyperlink
- **Journal** **(Page#129)**
- Container
- Navigation

Question No: 76 (Marks: 1) - Please choose one

App code-behind can not be omitted altogether.

True
False **(Page#125)**

Question No: 77 (Marks: 1) - Please choose one

WPFS controls have built-in behavior tied to various commands.

True **(Page#121)**
False

Question No: 78 (Marks: 1) - Please choose one

Stylus can behave like a _____ but has _____resolution.

- Keyboard, lower
- Mouse, lower
- **Mouse, higher** **(Page#122)**
- Keyboard, higher

Question No: 79 (Marks: 1) - Please choose one

Translation behavior has desired displacement, desireddeceleration, and initialvelocity.

True (Page#117)
False

Question No: 80 (Marks: 1) - Please choose one

Rotation behavior has desiredexpansion, desireddeceleration, initialradius, and initialvelocity.

True
False (Page#117)

Question No: 81 (Marks: 1) - Please choose one

_____ defines a number of built-in commands.

- OSPF
- **WPF (Page#121)**
- SPF
- PWF

Question No: 82 (Marks: 1) - Please choose one

By convention "tunneling event" names are prefixed with _____ and just come_____ the bubbling event occurs.

- View, after
- View, before
- **Preview, before (Page#117)**
- Preview, after

Question No: 83 (Marks: 1) - Please choose one

Commands have automatic support for _____ gestures.

- **input** **(Page#121)**
- output
- both input & output
- none of the given

Question No: 84 (Marks: 1) - Please choose one

Any number of child windows can be made by instantiating a Window derived class and calling Show.

True **(Page#124)**
False

Question No: 85 (Marks: 1) - Please choose one

Translation behavior has desired rotation, desired deceleration, and initial velocity.

True
False **(Page#117)**

Question No: 86 (Marks: 1) - Please choose one

All instances of routeduicommand implement icommand and support bubbling.

True **(Page#121)**
False

Question No: 87 (Marks: 1) - Please choose one

_____ are a more abstract and loosely coupled version of events.

- values
- **commands** **(Page#121)**
- functions
- numbers

Question No: 88 (Marks: 1) - Please choose one

Which "transform" property can help us to flip the element from its center?

- ScaleY
- ScaleX
- **RenderTransformOrigin** **(Page#126)**
- None of given options

Question No: 89 (Marks: 1) - Please choose one

Manipulation _____starting event when all fingers loose contact even before a completed event.

- event
- container
- **inertia** **(Page#117)**
- function

Question No: 90 (Marks: 1) - Please choose one

Page does everything window does except:

- uploading and downloading
- **onclosed and onclosing** **(Page#128)**
- onclosed and navigating
- navigating and onclosing

Question No: 91 (Marks: 1) - Please choose one

_____behavior has desired rotation, desired deceleration, and initial velocity.

- **Rotation** **(Page#117)**
- Translation
- Expansion
- None of the given

Question No: 92 (Marks: 1) - Please choose one

An "UIElement" receives mouse events only if "IsHitTestVisible" is _____.

- 0
- False
- **Null** **(Page#121)**
- True

Question No: 93 (Marks: 1) - Please choose one

By default, initialvelocity and initialradius are initialized with the _____ values.

- coming
- **current** **(Page#117)**
- multiple
- significant

Question No: 94 (Marks: 1) - Please choose one

Starting and started events _____to customize or cancel or disallow some manipulations.

- stop
- end
- not allowed
- **allow** **(Page#116)**

Question No: 95 (Marks: 1) - Please choose one

With navigation, content is usually in _____a simpler version of Window.

- Frame
- **Page** **(Page#128)**
- Box
- Button

Question No: 96 (Marks: 1) - Please choose one

Manipulations always done relative to a manipulation _____.

- event
- **container** **(Page#117)**
- inertia
- function

Question No: 97 (Marks: 1) - Please choose one

Modal dialogs include _____which are actually provided by win32.

- interfaces
- **common dialogs** **(Page#126)**
- designs
- structure

Question No: 98 (Marks: 1) - Please choose one

_____ behavior has desiredexpansion, desireddeceleration, initialradius, and initialvelocity.

- Rotation
- Translation
- **Expansion** **(Page#117)**
- None of the given

Question No: 99 (Marks: 1) - Please choose one

A Page can interact with its navigation container by using _____, which exposes relevant functionality regardless of whether the container is a navigation window or a Frame.

- navigation class
- **navigation service class** **(Page#128)**
- navigation page class
- navigation box class

Question No: 100 (Marks: 1) - Please choose one

One _____Thread and one render thread can create more with Dispatcher.Run which can improve responsiveness.

- EU
- **UI** **(Page#125)**
- IU
- IE

CS411

VISUAL PROGRAMMING

Final Term Quizzes MCQ's

1) While Manipulation, we combine information from multiple _____.

functions
events **(Page#116)**
integers
numbers

2) In Xaml browser application, how much isolated memory we can use?

256kb
128
512
1024

3) _____ behavior has desired displacement, desireddeceleration, and initialvelocity.

Rotation
Translation **(Page#117)**
Expansion
None of the given

4) AJAX stands for _____.

Abstract JSON and XML
Asynchronous JavaScript and XHtml
Asynchronous JavaScript and XML
Abstract Java and XML Library

5) Which of the following technologies is not used in AJAX?

DOM
DHTML
Flash
CSS

6) To be boundary aware manipulationboundaryfeedback _____ is used.

event **(Page#118)**
container
inertia
function

7) Which of these are defined as the model object?

UILabel *questionField;
NSMutableArray *questions@field;
NSMutableArray *questions;
Question= 0;

8) Which of the following is correct way to temporarily stop execution of a thread object named as “myThread” for 500 milisec?

myThread.Sleep(0.50);
myThread.Stop(0.05);
myThread.Sleep(500);
myThread.Stop(500);

9) “async” and _____ keywords in C# are the heart of asynchronous programming.

Return
Await
Wait
String

10) Controls have logic to _____ with commands through Command property.

design
interface **(Page#121)**
structure
none of the given

11) Which of the following C# Method reports a progress change?

Finalize
OnReport
GetType
GetHashCode

12) Which of the followings are keyboard events?

Key entered, key exist
Key strike, key release
Key down, key up
Key pressed, key released

13) In case of HTML page navigation, you must use the _____ of "Navigate" that _____ a URI.

Overload, rejects
Overload, accepts
IFrame, rejects
IFrame, accepts

14) Which one of the following can be used to set dependency property values?

Temporary resource
Permanent resource
Dynamic resource
Static resource

15) Child window like parent window but gets closed when parent and similarly minimized, also called _____.

modeless area
modeless dialog **(Page#124)**
modeless interface
modeless design

16) "Binding" technique binds two properties together and keeps a _____ open.

Memory location
Communication channel
Selection
Choice

17) A/An _____ screen is an image that appears while a game or program is loading

Opening
Startup
Initial
Splash

18) If we insert some objects in a single cell in Grid panel, then these objects are placed on _____.

One on Top of the other
One after the other
One on Bottom of the other
One on Side of the other

19) In a code segment, if the Visibility=”Collapsed” then which of the following events are not generated?

Routed Events
Keyboard Events
Key down Events
Mouse Events

20) Which of the following operation cannot be performed by using a View?

Sorting
Grouping
Searching
Filtering

21) lickonce benefits include:

builtin support for automatic updates
rollback to prev versions
both of above options **(Page#127)**
none of the given

22) In order to resize the row and column in Grid panel, _____ is used.

Grid converter
Grid splitter
Grid divider
Grid compiler

23) _____and _____ are the two important properties of Binding object.

Start, Destination
Items, Selected
Source, Path
BindTo, BindFrom

24) In multi touch events, a/an _____ id is assigned to each individual event.

Device
Event
CPU
Processor

25) Which of the following operation cannot be performed by using a View?

Filtering
Sorting
Grouping
Searching

26) Page also contains a few of its own properties that control the behavior of the _____.

container
parent container **(Page#128)**
child container
none of the given

27) Using _____ requires more overhead than _____ because of the extra tracking.

StaticResource, DynamicResource
PermanentResource, TemporaryResource
TemporaryResource, PermanentResource
DynamicResource, StaticResource

28) Which function is used to create cancel button event of dialog box _____.

OnClickCancel()
OnInitCancel()
OnCancel()
CancelClick()

29) Which of the following response show file not found error?

404
500
200
304

30) _____ are high order functions that compose, combine, or otherwise modify functions in useful and interesting ways.

Modifiers
Separators
Combinators
None of the given

31) Hyperlink element is used to link to _____.

html
xaml **(Page#129)**
http
website

32) "JavaScript" is _____ scripting Language.

Server
Client
Interpreter
Browser

33) The Java programming language has a specific class for creating splash screens, called _____.

Java.awt.start
Java.lang.splash
Java.start.screen
Java.awt.SplashScreen

34) To remove data binding between the source and the target, we can use _____ function.

StopBinding
RemoveBinding
ClearBinding
ResetBinding

35) You get Integrated Navigation in _____.

XAML
SDK
XBAP **(Page#132)**
HTML

36) In multi touch events, when multiple fingers are touching simultaneously, these events get raised for each finger _____.

Completely
Partially
Independently
Dependently

37) The _____ user interfaces, enabled by WPF, is getting a lot of attention.

Loutish
Polished
Coarse
Distracted

38) When your C# program is compiled, it creates a file called a/an_____, which is normally an executable or DLL library.

BIOS
C++
Assembly
Text

39) _____where multiple GUI elements and many sources of events exist.

Control Language Interface
Content property
Event properties
Visual programming

40) In Visual Studio 2012, expression blend is _____ WPF feature.

80%
70%
100%
90%

41) To define a binary resource, add a _____ and set the build action resource or content (loose file).

folder
file **(Page#133)**
icon
page

42) A (n) _____ is a component of the structure of an event.

Event Architecture
Event Request
Event Attribute
Event Response

43) A _____ block has access to an "Exception" object that contains information about the error.

Try
Catch
Throw
Finally

44) One of the ways to create an instance of a class is _____.

Sequential heap
System array
Factory method
XAML compiler

45) There is/are _____ type/s of parameter/s for attributes.

Four
Three
One
Two

46) A (n) _____ is a set of associated events.

Event Consumer
Event Producer
Event Channel
Event Stream

47) If we require to set components in table (column and row) format, then which panel will be used?

StackPanel
DockPanel
Gridpage
TabPanel

48) _____ Method is used to move the file from one directory to another.

Import
Move
Export
None of given options

49) Not implementing some methods of the interface in the class is _____.

A warning
An exception
An error
Acceptable

50) Logical resources are introduced by wpf. Arbitrary .net object stored and named in an elements Resources prop. Typically meant to be shared by multiple _____ objects.

parent
child
event **(Page#135)**
function

51) A (n) _____is a subscription mechanism for events.

Raw Event
Event Stream
Event Channel
Event Producer

52) "Situation" is an event occurrence that requires a (n) _____.

Class
Object
Reaction
Action

53) XML is a stricter version of _____.

CSS
C#
Java
HTML

54) _____ contacts the service provider or the server.

Event-driven
Service Requester
Both of the above
None of the given options

55) Any class, including the same class that the event is declared in, may register one of its methods with the _____.

Event
Delegate
Function
Class

56) _____ data source is provided by a Data Context.

Explicit
Implicit **(Page#142)**
Value
Context

57) _____ decouple producers and consumers.

Requests
Events
Functions
All of these

58) To declare an anonymous method, you just use keyword "_____".

handler
event
delegate
none of the given

59) The common type system of C# has _____ types and _____ types.

Reference, Value
Value, Reference
Function, Data
Data, Function

60) There are _____ methods to read XML document.

Two
Three
Four
Five

61) Collectionviewsource has its own sortdescriptions and groupdescriptions properties and a Filter event to be used from _____.

event
container
xaml **(Page#150)**
xtml

62) _____ can leave your program in an inconsistent state by not releasing resources or doing some other type of cleanup.

Functions
Exception
Abstraction
All of these

63) _____ are called by garbage collector.

Constructors
Destructors
Both of above
None of the given

64) Attributes are generally applied _____ in front of type and type member declarations.

Logically
Physically
Both of above
None of given

65) Multiple _____ are conventionally stored in multiple files.

Classes
Values
Variables
Functions

66) The _____, _____ and _____of applications that use events, either directly or indirectly is called event-based programming.

Model, Code, Operation
Analyze, Design, Operation
Design, Coding, Operation
Code, Operation, Maintain

67) C# is a _____ and .NET is a _____.

Platform, Language
Language, platform
Package, Language
Language, Package

68) Windows Presentation Foundations (WPF) was publicly announced in _____.

2000
2001
2002
2003

69) WPF stands for:

Windows Powerful Foundations
Windows Presentation Formations
Windows Presentation Functions
Windows Presentation Foundations

70) _____ declaratively instantiate the source object with a parameterized constructor.

data object
.net object **(Page#152)**
binding object
files object

71) Value-types hold their value in _____ where they are declared.

Variable
Memory
Object
Reference

72) Indexers can take any number of _____.

Arrays
Variables
Parameters
All of these

73) Attributes add _____to year program.

Clearance
Errors
Meta Data
Ambiguity

74) Cast back to base type to call a/an _____ method of base.

Instance
Base Class
Overridden
Derived Class

75) _____ is said to be stateless if the way it processes one event does not influence the way it processes any subsequent events.

Event Producer
Raw Event
Event Processing
Event Stream

76) _____ is built from request-response and it moves away from monolithic applications.

Event-driven architecture (EDA)
Service-oriented architecture (SOA)
Both of above
None of the given option

77) Object variables are references to the _____ object not the object themselves.

Original
Alternative
Primary
None of the given

78) An indexer enables your class to be treated like a/an _____.

Class
Function
Array
Pointer

79) Polymorphism needs the signatures to be the _____.

Different
Same
Alternative
None of the given

80) In the Binding Mode, the source is _____whenever the target changes.

remain same
updated **(Page#153)**
revised
expired

81) Types of class members in C# are _____ and _____.

ostatic, instance
oinstance, static
opublic, private
ostatic, private

82) Events may live _____ events like in event logs.

Inside
Outside
Both of above
None of the given

83) In event-driven architecture (EDA), _____ sends event to _____.

Event Consumer, Event Producer
Event Producer, Event Consumer
Event Handler, Event Producer
Event Generator, Event Producer

84) Additional XML namespaces (on the root or on children) must be _____ to be used on any identifiers from that namespace.

Given a common prefix
Qualified with “Pre” keyword
Given a distinct prefix
Qualified with “Ext” keyword

85) Thread pool creates or reduces _____threads using a hill climbing algo to maximize cpu usage and reduce slicing.

render
real **(Page#160)**
STA thread
none of the given

86) The interface forces each component to expose specific _____ members that will be used in a certain way.

Public
Private
Both of above
None of the given

87) _____ allows to keep the user interface description and implementation separate.

GDI
WPF
Avalon
Altia

88) Dependency properties are represented by _____.

System.Windows.DependencyProperty
System.Windows.Dependency.Property
System.Windows.Dependency
System.Windows.Property.Dependency

89) Suppose we want to restrict the user to don‟t decrease the height of a WPF element from a specific amount then which property will be used?

HeightMinimum
MinimumHeight
MinHeight
None of the given

90) _____indirectly uses task completion source to implement methods returning Tasks.

Debugger
Linker
Compiler **(Page#169)**
Loader

91) If we want to scale an object in the 2-D x-y coordinate system then which transform class will be used?

Rotate Transform
Matrix Transform
Translate Transform
Skew Transform

92) Which of the following is/are not derived from "Control" class?

ListBox
StatusBar
Button
None of the given

93) Wait for a single event is _____operation.

Waiting
Waste
Blocking
Idle

94) _____ method reads the whole document in memory.

XPath
XmlDocument
XmlReader
XmlLine

95) Which of the following events may or may not relate to an actual occurrence?

Probabilistic Events
Actual Events
Real Events
Expected Events

96) The System.Console is a _____ which enables us to do Console Input and Output.

Structure
Class
Namespace
Library

97) Clean room design is useful as a defense against _____.

Copyright
Trade secret infringement
Copyright and trade secret infringement
None of the given options

98) Clean room design is also called_____.

Clear Room Design
Wall Technique Design
Chinese Wall Technique
French Wall Technique

99) Which of the following is an occurrence within a particular system or domain?

Object
Event
Result
Message

100) TAP abbreviated as:

Task-based Async Plan
Task-based Async Pattern **(Page#173)**
Task-based Async Program
Task-based Async Preview

101) During the program execution, program can be stopped with the help of _____.

Attributes
Compiler
Breakpoints
Exception Handling

102) Which of the following does NOT include in event processing operations?

Deleting Events
Reading Events
Terminating Events
Transforming Events

103) The result of Exception can bring _____ in the program.

Reliability
Stability
Inconsistency
Consistency

104) Graphical user interfaces and the event-driven model is not applied on which of the following?

Web Application
Console Application
Mobile Application
Desktop Application

105) Events are based on the principle of _____.

Request
Response
Coupling
Decoupling

106) _____methods do not have names.

Attribute
Non-Anonymous
Anonymous
None of given

107) Which of the following is an entity that introduces event into the system?

Event Channel
Event Producer
Event Stream
Event Consumer

108) Anonymous methods _____ the code size.

Double
Expand
Reduce
Extend

109) _____ events may or may not relate to an actual occurrence

Layer
Properties
Event properties
Probabilistic

110) Task Parallel _____exploit multicore for real parallel tasks.

Structure
Library **(Page#175)**
Class
Namespace

111) The Main method returns a non-zero value which indicates the_____.

Source of Program
Error in Program
Termination of Program
Exception in Program

112) WPF4 was released in _____.

2012
2010
2005
2003

113) Extension methods can only be defined in _____ class.

Static class
Dynamic class
Normal class
String class

114) AML is compressed representation of _____.

CAML:NS
XAML
XAML:NS
CAML

115) Question # 56: _____ processing operations include reading, creating, transforming, and deleting events.

Observed Events
Common event
Deducted Events
Probabilistic Events

116) Static ctor exists to initialize class static members and called only _____.

Once
Twice
Anytime
All of the given

117) The _____ in the namespace indicate a hierarchy of nested namespaces.

Arrows
Dots
Semicolon
Brackets

118) _____ Interact with web page layout.

Java script
HTML
XML
DOM

119) Which of the following response show internal server error?

404
500
200
304

120) _____ is a formatting language.

HTML
CSS **(Page#180)**
Java
C#

121) Which of the following class supports data parallelism in "Task Parallel Library"?

System.Threading.Tasks
System.Task.Parallel
System.Task.Threading.Parallel
System.threading.Task.Parallel

122) What is the purpose of the following "JQuery" code? $(„#navbar a‟)

Find all the element whose name is navbar and then find a descendent a
Find all the element whose ID is navbar and then find a descendent a
Find all the element whose name is navbar and then find a ancestor a
Find all the element whose ID is navbar and then find a ancestor a

123) What message is used to send to destroy the object?

stop
delete
free
release

124) While dealing with "threads" in ".Net", one can avoid performance bottlenecks and enhance the overall responsiveness of his / her application by using _____ programming.

Multithreading programming
Asynchronous programming
Synchronous programming
Parallel programming

125) DOM is a _____ representation of data.

Tree
Graphical
Object
None of the given

126) "Callback" method is used to _____ the operation.

Invoke
Revoke
Terminate
Pause

127) Which of the following is not handled by the “Task Parallel Library (TPL)”?

Partitioning of the work
Scales the degree of concurrency
Scheduling of threads
Progress report handling

128) Why we use “Alloc” message?

To create a class
To create an instance
To extend functions
To inherent methods

129) Which of the following server(s) support “AJAX”?

WWW
SMTP
HTTP
Both SMTP and HTTP

130) Which of the following is not a part of "cancellation" task?

The calling thread does forcibly end the task
Notice and respond to the cancellation request in your user delegate
Pass a cancellation token to your user delegate and optionally to the task instance
Create and start a cancelable task

131) Jquery simplifies all steps except that of changing the webpage simplest is load function which loads _____ into an area of web page.

Function
Library
HTML **(Page#193)**
XAML

132) The "Task Based Asynchronous pattern" method returns either a "Task" or a "Task<TResult>", based on whether the corresponding _____ method returns "void" or a type "TResult".

Callback
Synchronous
Asynchronous
Return

133) "JavaScript" _____is available in web browser.

Compiler
Linker
Interpreter
Conjunction

134) Which of the following is not true about task completion source?

it is executable thing **(Page#175)**
Provide Continuation facility
Support result return facility
Provide exception handling facility

135) A class that implements the „lCommand‟ and supports „bubbling‟ just like a routed event is known as _____.

Executed command
CanexecuteChange
RoutedUlCommand
Canexecute

136) To plug custom logic, you need to add a _____ to the element that will execute the command or any parent element.

Execute command
Helpcanexecute
CommandBinding
RoutedUlCommand

137) In case of integrating Navigation to XBAP, "ShowNavigationUI" should be set to _____ in order to bypass integration.

True
Null
False
0

138) We use "GetDefaultView" method of _____ to get the default View.

DataSource
ViewDescription
ViewCollection
CollectionViewSource

139) Which one of the following is used, if we want to add a window resource in procedural code?

Window.Resource.Add ();
Window.Resoruces.Add();
<Widows.Resources></Widows.Resources>
<Window.Resources></Window.Resources>

140) We can perform sorting through View by using an object of_____ class.

SortOrder
ViewDescription
SortDescription
ViewOrder

141) Question # 261: A _____ receives keyboard events only if it has keyboard focus.

Keyboard Controls
UIElement
Mouse
Keyboard

142) Frame has a _____ property used in enabling or disabling the bar.

NavigationUIBlock
ShowsNavigationUI
BlocknavigationUI
NavigationUIVisibility

143) A Window can spawn _____ number of additional Windows by instantiating a Window - derived Class and calling Show (...).

Four
Six
Five
Any

144) _____ provides the logic behind the "Back" and "Forward" buttons.

Hyperlink
IFrame
Journal
Navigation Window

145) A\An _____ allows the user to input text to label the current location on the map.

mkmapview
uiactivityindicatorview
uitextfield **(Page#213)**
mapmkview

146) In _____ browser app, you change"<TargetZone>Internet</TargetZone>" to "<TargetZone>Custom</TargetZone>"

Partial - reject
Full - trust
Partial - trust
Full - reject

147) To check that the user either pressed left Alt key or right Alt key, the _____ is used.

IsDown
KeyStates.IsKeyDown
IsKeyDown
keyboardDevice.IsKeyDown

148) _____ supports the creation of applications that run directly in a web browser.

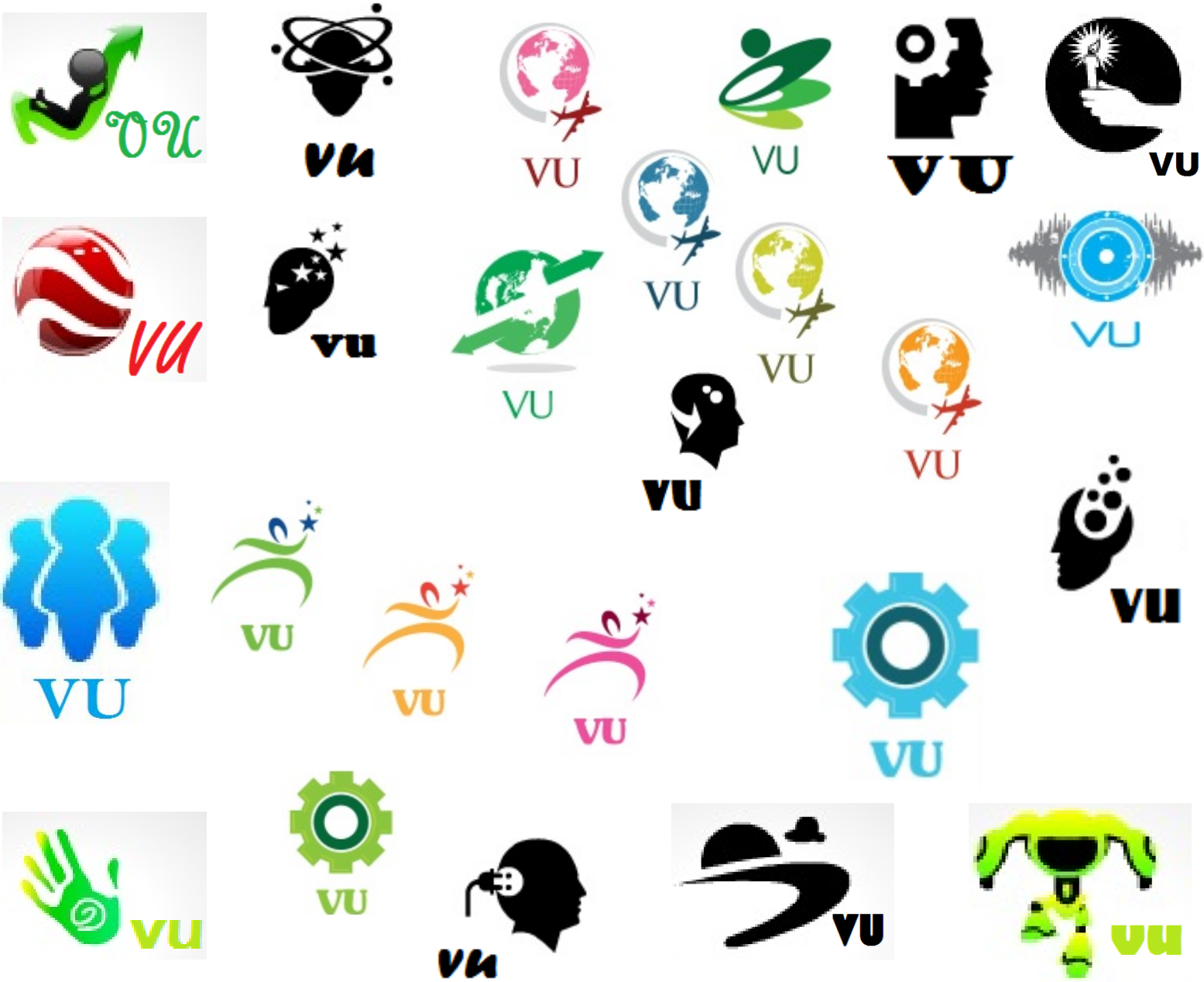
C#
WPF
HTML
C/C++

149) Which one of the following panels is most powerful, versatile and customizable?

Doc panel
Stack panel
Grid Panel
Canvas panel

150) Touch events are hallmark of _____.

computers
laptops
mobile devices **(Page#216)**
none of the given



CS411 More MCQS

Final Term Quizzes

Visual Programming

CS411 - Visual Programming

Protocols are like _____ in other languages.

web pages
interfaces **(Page#212)**
structures
codes

Data binding is about tying together arbitrary .NET _____.

Functions
Properties
Objects
Variables

An event that is raised whenever the value of "CanExecute" changes is known as _____.

Command
Execute
CanExecute
CanExecuteChanged

Multi touch events are categorized into _____ and _____.

Touch events, touchup events
Touchdown events, touchup events
Basic touch events, higher-level manipulation events
Advanced touch events, low-level manipulation events

In context of data binding, _____ contains the current item to get it synchronized with data Source.

String
Template
List
View

Which of the following is an example of tunnel key event?

Key up
Preview key entered
Preview key down
Key down

_____ has simple commands called tags.

JAVA
CSS
C#
HTML **(Page#179)**

Commonly we have two types of custom command bindings named as _____.

StartBindings and StopBindings
KeyBindings and MouseBindings
EventBindings and InstanceBindings
ButtonBindings and TextboxBindings

Which of the following is not a string format property throughout WPF?

ItemStringFormat
StringFormat
ContentStringFormat
RowHeaderStringFormat

To design a complex Interface, which of the following technique is best to achieve the required results?

Panels composed in three panels
Panels composed in two panels
Panels composed within panels
Panels composed in single panel

_____ is used to perform navigation.

Session
Hyperlink
Internet protocol
Hypertext markup language code

The Java programming language has a specific class for creating splash screens, called _____.

Java.awt.start
Java.lang.splash
Java.start.screen
Java.awt.SplashScreen

We use "UIElement" property ClipToBounds= “false” when _____.

We want area child element don‟t cross the boundary of parent
We want area of child element is allowed to cross the boundary of parent
Parent area also increases on increasing of area of child element
None of given

In Parallelism true concurrency at _____level operations.

upper
bottom **(Page#171)**
high
lower

Consider the following C# code segment: button.Background =
(Brush) new BrushConverter().ConvertFrom (“SystemColors.WindowBrush”);
Which one of the following is equivalent XAML code?

Button Background= “SystemColor.WindowsBrush”/>

Button Background= “SystemColor.WindowBrush”/>

Button Background= “SystemColors.WindowsBrush”/>

Button Background= “SystemColors.WindowBrush”/>

All elements of WPF have following property/properties.

Width
Height
Height and Width
None of given options

With the help of WPF, an ugly looking application can be _____ by the designers.

Rearranged
Re-created
Rejected
Re-themed

Which of the following components generates an input Event?

Printer
Plotter
Multi-Touch **(Page#114)**
Speaker

Graphical user interfaces and the event-driven model are not applied on which of the following?

Web Applications
Mobile applications
Desktop Applications
Console Applications

C# is most often used with some implementation of the _____.

Compact Language Interface
Common Language Infrastructure
Control Language Infrastructure
Control Language Interface

In C#, comments are written using _____.

and */ /*

// and /* */

and /*

// and */ /*

Binding binds _____ properties together and keeps a communication channel open.

two **(Page#140)**
three
four
five

Attributes are declared _____ the class/function.

Outside
After
Within
Above

XML document is readable by _____.

Human only
Machine only
No one
Machine and Human both

"Console" is a/an _____ class in the "System" namespace.

Input
Dynamic
Static
Virtual

The only _____ .Net language has an intrinsic understanding of dependency Properties.

Vb.net
Xaml
C#
J#

Which of the following grabs the “OK” button by only knowing its name?

Button okButton = (Button)window.GetName("okButton")
Button okButton = (Button)window.getElementByName("okButton")
Button okButton = (Button)window.FindName("okButton")
Button okButton = (Button)window.GetButton("okButton")

Unlike type converters, however, markup extensions are invoked from XAML with _____.

Bridge assemblies
Explicit and consistent syntax
Wrapper code
None of the given options

Which statement is True about interfaces?

An interface can contain only methods, variables, events, and indexers.
An interface can contain only variables, properties, events, and indexers.
An interface can contain only methods, properties, variables, and indexers.
An interface can contain only methods, properties, events, indexers.

Events can be _____.

Specialized
Composed
Generalized
All of the given options

g++ is a command of _____.

Dev C++
Unix based system
Windows based system
None of the given options

_____ reads entire document in memory.

Xmldocuments
Xmlreader
Both of above
None of the given

_____ may contain events.

Requests
Service
Messages
Applications

A customer order can be represented as _____.

Event
Request
Event or Request
All of the given

Markup extensions are invoked from _____ with explicit and consistent syntax.

C#
HTML
XAML
None of the given

All the members of interfaces are implicitly_____.

Private
Static
Public
Abstract

A compiled C# file is called _____.

Complied File
Source File
Assembly File
EXE File

_____ add declarative information to your programs.

Attributes
Functions
Exception
Objects

Anonymous method is a method without a name.

True
False

“out” parameters which are not used for return values or the “params” argument for variable arguments.

True
False

_____types have the notion of referential identity.

Value
Reference
Both of above
None of the given

_____let any classes behave like an array.

Properties
Fields
Indexers
Finalizers

_____ are algorithms that operate on data.

Variables
Functions
Methods
Classes

_____ is an event that is introduced into an event processing system by an event producer.

Event Producer
Event Consumer
Raw Event
Event Stream

Events and _____ work hand in hand.

Parameters
Delegates
Functions
Data Members

C# event is a class member that is _____ whenever the event it was designed for occurs (fires).

Activated
Deactivated
Transformed
Associated

_____ is a static class in the System namespace.

Console
Pointer
Variable
Function

Default value of first member is _____.

0
1
2
3

_____ are inherited by classes which provide the real implementation.

Functions
Variables
Interfaces
Objects

_____ is basically array of arrays.

Stack
Pointer
Jagged
Function

The initial name for C# was _____.

COAL
COOP
COAP
COOL

BAML is compressed representation of _____.

CAML:NS
XAML
XAML:NS
CAML

Extension methods _____ access the private data members of the class in which they are extended.

May
Cannot
May not
Can

Which of the following is/are not derived from “Control” class?

ListBox
StatusBar
Both of above
None of the given

_____ events may or may not relate to an actual occurrence.

Observed Events
Deducted Events
Probabilistic Events
None of the above

_____ is an entity that receives events from the system.

Event Producer
Event Consumer
Event Channel
Event Generator

The easiest way to declare an event is to put the event keyword in front of a _____ member.

Interface
Delegate
Class
Struct

_____ can be used on non-data bound but inside the element it almost always make sense to use data binding.

Event
Template **(Page#144)**
Data
Function

_____ is typing together arbitrary objects.

data frame
data store
data binding **(Page#140)**
markup extension

----------Wish U Best of L|U|C|K for EXAMS ----------

Chapter 19

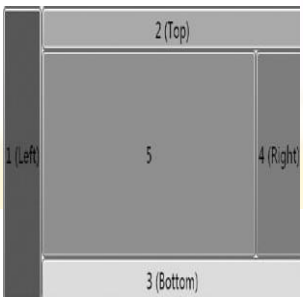
Lecture 19

Dock panel allows easy docking of elements to an entire side. Dock attached property has left, right, top, bottom values. Last child fills space unless lastchildfill=false.

```
<dockpanel>
  <Button dockpanel.Dock="Top" Background="Red">1(Top)</Button>
  <Button dockpanel.Dock="Left" Background="Orange">2 (Left)</Button>
  <Button dockpanel.Dock="Right" Background="Yellow">3 (Right)</Button>
  <Button dockpanel.Dock="Bottom" Background="Lime">4 (Bottom)</Button>
  <Button  Background="Aqua">5</Button>
</dockpanel >
```

Why are elements streching. Let's change alignment and see.

```
<dockpanel>
  <Button dockpanel.Dock="Top" horizontalalignment=”Right” Background=”Red”>1 (Top,
Align=Right) </Button>
  <Button dockpanel.Dock=”Left” verticalalignment=”Bottom” Background=”Orange”>2 (Left,
Align=Bottom)</Button>
  <Button dockpanel.Dock=”Right” verticalalignment=”Bottom” Background=”Yellow”>3 (Right,
Align=Bottom)</Button>
  <Button dockpanel.Dock=”Bottom” horizontalalignment=”Right” Background=”Lime”>4
(Bottom, Align=Right)</Button>
  <Button Background=”Aqua”>5</Button></dockpanel>
```

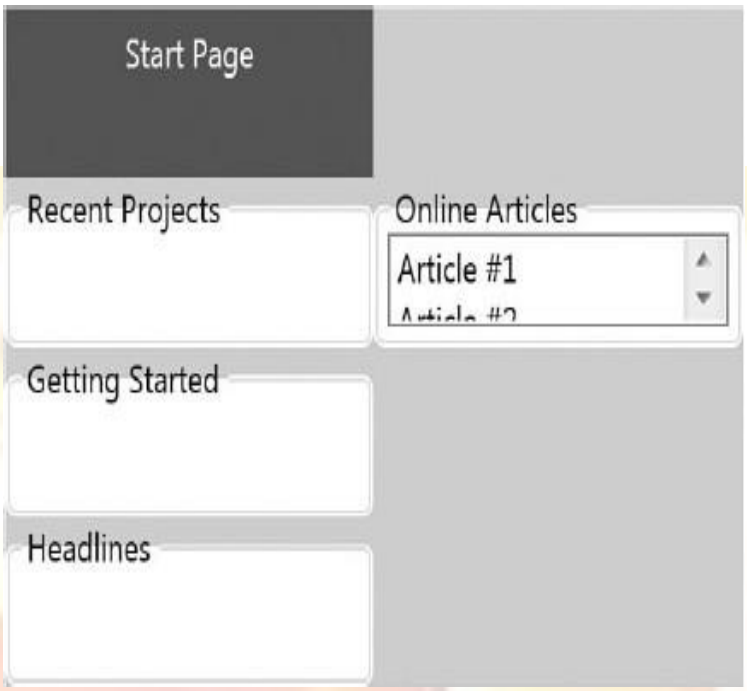


Its useful in a toplevel interface. Order of adding matters for the corners.

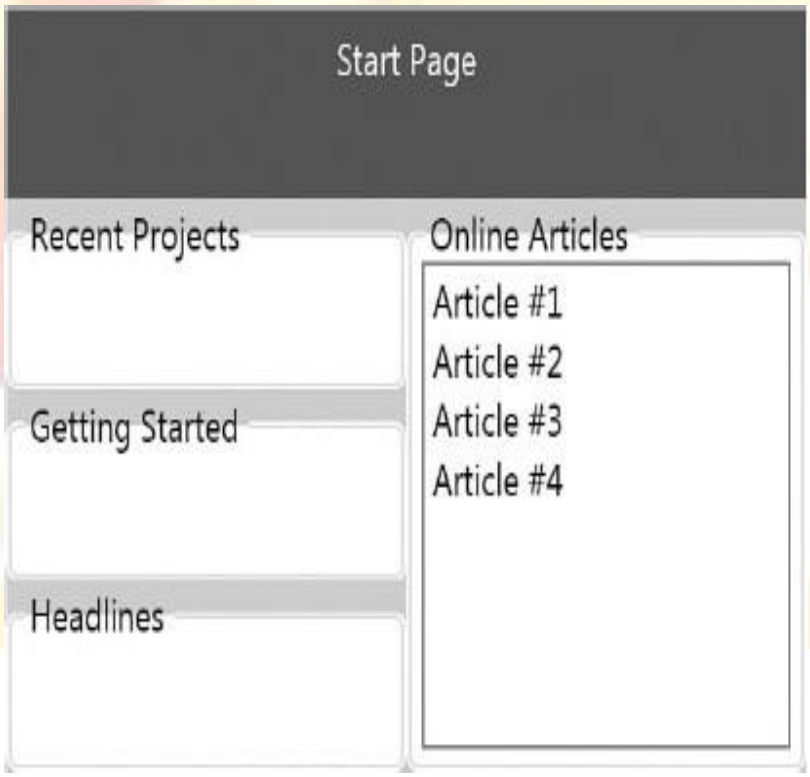
More elements can be added to any side. Its a superset of stackpanel.

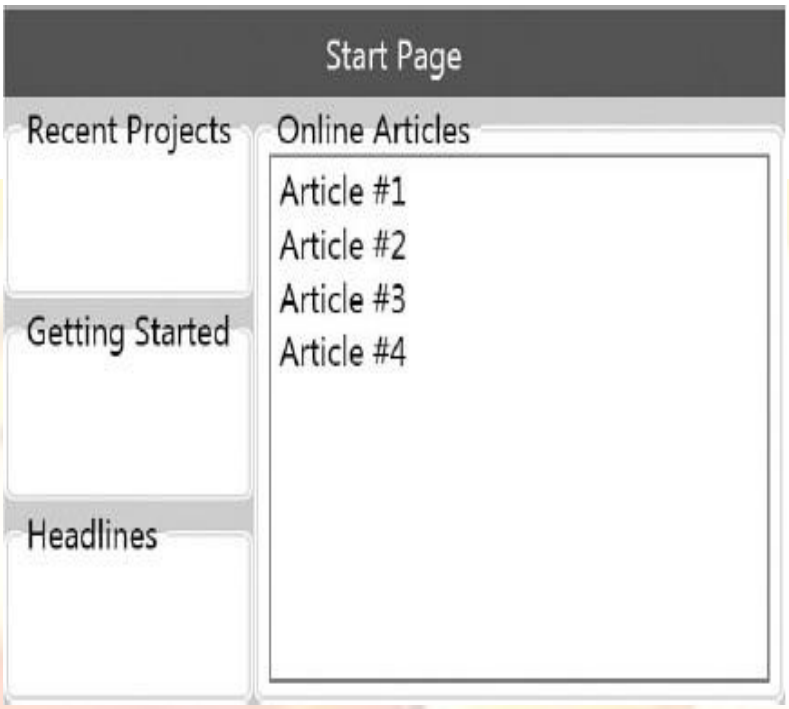
Grid is the most versatile. Its the default panel in VS and expression blend. You arrange things in multirow multicolumn, like a table in html. Also a Table class which is a frameworkcontentelement not a uielement.

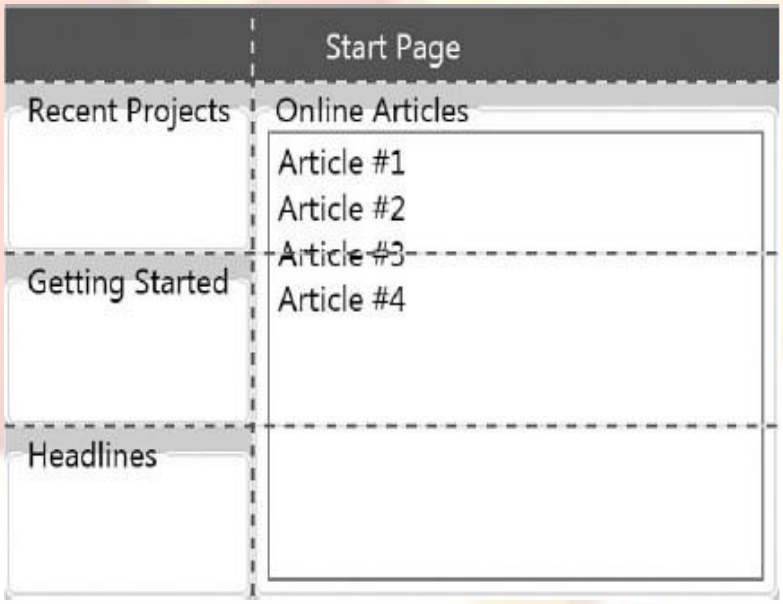
```
<Grid Background="lightblue">
  <!--  Define  four  rows: -->
  <Grid.rowdefinitions>
    <rowdefinition/>
    <rowdefinition/>
    <rowdefinition/>
    <rowdefinition/>
  </Grid.rowdefinitions >
  <!--  Define  two  columns:-->
  <Grid.columndefinitions>
    <columndefinition/>
    <columndefinition/>
  </Grid.columndefinitions >
  <!--  Arrange  the  children: -->
  <Label   Grid.Row="0" Grid.Column="0" Background="Blue" Foreground="White"
horizontalcontenta=""
<groupbox Grid.Row="1" Grid.Column="0" Background="White" Header="Recent Projects">
 </groupbox>
  <groupbox Grid.Row="2" Grid.Column="0" Background="White" Header="Getting
Started"></groupbox>
  <groupbox Grid.Row="3" Grid.Column="0" Background="White"
Header="Headlines"></groupbox>
  <groupbox Grid.Row="1" Grid.Column="1" Background="White" Header="Online
Articles"></groupbox>
  20 <listbox>
    <listboxitem>Article #1</listboxitem>
    <listboxitem>Article #2</listboxitem>
    <listboxitem>Article #3</listboxitem>
    <listboxitem>Article #4</listboxitem>
  </listbox >
  </groupbox >
</Grid >
```



We specify individual rows and cols. Its verbose but useful to specify details. Default is 1 cell. We position elements using Row and Column attached properties (zero-based) default is 0,0. Multiple elements in same cell simply overlap by z-order. Cells can be empty.







Online article list too small and and start page label not full width. We can make spanning rows and cols. Rowspan and columnspan 1 by default. By default height n width are same. Height and Width = "Auto" for sizing to content. Showgridlines = "True" to show grid lines.

For sizing rows and columns, use rowdefinition and Col. Height and Width not double but gridlength and not default to Auto or nan. Three types of sizing. Absoulte sizing: device independent pixels means no grow shrink, Autosizing: size to content, or Proportional or star sizing: grows or shrinks. When 1 row col * all remaining space is taken by it. When more row col * they divides remaining space. It can an be 2* or

5.5*. 2* is twice as * (1*). 5.5* is twice as 2.75*. Remaining space is after absolute and autosized rows or cols. Default width and height are *.

Here is how gridlength is used in procedural code.


```
Gridlength  length = new gridlength(100);
Gridlength length = new gridlength(0, gridunittype.Auto);
Gridlength length = new gridlength(100, gridunittype.Pixel);
Gridlength length = new gridlength(2, gridunittype.Star);
```

Grid splitter is used for interactive sizing. Any number of gridsplitters can be added to a Grid. Can be used to move entire row or column. Atleast one cell resizes and other cells behavior depends on weather absolute sizing or proportional is applied on them. It fits in one cell, but behavior affects entire row or col so better to use span. Which cells are affected depends on gridsplitters alignment values. Default horizontal alignment=right and vertical alignment=stretch. Reasonable use requires stretch in one dimension.

HorizontalAlignment

VerticalAlignment	Left	Right	Center	Stretch
Top	Current cell and cell to the left	Current cell and cell to the right	Cells to the left and right	Current cell and cell above
Bottom	Current cell and cell to the left	Current cell and cell to the right	Cells to the left and right	Current cell and cell below
Center	Current cell and cell to the left	Current cell and cell to the right	Cells to the left and right	Cells above and below
Stretch	Current cell and cell to the left	Current cell and cell to the right	Cells to the left and right	Cells to the left and right if `GridSplitter` is taller than it is wide, *or* cells to the top and bottom if `GridSplitter` is wider than it is tall

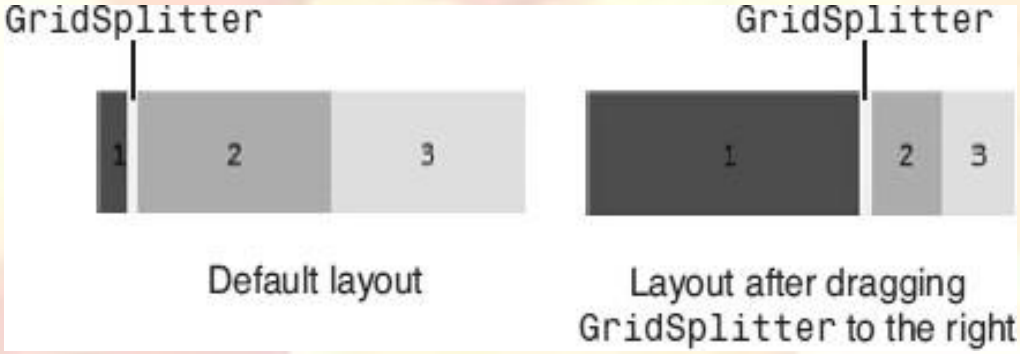
When all proportionally sized, changes co-efficients accordingly. When absolute sizing, changes only top or left of the cells. Remaining cells pushed down or right. Also has resize direction (default Auto, or can be set to Rows or Cols) and resize behavior for explicit control. Resize direction effect only when stretching in both directions. Resize behavior defaults to based on alignment. It can be set to previous and current, current and next, or previous and next to control which 2 rows or columns should be directly affected by resizing. Best way is to place it in its own auto sized row or column so it doesn't overlap existing content.

Chapter 20

Lecture 20

Let's revise grid.

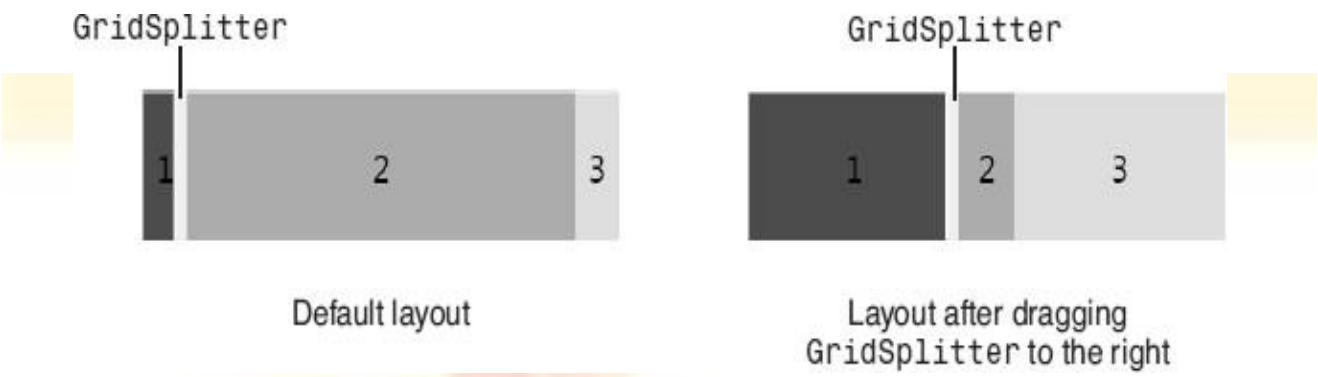
```
<Grid>
  <Grid.columndefinitions>
    <columndefinition  Width="Auto"/>
    <columndefinition/>
    <columndefinition/>
  </Grid.columndefinitions >
  <Label Grid.Column="0" Background="Red" horizontalcontentalignment="Center"
verticalcontenta=""
  </Label >
  <gridsplitter Grid.Column="0" Width="5"/>
  <Label Grid.Column="1" Background="Orange" horizontalcontentalignment="Center"
verticalconte=""
  </Label >
  <Label Grid.Column="2" Background="Yellow" horizontalcontentalignment="Center"
verticalconte=""
  </Label >
</Grid >
```



Sharedsizegroup enables multiple row cols to remain the same width height when length changed via gridsplitter.

```
<Grid issharedsizescope="True">
  <Grid.columndefinitions>
    <columndefinition  Width="Auto" sharedsizegroup="mygroup"/>
    <columndefinition/>
    <columndefinition  sharedsizegroup="mygroup"/>
  </Grid.columndefinitions >
  <Label Grid.Column="0" Background="Red" horizontalcontentalignment="Center"
verticalcontenta=""
  </Label >
  <gridsplitter Grid.Column="0" Width="5"/>
  <Label Grid.Column="1" Background="Orange" horizontalcontentalignment="Center"
verticalconte=""
  </Label >
  <Label Grid.Column="2" Background="Yellow" horizontalcontentalignment="Center"
verticalconte=""
```

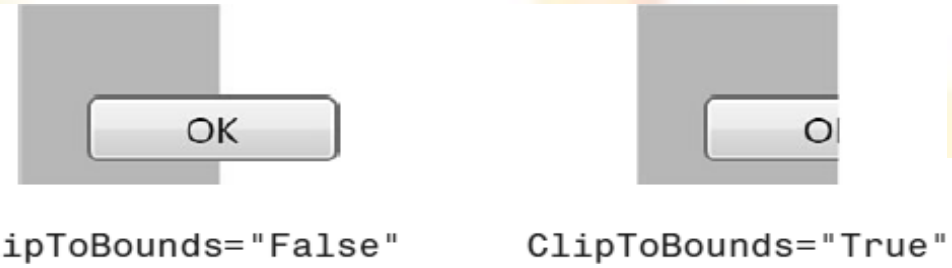
```
  </Label >
</Grid >
```



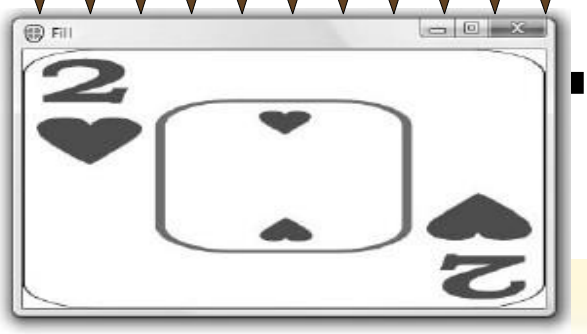
Issharedscope because sizegroups can be shared across multiple grids. All uses must be under a common parent with issharedscope set to true. It is an attached property of grid as well. Grid is usually the best choice. Except wraping, it can do what most panels do.

Content overflow can be dealt with Clipping, Scrolling, Scaling, Wrapping and Trimming. Wrapping already seen. Only way for non-text to wrap is using wrappanel. Trimming is intelligent form of clipping. Supported for text by textblock and accesstext. They have texttrimming=None (default), or characterellipsis, orwordellipsis.

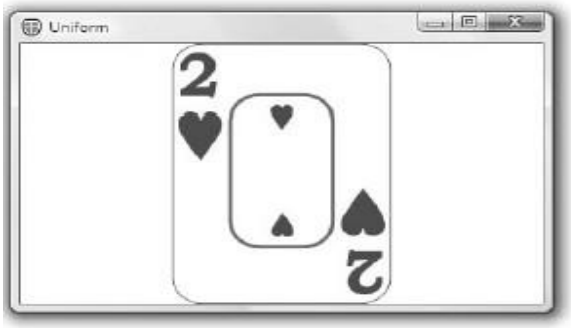
Clipping is default behavior. Edges of panel or cell area of dock area. All uielements have cliptobounds. Controls if it can draw outside bounds. Still cannot draw outside the window or page. Most panels clip regardless of this property. Canvas and uniformgrid dont unless this property is set. Button has this property false by default. Place canvas inside a grid cell to avert clipping. Clipping is done before rendertransform. Cant shrink back.



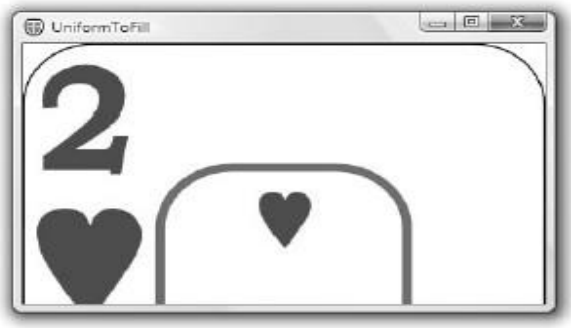
Stretch="None"



Stretch="Fill"



Stretch="Uniform"



Stretch="UniformToFill"

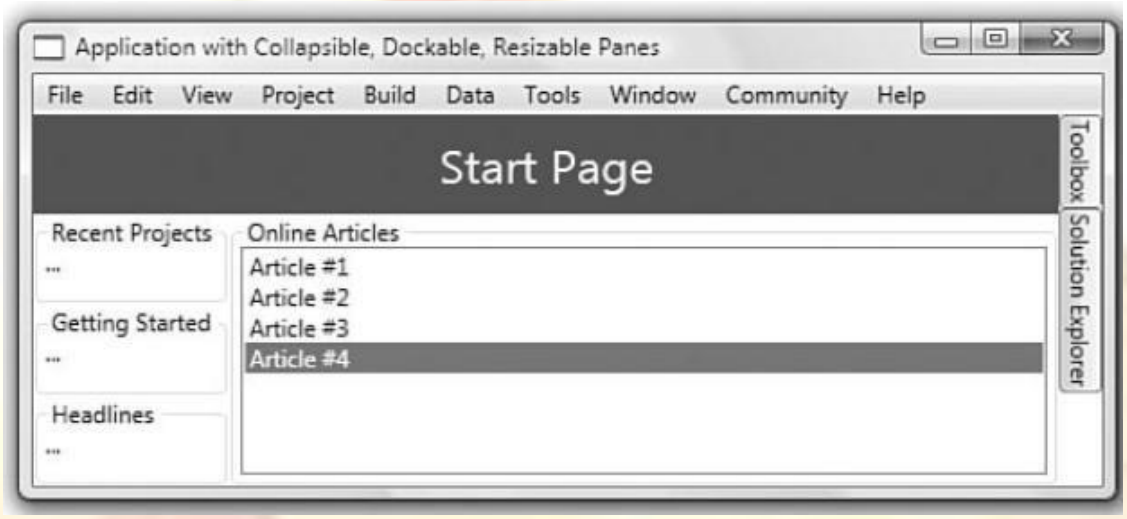
System.Windows.Controls.scrollviewer control can be used for scrolling. Its content property is set and verticalscrollbarvisibility and horizontalscrollbarvisibility properties determine the display.

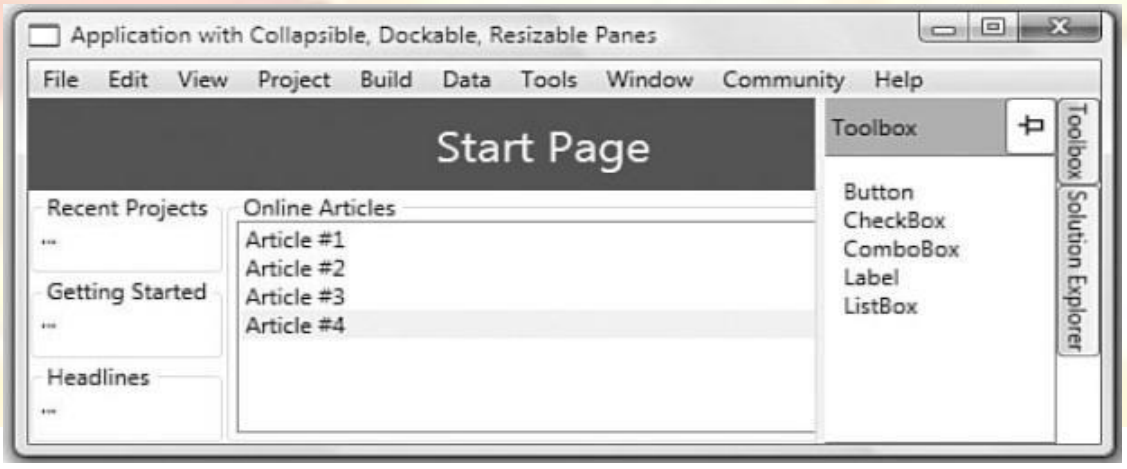
```
<Window Title="Using scrollviewer"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation">
    <scrollviewer>
        <stackpanel>
            …
        </stackpanel>
    </scrollviewer>
</Window>
```

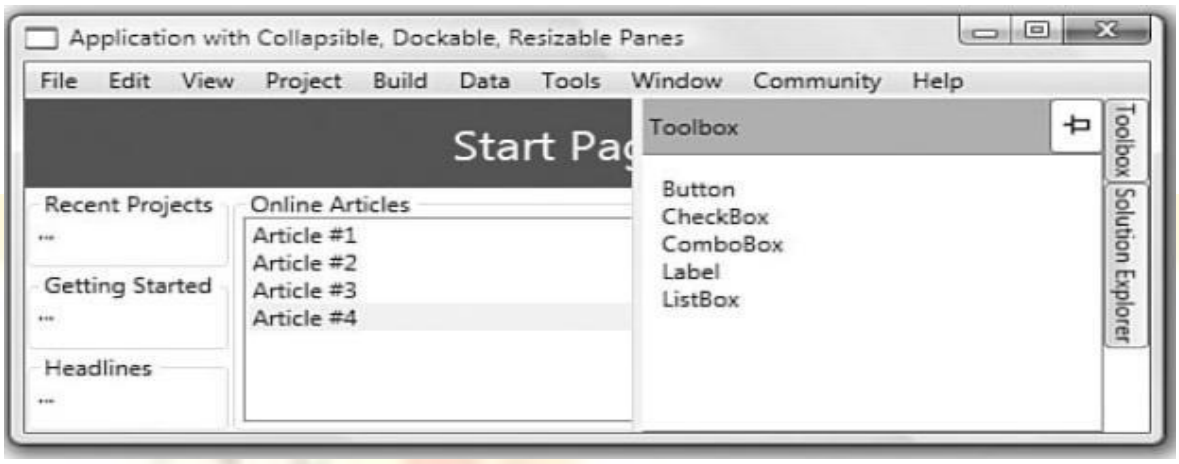
Scrolling more popular but e.g. Card game needs scaling. Scaletransform works relative to their own size not the available size. System.Windows.Controls.Viewbox, a type of class called Decorator (also Border). A panel-like thing but has only one child. Stretches to fill available space by default but also Stretch=None (like not using it at all), Fill, Uniform (aspect ratio, default), uniformtofill (cropped) strechdirection=uponly, downonly, Both (default).

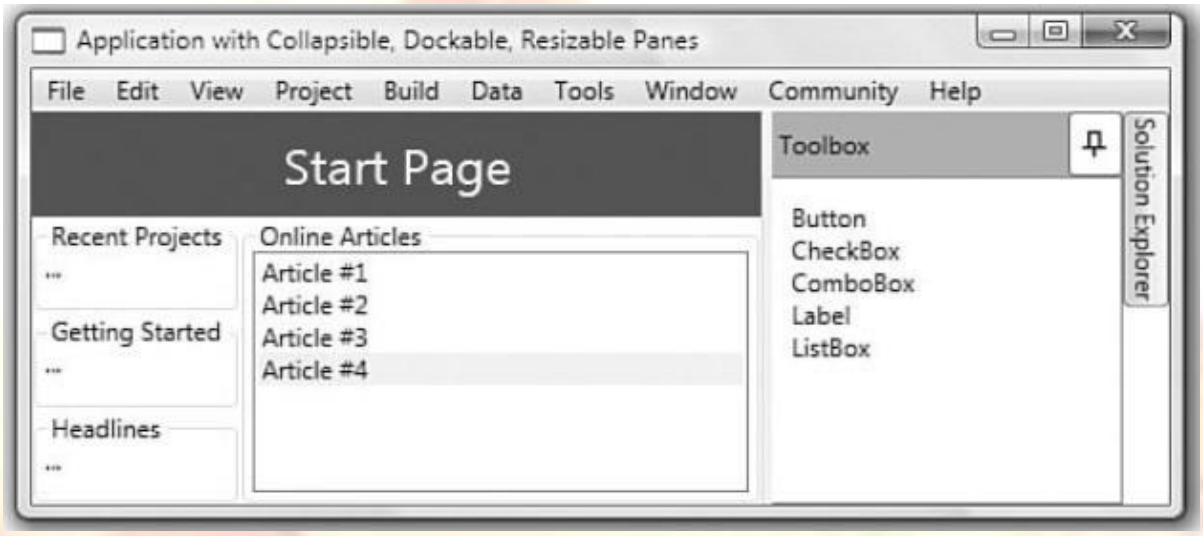
```
<Window Title="Using Viewbox"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation">
    <Viewbox>
        <stackpanel>
            …
        </stackpanel>
    </Viewbox>
</Window>
```

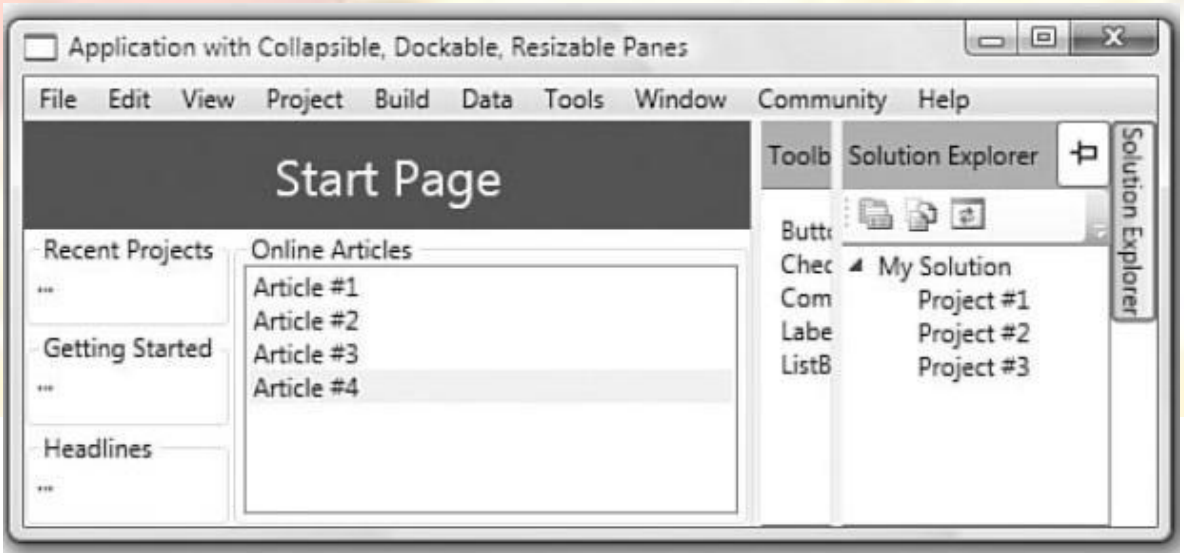
Let's make a visual studio like interface.

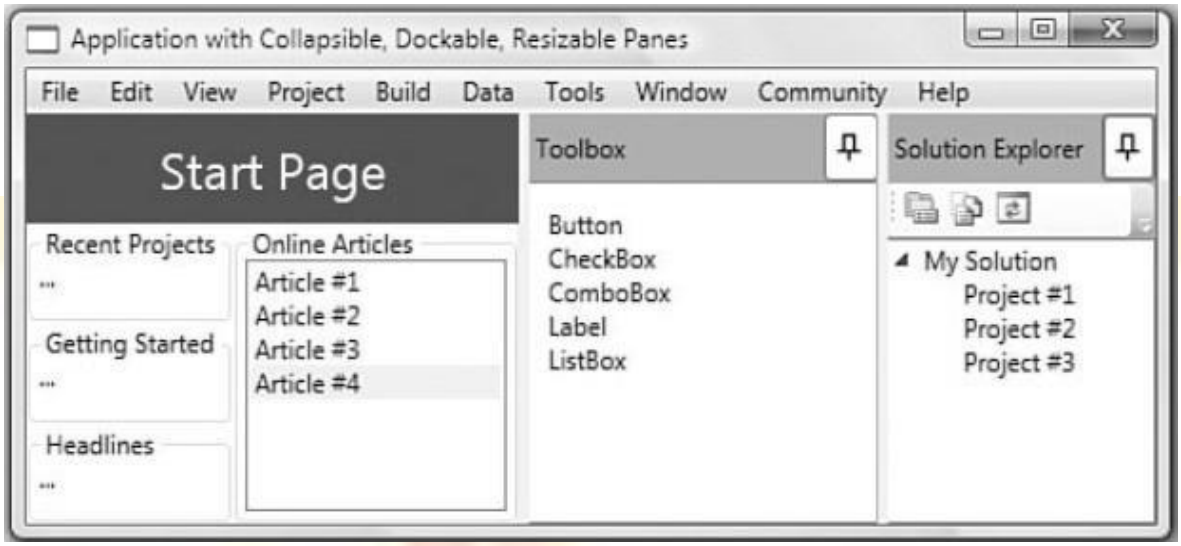












Chapter 21

Lecture 21

Because splitters are needed Grid is a reasonable idea. Three independent grids, because overlapping. Sharedsizegroup used to keep them in sync when docked. When docking cells are added / removed. Z order between layer 1 and 2 so undocked is on top. All 3 placed in another grid of single row col.

```
<Window x:Class="mainwindow"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Application with Collapsible, Dockable, Resizable Panes">
    <dockpanel>
        <Menu dockpanel.Dock="Top">
            My Interface
        </Menu>


        <!-- The bar of buttons docked along the right edge: -->
        <stackpanel Name="buttonbar" Orientation="Horizontal" dockpanel.Dock="Right">
            <stackpanel.layouttransform>
                <rotatetransform Angle="90"/>
            </stackpanel.layouttransform>
            <Button Name="pane1button" mouseenter="pane1button_mouseenter">
                Toolbox
            </Button>
            <Button Name="pane2button" mouseenter="pane2button_mouseenter">
                Solution Explorer
            </Button>
        </stackpanel>
        <!-- The Grid containing the three child Grids fills the dockpanel: -->
        <Grid Name="parentgrid" Grid.issharedsizescope="True">
            <Grid.columndefinitions>
                <columndefinition Width="5*"/>
                <columndefinition Width="364*"/>
            </Grid.columndefinitions>
            <!-- Layer 0: -->
            <Grid Name="layer0" mouseenter="layer0_mouseenter" Grid.columnspan="2">
                <!-- … (content of this Grid is similar to Listing 5.2) -->
            </Grid>
            <!-- Layer 1: -->
            <Grid Name="layer1" Visibility="Collapsed" Grid.columnspan="2">
                <Grid.columndefinitions>
                    <columndefinition/>
                    <columndefinition sharedsizegroup="column1" Width="auto"/>
                </Grid.columndefinitions>
                <!-- Column 0 is empty, but column 1 contains a Grid and a gridsplitter:
-->
                <Grid Grid.Column="1" mouseenter="pane1_mouseenter"
                      Background="{dynamicresource {x:Static
systemcolors.activecaptionbrushkey}}">
                    <Grid.rowdefinitions>
                        <rowdefinition Height="auto"/>
                        <rowdefinition/>
                    </Grid.rowdefinitions>
                    <!-- Row 0 contains a header, and row 1 contains pane-specific
content: -->
                    <dockpanel Grid.Row="0">
                        <Button Name="pane1pin" Width="26" dockpanel.Dock="Right"
                                Click="pane1pin_Click" Background="White">
                            <Image Name="pane1pinimage" Source="pinhorizontal.gif"/>
```

```
                        </Button>
                        <textblock Padding="8" texttrimming="characterellipsis"
                                   Foreground="{dynamicresource {x:Static
systemcolors.activecaptiontextbrushkey}}"
                                   Dockpanel.Dock="Left">Toolbox
                        </textblock>
                    </dockpanel>
                    <!-- … (pane-specific content fills row 1) -->
                </Grid>
                <gridsplitter Width="5" Grid.Column="1" horizontalalignment="Left"/>
            </Grid>
            <!-- Layer 2: -->
            <Grid Name="layer2" Visibility="Collapsed" Grid.columnspan="2">
                <Grid.columndefinitions>
                    <columndefinition/>
                    <columndefinition sharedsizegroup="column2" Width="auto"/>
                </Grid.columndefinitions>
                <!-- Column 0 is empty, but column 1 contains a Grid and a gridsplitter:
-->
                <Grid Grid.Column="1" mouseenter="pane2_mouseenter"
                      Background="{dynamicresource {x:Static
systemcolors.activecaptionbrushkey}}">
                    <Grid.rowdefinitions>
                        <rowdefinition Height="auto"/>
                        <rowdefinition Height="auto"/>
                        <rowdefinition/>
                    </Grid.rowdefinitions>
                    <!-- Row 0 contains a header, and rows 1 & 2 contain pane-specific
content: -->
                    <dockpanel Grid.Row="0">
                        <Button Name="pane2pin" Width="26" dockpanel.Dock="Right"
                                Click="pane2pin_Click" Background="White">
                            <Image Name="pane2pinimage" Source="pinhorizontal.gif"/>
                        </Button>
                        <textblock Padding="8" texttrimming="characterellipsis"
                                   Foreground="{dynamicresource {x:Static
systemcolors.activecaptiontextbrushkey}}"
                                   Dockpanel.Dock="Left">Solution Explorer
                        </textblock>
                    </dockpanel>
                    <!-- … (pane-specific content fills rows 1 & 2)-->
                </Grid>
                <gridsplitter Width="5" Grid.Column="1" horizontalalignment="Left"/>
            </Grid>
        </Grid>
    </dockpanel>
</Window>


Using System;
Using System.Windows;
Using System.Windows.Controls;
Using System.Windows.Media.Imaging;

Public partial class mainwindow : Window
{
    // Dummy columns for layers 0 and 1:
```

```
Columndefinition column1cloneforlayer0;
Columndefinition column2cloneforlayer0;
Columndefinition column2cloneforlayer1;
Public mainwindow()
{
    Initializecomponent();
    // Initialize the dummy columns used when docking:
    Column1cloneforlayer0 = new columndefinition();
    Column1cloneforlayer0.sharedsizegroup = "column1";
    Column2cloneforlayer0 = new columndefinition();
    Column2cloneforlayer0.sharedsizegroup = "column2";
    Column2cloneforlayer1 = new columndefinition();
    Column2cloneforlayer1.sharedsizegroup = "column2";
}
// Toggle between docked and undocked states (Pane 1)
Public void pane1pin_Click(object sender, routedeventargs e)
{
    If (pane1button.Visibility == Visibility.Collapsed)
        Undockpane(1);
    Else
        Dockpane(1);
}
// Toggle between docked and undocked states (Pane 2)
Public void pane2pin_Click(object sender, routedeventargs e)
{
    If (pane2button.Visibility == Visibility.Collapsed)
        Undockpane(2);
    Else
        Dockpane(2);
}
// Show Pane 1 when hovering over its button
Public void pane1button_mouseenter(object sender, routedeventargs e)
{
    Layer1.Visibility = Visibility.Visible;
    // Adjust Z order to ensure the pane is on top:
    Grid.setzindex(layer1, 1);
    Grid.setzindex(layer2, 0);
    // Ensure the other pane is hidden if it is undocked
    If (pane2button.Visibility == Visibility.Visible)
        Layer2.Visibility = Visibility.Collapsed;
}
// Show Pane 2 when hovering over its button
Public void pane2button_mouseenter(object sender, routedeventargs e)
{
    Layer2.Visibility = Visibility.Visible;
    // Adjust Z order to ensure the pane is on top:
    Grid.setzindex(layer2, 1);
    Grid.setzindex(layer1, 0);
    // Ensure the other pane is hidden if it is undocked
    If (pane1button.Visibility == Visibility.Visible)
        Layer1.Visibility = Visibility.Collapsed;
}
// Hide any undocked panes when the mouse enters Layer 0
Public void layer0_mouseenter(object sender, routedeventargs e)
{
    If (pane1button.Visibility == Visibility.Visible)
        Layer1.Visibility = Visibility.Collapsed;
```

```
        If (pane2button.Visibility == Visibility.Visible)
            Layer2.Visibility = Visibility.Collapsed;
    }
    // Hide the other pane if undocked when the mouse enters Pane 1
    Public void pane1_mouseenter(object sender, routedeventargs e)
    {
        // Ensure the other pane is hidden if it is undocked
        If (pane2button.Visibility == Visibility.Visible)
            Layer2.Visibility = Visibility.Collapsed;
    }
    // Hide the other pane if undocked when the mouse enters Pane 2
    Public void pane2_mouseenter(object sender, routedeventargs e)
    {
        // Ensure the other pane is hidden if it is undocked
        If (pane1button.Visibility == Visibility.Visible)
            Layer1.Visibility = Visibility.Collapsed;
    }
    // Docks a pane, which hides the corresponding pane button
    Public void dockpane(int panenumber)
    {
        If (panenumber == 1)
        {
            Pane1button.Visibility = Visibility.Collapsed;
            Pane1pinimage.Source = new bitmapimage(new Uri("pin.gif", urikind.Relative));
            // Add the cloned column to layer 0:
            Layer0.columndefinitions.Add(column1cloneforlayer0);
            // Add the cloned column to layer 1, but only if pane 2 is docked:
            If (pane2button.Visibility == Visibility.Collapsed)
                Layer1.columndefinitions.Add(column2cloneforlayer1);
        }
        Else if (panenumber == 2)
        {
            Pane2button.Visibility = Visibility.Collapsed;
            Pane2pinimage.Source = new bitmapimage(new Uri("pin.gif", urikind.Relative));
            // Add the cloned column to layer 0:
            Layer0.columndefinitions.Add(column2cloneforlayer0);
            // Add the cloned column to layer 1, but only if pane 1 is docked:
            If (pane1button.Visibility == Visibility.Collapsed)
                Layer1.columndefinitions.Add(column2cloneforlayer1);
        }
    }
    // Undocks a pane, which reveals the corresponding pane button
    Public void undockpane(int panenumber)
    {
        If (panenumber == 1)
        {
            Layer1.Visibility = Visibility.Visible;
            Pane1button.Visibility = Visibility.Visible;
            Pane1pinimage.Source = new bitmapimage(new Uri("pinhorizontal.gif",
urikind.Relative));
            // Remove the cloned columns from layers 0 and 1:
            Layer0.columndefinitions.Remove(column1cloneforlayer0);
            // This won't always be present, but Remove silently ignores bad columns:
            Layer1.columndefinitions.Remove(column2cloneforlayer1);
        }
        Else if (panenumber == 2)
        {
```

```
            Layer2.Visibility = Visibility.Visible;
            Pane2button.Visibility = Visibility.Visible;
            Pane2pinimage.Source = new bitmapimage(new Uri("pinhorizontal.gif",
urikind.Relative));
            // Remove the cloned columns from layers 0 and 1:
            Layer0.columndefinitions.Remove(column2cloneforlayer0);
            // This won't always be present, but Remove silently ignores bad columns:
            Layer1.columndefinitions.Remove(column2cloneforlayer1);
        }
    }
}
```

Let's discuss Input Events now i.e. Keyboard, mouse, stylus, multi-touch. We will start with routed events and commands. Routed events are like dependency properties on top of .net properties. Similarly routed events are a layer on top of .net events. They can travel up or down a visual or logical tree. Helps apps remain oblivious to the visual tree. E.g. Button exposes Click based on mouseleftbuttondown and keydown but actually buttonchrome or textblock visual child fires it. Event travels up the tree and Button eventually sees it. Or like in a previous example rectangle on top of button. So arbitrary complex content but Click still raised. Otherwise would require custom code on inner content or consumers. Very similar to dependency property concept.

Its an implementation, not a language feature other than xaml, like dependency properties. A handful of API's. Public static routedevent fields, and by convention Event suffix. Registered in the static ctor and a .net event wrapper. (wrapper shouldn't do anything else).

```
Public class Button: buttonbase
{
// The routed event
Public static readonly  routedevent  clickevent;
Static Button()
{
// Register the event
Button.clickevent = eventmanager.registerroutedevent("Click",routingstrategy.Bubble,
Typeof(routedeventhandler), typeof(Button));
//
}
// A .NET event wrapper (optional)
Public event  routedeventhandler  Click
{
Add{ addhandler(Button.clickevent, value);}
Remove{ removehandler(Button.clickevent, value);}
}
Protected  override void  onmouseleftbuttondown(mousebuttoneventargs e)
{
//
// Raise the event
Raiseevent(new routedeventargs(Button.clickevent, this));
//
}
//
}
```

Addhandler, removehandler, raiseevent are from uielement not dependencyobject.

Routing Strategies include Tunneling: from root down to source or until marked handled, Bubbling: from source to root or until marked handled, and Direct: only on source just like .net events but still participate in routed event specific things like event triggers.

Handlers for routed events have a System.Object parameter (sender to which handler was attached), and System.eventargs, and source which is the logical tree element that originally raised the event and originalsource which is the element in visual tree that raised the event e.g. Textblock on top of a button. Routedevent, actual routed event object like Button.clickevent which is useful when same handler is used for multiple events.

There aer many keyboard, mouse, multi-touch, and stylus events. Most are bubbling events but many have a tunneling counterpart. By convention tunneling event names are prefixed with Preview and come just before the bubbling event comes. Previewmousemove comes before mousemove. By convention, actions are only taken on bubbling event so tunneling gives a chance to cancel or modify the event e.g. Textbox with restrictive input should use previewkeydown.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
X:Class="aboutdialog" mouserightbuttondown="aboutdialog_mouserightbuttondown"
Title="About WPF 4 Unleashed" sizetocontent="widthandheight"
Background="orangered">
    <stackpanel>
        <Label fontweight="Bold" fontsize="20" Foreground="White">
            WPF 4 Unleashed
        </Label>
        <Label>© 2010 SAMS Publishing</Label>
        <Label>Installed Chapters:</Label>
        <listbox>
            <listboxitem>Chapter 1</listboxitem>
            <listboxitem>Chapter 2</listboxitem>
        </listbox>
        <stackpanel Orientation="Horizontal" horizontalalignment="Center">
            <Button minwidth="75" Margin="10">Help</Button>
            <Button minwidth="75" Margin="10">OK</Button>
        </stackpanel>
        <statusbar>You have successfully registered this product.</statusbar>
    </stackpanel>
</Window>
```

```
Using  System.Windows;
Using System.Windows.Input;
Using System.Windows.Media;
Using System.Windows.Controls;

Public partial class aboutdialog : Window
{
    Public aboutdialog()
    {
        Initializecomponent();
    }
    Void aboutdialog_mouserightbuttondown(object sender, mousebuttoneventargs e)
    {
        // Display information about this event
        This.Title = "Source = " + e.Source.gettype().Name + ", originalsource = " +
            E.originalsource.gettype().Name + " @ " + e.Timestamp;
        // In this example, all possible sources derive from Control
        Control source = e.Source as Control;
        // Toggle the border on the source control
        If (source.borderthickness != new Thickness(5))
        {
            Source.borderthickness = new Thickness(5);
            Source.borderbrush = Brushes.Black;
        }
        Else
            Source.borderthickness = new Thickness(0);
    }
}
```

Chapter 22

Lecture 22

Let's go over the code from last lecture again.

```
Using  System.Windows;
Using System.Windows.Input;
Using System.Windows.Media;
Using System.Windows.Controls;

Public partial class aboutdialog : Window
{
    Public aboutdialog()
    {
        Initializecomponent();
    }
    Void aboutdialog_mouserightbuttondown(object sender, mousebuttoneventargs e)
    {
        // Display information about this event
        This.Title = "Source = " + e.Source.gettype().Name + ", originalsource = " +
            E.originalsource.gettype().Name + " @ " + e.Timestamp;
        // In this example, all possible sources derive from Control
        Control source = e.Source as Control;
        // Toggle the border on the source control
        If (source.borderthickness != new Thickness(5))
        {
            Source.borderthickness = new Thickness(5);
            Source.borderbrush = Brushes.Black;
        }
        Else
            Source.borderthickness = new Thickness(0);
    }
  }
```

Note that we never receive an event on listbox item. Button has no border element so setting border has no effect. Handling a halted event can only be done from procedural code.

Let's discuss attached events. Attached events can bubble and tunnel through elements who have not defined the event. Like prop value inheritance and attach properties. Let's see a new example with window handling listbox selectionchanged and button click events but does not define them.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
X:Class="aboutdialog" listbox.selectionchanged="listbox_selectionchanged"
Button.Click="Button_Click"
Title="About WPF Unleashed" sizetocontent="widthandheight"
Background="orangered">
    <stackpanel>
        <Label fontweight="Bold" fontsize="20" Foreground="White">
            WPF 4 Unleashed
        </Label>
        <Label>© 2010 SAMS Publishing</Label>
        <Label>Installed Chapters:</Label>
```

```
        <listbox>
            <listboxitem>Chapter 1</listboxitem>
            <listboxitem>Chapter 2</listboxitem>
        </listbox>
        <stackpanel Orientation="Horizontal" horizontalalignment="Center">
            <Button minwidth="75" Margin="10">Help</Button>
            <Button minwidth="75" Margin="10">OK</Button>
        </stackpanel>
        <statusbar>You have successfully registered this product.</statusbar>
    </stackpanel>
</Window>


Using System.Windows;
Using System.Windows.Controls;
Public partial class aboutdialog : Window
{
    Public aboutdialog()
    {
        Initializecomponent();
    }
    Void listbox_selectionchanged(object sender, selectionchangedeventargs e)
    {
        If (e.addeditems.Count > 0)
            Messagebox.Show("You just selected " + e.addeditems[0]);
    }
    Void Button_Click(object sender, routedeventargs e)
    {
        Messagebox.Show("You just clicked " + e.Source);
    }
}
```

Xaml valid because compiler sees relevant events defined in button and listbox at runtime however addhandler is directly called. Equivalent code is:

```
Public  aboutdialog()
{
Initializecomponent();
This.addhandler(listbox.selectionchangedevent,
New selectionchangedeventhandler(listbox_selectionchanged));
This.addhandler(Button.clickevent,new routedeventhandler(Button_Click));
}
```

Theoretically one handler can do all event handling. We can use delegate contravariance i.e. Arguments can be of base type than delegate.

```
Void generichandler(object sender, routedeventargs e)
{
If(e.routedevent == Button.clickevent)
{
Messagebox.Show("You just clicked "+ e.Source);
}
Else if (e.routedevent == listbox.selectionchangedevent)
{
Selectionchangedeventargs sce= (selectionchangedeventargs)e;
If(sce.addeditems.Count > 0)
Messagebox.Show("You just selected " + sce.addeditems[0]);
}
```

```
}
```

Let's discuss keyboard events keydown, keyup, and preview versions of them. Keyeventargs contains Key, imepro-cessedkey, deadcharprosessedkey, systemkey, isup, isdown, istoggled, keystates, isrepeat, and keyboarddevice.System.Windows.Input.Keyboard and its primarydevice property are accessible everywhere.

```
protected override void OnKeyDown(KeyEventArgs e)
{
if ((e.KeyboardDevice.Modifiers & ModifierKeys.Alt) == ModifierKeys.Alt && (e.Key ==
Key.A || e.SystemKey == Key.A))
{
// Alt+A has been pressed, potentially also with Ctrl, Shift, and/or Windows
}
base.OnKeyDown(e);
}

protected override void OnKeyDown(KeyEventArgs e)
{
if (e.KeyboardDevice.Modifiers == ModifierKeys.Alt && (e.Key == Key.A || e.SystemKey ==
Key.A))
{
// Alt+A and only Alt+A has been pressed
}
base.OnKeyDown(e);
}
```

You can use keyboarddevice.iskeydown to even check if left or right alt is down etc. What is keyboard focus. Uiele-ment Focusable property (true by default). Focusablechanged event, iskeyboardfocused, iskeyboardfocuswithin (readonly). To set focus, use Focus or movefocus. Iskeyboardfocusedchanged, iskeyboardfocuswithinchanged, gotkeyboardfocus, lostkeyboardfocus, previewgotkeyboardfocus, and previewlostkeyboardfocus are intersting events.

Mouse events are mouseenter and mouseleave (e.g. To create rollover effect however the preferred way is to use trigger with ismouseover property or ismousedirectlyover and ismousedirectlyoverchanged properties), mousemove and previewmousemove, mouseleftbuttondown, mouserightbuttondown, mouseleftbuttonup, mouserightbuttonup, and the more generic mousedown and mouseup, as well as the previewxxx versions of all six of these events, mousewheel and previewmousewheel. If Visibility=Collapsed no mouse events are generated but opacity=0 generates all events. Null background fill stroke produce areas with no events however "transparent" or color does. So null and transparent brush look same but different for hit test.

Mouseeventargs has 5 props of type mousebuttonstate (Pressed or Released) leftbutton, rightbutton, middlebutton, xbutton1, and xbutton2. Getposition function returning a Point with x and y props. Pass null for screen relative position or an element for position with respect to that element. Mousewheel and previewmousewheel get mousewheeleventargs derived from mouseeventargs (add a Delta property). 12 events in mouse up down get mousebuttoneventargs adds a changedbutton property, a buttonstate property (for the changed button) and a clickcount (clicks with time ¡ double click speed). Button base class raises a mousedoubleclick by checking clickcount in mouseleftbuttondown, similarly in preview versions.

For drag and drop, we have dragenter, dragover, dragleave, with previewdragenter, previewdragover, and previewdragleave, Drop and previewdrop, querycontinuedrag and previewquerycontinuedrag. Drag and drop of clipboard content to/from elements not elements themselves. Its enabled by alldrop=true. First two sets give drageventargs with getposition, Data, Effects and allowedeffects (Copy, Move, Link, Scroll, All, None), keystates (leftmousebutton, rightmousebutton, middlemousebutton, shiftkey, controlkey, altkey, or None) Continue evnets are raised when a keyboard or mouse state changes during the operation and querycontinuedragargs has keystates, escapepressed, Action (Continue, Drop, Cancel).

To understand mouse capturing imagine drag and drop of elements. How to implement with mouseleftbuttondown, mousemove, and mouseleftbuttonup. But mouse too fast or under another element. Uielements can capture mouse and release mousecapture. Properties is mouse captured and is mouse capture within, and the events got mouse capture, lost mouse capture, is mouse capture changed, and is mouse capture within changed are related to capturing. Inside mouse move you can use a layout or rendertransform to move.

Stylus can behave like a mouse but has higher resolution, inverted, in air or not, pressure sensitivity. Similarly multi-touch events are like mouse but not the other way around. There are available on win7 or later with multi-touch hardware. We can either use basic touch events or higher level manipulation events.

Basic touch events are touchenter and touchleave, touchmove and previewtouchmove, touchdown, touchup, previewtouchdown and previewtouchu, gottouchcapture and losttouchcapture. With multiple fingers, events raised for each finger separately. For first finger mouse events are generated as well. Toucheventargs has gettouchpoint, getintermediatetouchpoints, touchdevice. Touchpoint has Position, Size, Bounds, touchdevice, Action (Down, Up, Move). Each finger has its own touchdevice identified by Id porp.

Some notebooks support only two simultaneous touch points. Silverlight 4 doesnt support them. A lower level framereported event that isnt even routed is however raised. Now Let's discuss manipulation events for panning, rotating, and zooming. They are easy to apply with a transform. Swipe easy with the other events as well but rotate etc. Difficult with previous events and lack of consistency in their behaviour between applications. Therefore using the manipulation events is preferred.

Chapter 23

Lecture 23

Let's see an example using touch events.

```
<Window x:Class="touchevents.mainwindow"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Touch Events">
    <Canvas Name="canvas">
        <Canvas.Background>
            <lineargradientbrush>
                <gradientstop Color="Black"/>
                <gradientstop Color="Red" Offset="1"/>
            </lineargradientbrush>
        </Canvas.Background>
    </Canvas>
</Window>

Using System;
Using System.Collections.Generic;
Using System.Windows;
Using System.Windows.Controls;
Using System.Windows.Input;
Using System.Windows.Media;
Using System.Windows.Media.Imaging;

Namespace touchevents
{
    Public partial class mainwindow : Window
    {
        // Keep track of which images are used for which touchdevices
        Dictionary<touchdevice, Image> fingerprints =
        New Dictionary<touchdevice, Image>();
        Public mainwindow()
        {
            Initializecomponent();
        }
        Protected override void ontouchdown(toucheventargs e)
        {
            Base.ontouchdown(e);
            // Capture this touch device
            Canvas.capturetouch(e.touchdevice);
            // Create a new image for this new touch
            Image fingerprint = new Image {
                Source = new bitmapimage(new
Uri("pack://application:,,,/fingerprint.png"))
            };
            // Move the image to the touch point
            Touchpoint point = e.gettouchpoint(canvas);
            Fingerprint.rendertransform = new translatetransform(point.Position.X,
point.Position.Y);
            // Keep track of the image and add it to the canvas
```

```
            Fingerprints[e.touchdevice] = fingerprint;
            Canvas.Children.Add(fingerprint);
        }
        Protected override void ontouchmove(toucheventargs e)
        {
            Base.ontouchmove(e);
            If (e.touchdevice.Captured == canvas)
            {
                // Retrieve the right image
                Image fingerprint = fingerprints[e.touchdevice];
                Translatetransform transform =
                Fingerprint.rendertransform as translatetransform;
                // Move it to the new location
                Touchpoint point = e.gettouchpoint(canvas);
                Transform.X = point.Position.X;
                Transform.Y = point.Position.Y;
            }
        }
        Protected override void ontouchup(toucheventargs e)
        {
            Base.ontouchup(e);
            // Release capture
            Canvas.releasetouchcapture(e.touchdevice);
            // Remove the image from the canvas and the dictionary
            Canvas.Children.Remove(fingerprints[e.touchdevice]);
            Fingerprints.Remove(e.touchdevice);
        }
    }
}
```

Manipulation events are manipulationstarting and manipulationstarted, manipulationdelta, manipulationcompleted. We combine information from multiple events. Manipulation events work if ismanipulationenabled=true on this or a parent and basic events not handled. Manipulation starting and started events at first finger touchdown, delta at touchmove, and completed after touchup for all fingers. Starting and started events allow to customize or cancel or disallow some manipulations. Manipulationdelta class has Translation, Scale, Rotation, Expansion (like Scale but dev ind px instead of scale factor). Manipulationdeltaeventargs has deltamanipulation and cumulativemanipulation. Let's see an example to move rotate zoom a photo.

```
<Window x:Class="manipulationevents.mainwindow"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Manipulation Events">
    <Canvas Name="canvas" ismanipulationenabled="True">
        <Image Name="photo" Source="photo.jpg">
            <Image.rendertransform>
                <matrixtransform/>
            </Image.rendertransform>
        </Image>
    </Canvas>
</Window>

Using System;
Using  System.Windows;
Using System.Windows.Input;
Using System.Windows.Media;
```

```
Namespace manipulationevents
{
    Public partial class mainwindow : Window
    {
        Public mainwindow()
        {
            Initializecomponent();
            Canvas.manipulationdelta += Canvas_manipulationdelta;
        }
        Void Canvas_manipulationdelta(object sender, manipulationdeltaeventargs e)
        {
            Matrixtransform transform = photo.rendertransform as matrixtransform;
            If (transform != null)
            {
                // Apply any deltas to the matrix,
                // then apply the new Matrix as the matrixtransform data:
                Matrix matrix = transform.Matrix;
                Matrix.Translate(e.deltamanipulation.Translation.X,
e.deltamanipulation.Translation.Y);
                Matrix.rotateat(e.deltamanipulation.Rotation, e.manipulationorigin.X,
e.manipulationorigin.Y);
                Matrix.scaleat(e.deltamanipulation.Scale.X, e.deltamanipulation.Scale.Y,
e.manipulationorigin.X, e.manipulationorigin.Y);
                Transform.Matrix = matrix;
                E.Handled = true;
            }
        }
    }
}
```

Manipulations always done relative to a manipulation container. By default element with is manipulation enabled=true or handle manipulation starting event and set manipulation starting event args.manipulation container. Now what is Inertia. Manipulation inertia starting event when all fingers loose contact even before a completed event. By default manipulationcompleted thrown right away but you can set manipulation inertia starting event args.translation behavior, rotation behavior, and/or expansion behavior which will cause deltas thrown and then completed until inertia is completed. Translation behavior has desired displacement, desireddeceleration, and initialvelocity. Rotation behavior has desired rotation, desired deceleration, and initial velocity.

Expansionbehavior has desiredexpansion, desireddeceleration, initialradius, and initialvelocity. Typically only set desireddeceleration or the behavior-specific desireddisplacement, desiredrotation, or desiredexpansion. The latter properties are useful for ensuring that the element doesnt go too far. By default, initialvelocity and initialradius are initialized with the current values. You can get the various velocities by checking manipulationinertiastartingeventargs.initialvelocities, which has linearvelocity, angularvelocity, and expansionvelocity.

```
Using System;
Using System.Windows;
Using  System.Windows.Input;
Using  System.Windows.Media;
Namespace manipulationevents
{
    Public partial class mainwindow : Window
    {
        Public mainwindow()
        {
            Initializecomponent();
```

```
            Canvas.manipulationdelta += Canvas_manipulationdelta;
            Canvas.manipulationinertiastarting += Canvas_manipulationinertiastarting;
        }
        Void Canvas_manipulationinertiastarting(object sender,
manipulationinertiastartingeventargs e)
        {
            E.translationbehavior.desireddeceleration = 0.01;
            E.rotationbehavior.desireddeceleration = 0.01;
            E.expansionbehavior.desireddeceleration = 0.01;
        }
        Void Canvas_manipulationdelta(object sender, manipulationdeltaeventargs e)
        {
            Matrixtransform transform = photo.rendertransform as matrixtransform;
            If (transform != null)
            {
                // Apply any deltas to the matrix,
                // then apply the new Matrix as the matrixtransform data:
                Matrix matrix = transform.Matrix;
                Matrix.Translate(e.deltamanipulation.Translation.X,
e.deltamanipulation.Translation.Y);
                Matrix.rotateat(e.deltamanipulation.Rotation, e.manipulationorigin.X,
e.manipulationorigin.Y);
                Matrix.scaleat(e.deltamanipulation.Scale.X, e.deltamanipulation.Scale.Y,
e.manipulationorigin.X, e.manipulationorigin.Y);
                Transform.Matrix = matrix;
                E.Handled = true;
            }
        }
    }
}
```

To be boundary aware manipulationboundaryfeedback event is used. Inside a manipulationdelta event handler, you can call the reportboundaryfeedback method on the passed-in manipulationdeltaeventargs instance to make the window bounce similar to iphone bounce list behavior. Let's see a spin prize wheel example.

```
<Window x:Class="spintheprizewheel.mainwindow"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Spin the Prize Wheel">
    <Window.Background>
        <lineargradientbrush>
            <gradientstop Color="White"/>
            <gradientstop Color="Orange" Offset="1"/>
        </lineargradientbrush>
    </Window.Background>
    <Grid Name="grid" ismanipulationenabled="True">
        <Image Name="prizewheel" rendertransformorigin="0.5,0.5"
               Source="prizewheel.png" Margin="0 30 0 0">
            <Image.rendertransform>
                <rotatetransform/>
            </Image.rendertransform>
        </Image>
        <Image Source="arrow.png" verticalalignment="Top" Stretch="None"/>
    </Grid>
</Window>
```


```
Using System;
Using System.Windows;
Using System.Windows.Input;
Using System.Windows.Media;
Namespace spintheprizewheel
{
    Public partial class mainwindow : Window
    {
        Public mainwindow()
        {
            Initializecomponent();
            Grid.manipulationstarting += Grid_manipulationstarting;
            Grid.manipulationdelta += Grid_manipulationdelta;
            Grid.manipulationinertiastarting += Grid_manipulationinertiastarting;
            Grid.manipulationcompleted += Grid_manipulationcompleted;
        }
        Void Grid_manipulationstarting(object sender, manipulationstartingeventargs e)
        {
            E.Mode = manipulationmodes.Rotate; // Only allow rotation
        }
        Void Grid_manipulationdelta(object sender, manipulationdeltaeventargs e)
        {
            (prizewheel.rendertransform as rotatetransform).Angle +=
            E.deltamanipulation.Rotation;
        }
        Void Grid_manipulationinertiastarting(object sender,
manipulationinertiastartingeventargs e)
        {
            E.rotationbehavior.desireddeceleration = 0.001;
        }
        Void Grid_manipulationcompleted(object sender, manipulationcompletedeventargs e)
        {
            // Now that the wheel has stopped, tell the user what s/he won!
        }
    }
}
```

You can take advantage of panning support built into scrollviewer by setting its panningmode property to horizontalonly, verticalonly, horizontalfirst, verticalfirst, or Both. You can download the Surface Toolkit for Windows Touch to get numerous slick Microsoft Surface WPF controls that are optimized for multi-touch. This includes “surface versions” of most common controls (such as surfacebutton and surfacecheckbox) and brand-new controls (such as scatterview and librarystack).

Chapter 24

Lecture 24

Commands are a more abstract and loosely coupled version of events e.g. Cut copy paste commands. They are exposed in various ways. They can be enabled disabled e.g. If there is nothing to paste. Two-way communication gets cumbersome especially if you dont want the list of controls hard-coded.

WPF defines a number of built-in commands. Commands have automatic support for input gestures (such as keyboard shortcuts). Some of wpfs controls have built-in behavior tied to various commands. Any object implementing icommand which defines Execute, canexecute, canexecutechanged can work as a Command. For Cut, Copy, and Paste, you could define and implement three classes implementing icommand, find a place to store them (perhaps as static fields of the main Window), call Execute from relevant event handlers (when canexecute returns true), and handle the canexecutechanged event to toggle the isenabled property on the relevant pieces of user interface. Controls have logic to interface with commands through Command property. We can also do it all in xaml.

Following are pre-defined builtin commands. Applicationcommands e..g Close, Copy, Cut, Delete, Find, Help, New, Open, Paste, Print, printpreview, Properties, Redo, Replace, Save, saveas, selectall, Stop, Undo, and more. Componentcommands e.g. Movedown, moveleft, moveright, moveup, scrollbyline, scrollpagedown, scrollpageleft, scrollpageright, scrollpageup, selecttoend, selecttohome, selecttopagedown, selecttopageup, and more. Mediacommands e.g. Channeldown, channelup, decreasevolume, fastforward, increasevolume, mutevolume, nexttrack, Pause, Play, previoustrack, Record, Rewind, Select, Stop, and more. Navigationcommands e.g. Browseback, browseforward, browsehome, browsestop, Favorites, firstpage, gotopage, lastpage, nextpage, previouspage, Refresh, Search, Zoom, and more. Editingcommands e.g. Aligncenter, alignjustify, alignleft, alignright, correctspellingerror, decreasefontsize, decreaseindentation, enterlinebreak, enterparagraphbreak, ignorespellingerror, increasefontsize, increaseindentation, movedownbyline, movedownbypage, movedownbyparagraph, moveleftbycharacter, moveleftbyword, moverightbycharacter, moverightbyword, and more.

All instances of routeduicommand implement icommand and support bubbling.

Helpbutton.Command= applicationcommands.Help;

If you run it would be always disabled. We need to add commandbinding to the element or a Parent (bubbling). All uielements have commandbindings collection.

```
This.commandbindings.Add(new commandbinding(applicationcommands.Help, helpexecuted,
helpcanexecuted));
// Navigate to a page instance
Photopage  nextpage= new photopage();
This.navigationservice.Navigate(nextpage);
// Or navigate to a page via a URI
This.navigationservice.Navigate(new Uri("photopage.xaml", urikind.Relative));
This.commandbindings.Add( new commandbinding(applicationcommands.Help, helpexecuted,
helpcanexecuted));
```

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
X:Class="aboutdialog"
Title="About WPF Unleashed" sizetocontent="widthandheight"
Background="orangered">
    <Window.commandbindings>
```

```
        <commandbinding Command="Help"
Canexecute="helpcanexecute" Executed="helpexecuted"/>
    </Window.commandbindings>
    <stackpanel>
        <Label fontweight="Bold" fontsize="20" Foreground="White">
            WPF 4 Unleashed
        </Label>
        <Label>© 2010 SAMS Publishing</Label>
        <Label>Installed Chapters:</Label>
        <listbox>
            <listboxitem>Chapter 1</listboxitem>
            <listboxitem>Chapter 2</listboxitem>
        </listbox>
        <stackpanel Orientation="Horizontal" horizontalalignment="Center">
            <Button minwidth="75" Margin="10" Command="Help"
                    Content= "{
                        Binding relativesource={relativesource Self}, Path=Command.Text
                    }"/>
            <Button minwidth="75" Margin="10">OK</Button>
        </stackpanel>
        <statusbar>You have successfully registered this product.</statusbar>
    </stackpanel>
</Window>

Using System.Windows;
Using System.Windows.Input;
Public partial class aboutdialog : Window
{
    Public aboutdialog()
    {
        Initializecomponent();
    }
    Void helpcanexecute(object sender, canexecuteroutedeventargs e)
    {
        E.canexecute = true;
    }
    Void helpexecuted(object sender, executedroutedeventargs e)
    {
        System.Diagnostics.Process.Start("http://www.adamnathan.net/wpf");
    }
}
```

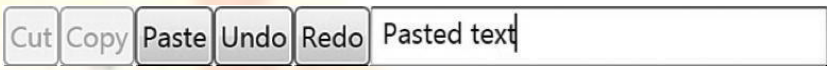
There is a commandconverter type converter. X:Static applicationcommands.Help.

Custom commands dont get this treatment. Now even F1 works, its an input gesture. You can also add keybindings and mousebindings yourself.

```
This.inputbindings.Add(
New keybinding(applicationcommands.Help, new keygesture(Key.F2)));

<Window.inputbindings>
<keybinding  Command="Help" Key="F2"/>
<keybinding  Command="notacommand" Key="F1"/>
</Window.inputbindings >

This.inputbindings.Add(
New keybinding(applicationcommands.notacommand, new keygesture(Key.F1)));
```

Controls with builtin command bindings e.g. Textbox that responds to ctrl-z etc.

```
<stackpanel xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Orientation="Horizontal" Height="25">
<Button Command="Cut" commandtarget="{Binding elementname=textbox}"
Content="{Binding  relativesource={relativesource Self}, Path=Command.Text}"/>
<Button Command="Copy" commandtarget="{Binding elementname=textbox}"
Content="{Binding  relativesource={relativesource Self}, Path=Command.Text}"/>
<Button Command="Paste" commandtarget="{Binding elementname=textbox}"
Content="{Binding  relativesource={relativesource Self}, Path=Command.Text}"/>
<Button Command="Undo" commandtarget="{Binding elementname=textbox}"
Content="{Binding  relativesource={relativesource Self}, Path=Command.Text}"/>
<Button Command="Redo" commandtarget="{Binding  elementname=textbox}"
```




```
Content="{Binding  relativesource={relativesource Self}, Path=Command.Text}"/>
<textbox x:Name="textbox" Width="200"/>
</stackpanel >
```

Button and textbox have no direct knowledge of each other. Through commands we have a rich interaction. The more the standardization on builtin commands, the more seamless and declarative the interaction can be between controls.

In summary, wpf input events make possible rich interactive content. We focused on uielement but the same events can be used with contentelement.

Chapter 25

Lecture 25

Let's discuss structuring and deploying an application. A standard windows app or partial trust web app or loose xaml. We will discuss Photo Gallery example with the book. App.xaml and mainwindow.xaml and code-behind files on a new wpf project. WPF Window is a win32 window, same chrome (non-client area) same taskbar behavior etc. Icon Title windowstyle topmost showintaskbar properties. Left and Top props or windowstartuplocation=centerscreen or centerowner. Any number of child windows can be made by instantiating a Window dervied class and calling Show. Child window like parent window but gets closed when parent and similarly minimized, also called modeless dialog.

Another windows Owner property after parent shown, ownedwindows property. There are Activated & Deactivated events. Activate method (like setforegroundwindow). Showactivated=false, initially not shown.

```
Public partial class mainwindow : Window
{
Public  mainwindow()
{
Initializecomponent();
}
Protected  override void onclosing(canceleventargs e)
{
Base.onclosing(e);
If (messagebox.Show("Are you sure you want to close Photo Gallery?",
"Annoying Prompt", messageboxbutton.yesno, messageboximage.Question)
== messageboxresult.No)
E.Cancel = true;
}
Protected  override void onclosed(eventargs e)
{
Base.onclosed(e);
// Persist the list of favorites
//
}
Protected  override void oninitialized(eventargs e)
{
Base.oninitialized(e);
// Retrieve the persisted list of favorites
//
}
Void exitmenu_Click(object sender, routedeventargs e)
{
This.Close();
}
}
Public static void Main()
{
Mainwindow window = new mainwindow();
Window.Show();
}
```

End result same whether you attach event handler or override this method. Technically a bit faster and more

neat when subclass wants to handle something base class throws. Dispatcherobject cant be accessed from a different thread which avoids many issues. App terminating, you need to add a message loop to process windows messages and pass them to the correct window.

```
[stathread]
Public static void Main()
{
Application app = new Application();
Mainwindow window= new mainwindow();
Window.Show();
App.Run(window);
}
[stathread]
Public static void Main()
{
Application app= new Application();
App.startupuri = new Uri("mainwindow.xaml", urikind.Relative);
App.Run();
}


<Application x:Class="photogallery.App"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Startupuri="mainwindow.xaml"/>
Using System.Windows;
Namespace  photogallery
{
Public partial class App : Application
{
Public App()
{
Initializecomponent();
}
}
}
Main()
[System.stathreadattribute()]
Public static void Main()
{
Photogallery.App app = new photogallery.App();
App.initializecomponent();
App.Run();
}
```

App code-behind can be omitted altogether. But where is main. App.xaml is assigned applicationdefinition which causes App.g.cs to be generated and can be seen in Solution Explorer if you Show all files. System.Environment.getcommandlinea or set build action to Page and write Main. Startup, Exit,Activated, Deactivated (any window), sessionending (can-cellable logoff shutdown). A read-only Windows collection, mainwindow property (read-write). Shutdownmode (when to end the app). Properties dictionary for sharing between windows. Application.Current from any window. Let's discuss creating a single instance app. Can create app without Application using Dispatcher.Run or Win32 message loop but explicit termination and other issues. One UI Thread and one render thread. Can create more with Dispatcher.Run which can improve responsiveness. Can schedule on UI thread with Dispatcher.Invoke and begininvoke and priority from send (now) to systemidle.

```
Bool mutexisnew;
Using (System.Threading.Mutex m =
New System.Threading.Mutex(true, uniquename, out mutexisnew))
{
If (mutexisnew)
// This is the first instance. Run the application.
Else
// There is already an instance  running. Exit!
}
```

You can also create a splash screen. Nothing fancy because wpf not loaded yet. In wpf project - add new item -splash screen, adds an images with build action splashscreen.

Modal dialogs include common dialogs which are actually provided by win32. Instantiate, call showdialog and process result.

```
Void printmenu_Click(object sender, routedeventargs e)
{
String filename = (picturebox.selecteditem as listboxitem).Tag as string;
Image image = new Image();
Image.Source= new bitmapimage(new Uri(filename, urikind.relativeorabsolute));
Printdialog pd = new printdialog();
If (pd.showdialog()== true) // Result could be true, false, or null
Pd.printvisual(image, Path.getfilename(filename)+ " from Photo Gallery");
}


Void renamemenu_Click(object sender, routedeventargs e)
{
String filename = (picturebox.selecteditem as listboxitem).Tag as string;
Renamedialog dialog = new renamedialog(Path.getfilenamewithoutextension(filename));
If (dialog.showdialog() == true) // Result could be true, false, or null
{
// Attempt to rename the file
Try
{
File.Move(filename, Path.Combine(Path.getdirectoryname(filename),
Dialog.newfilename)+ Path.getextension(filename));
}
Catch (Exception ex)
{
Messagebox.Show(ex.Message,   Cannot Rename File , messageboxbutton.OK,
Messageboximage.Error);
}
}
}
```

A window to be shown as a dialog sets its dialogresult to bool. Setting it closes window or set Button's isdefault prop to true.

```
Void okbutton_Click(object sender, routedeventargs e)
{
This.dialogresult = true;
}
```

```
[stathread]
Public static void Main()
{
Mainwindow window = new mainwindow();
Window.showdialog();
}
```

Let's discuss persisting and restoring. Can use registry or file system but Let's use .net isolated storage. Physically located in the users document folder in a hidden folder. VS generated Settings class provide an even easier and strongly typed way but in a app. Config file.

```
Protected  override void onclosed(eventargs e)
{
Base.onclosed(e);
// Write each favorites item when the application is about to close
Isolatedstoragefile f = isolatedstoragefile.getuserstoreforassembly();
Using (isolatedstoragefilestream stream = new isolatedstoragefilestream("myfile",
filemode.Create, f));
Using (streamwriter  writer = new streamwriter(stream));
{
Foreach (treeviewitem item in favoritesitem.Items)
Writer.writeline(item.Tag as string);
}
}
Protected  override void oninitialized(eventargs e)
{
Base.oninitialized(e);
// Read each favorites item when the application is initialized
Isolatedstoragefile f = isolatedstoragefile.getuserstoreforassembly();
Using (isolatedstoragefilestream stream =
New isolatedstoragefilestream("myfile", filemode.openorcreate, f))
Using (streamreader reader = new streamreader(stream))
{
String line = reader.readline();
While (line != null)
{
Addfavorite(line);
Line = reader.readline();
}
}
//
}
```

Let's discuss clickonce vs. Windows installer. VS setup and deployment projects or a simpler clickonce wizard from build-¿publish menu. Windows installer benefits include showing custom setup UI like a EULA, control over where files installed, arbitrary code at setup time, install shared assemblies in global assembly cache, register COM components and file associations, install for all users, o✏ine installation from cd. Clickonce benefits include builtin support for automatic updates and rollback to prev versions, a web-like "go-away" experience or start menu and control panel program list, all files in isolated area, so no effect on other apps, clean uninstallation, but full trust apps can do things while they run, .net code access security, partial trust.

Chapter 26

Lecture 26

Let's discuss navigation based apps like windows explorer, media player, photo gallery. Can use navigation support for a wizard or organize the whole ui around navigation. With nav, content is usually in "Page" a simpler version of Window. Then hosted in a navigationwindow or a Frame. They provide support for navigating, a history journal, and nav related events. Navigationwindow more like a top-level window whereas Frame more like an HTML frame or iframe By default navwind has a bar on top with back/fwd, frame doesnt but can be shown.

```
<navigationwindow
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
Xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
Mc:Ignorable="d" x:Class="photogallery.Container"
Title="Photo Gallery" Source="mainpage.xaml"
D:designwidth="1320" d:designheight="919"
/>
```

Nav enabled version points startup uri to this and and mainpage.xaml referenced above has the content.

```
<Page x:Class="photogallery.mainpage"
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Photo Gallery" Loaded="Page_Loaded">
<!--   A p p l i c a t i o n -specific  c o n t e n t -->
</Page >
```

Page does everything window does except onclosed and onclosing. With these changes, nothing much changes in our app, just some disabled back/fwd buttons. Navigation can also happen between html files. A Page can interact with its navigation container by using the navigationservice class, which exposes relevant functionality regardless of whether the container is a navigationwindow or a Frame. You can get an instance of navigationservice by calling the static navigationservice.getnavigationservice method and passing the instance of the Page. But even more easily, you can simply use Pages navigationservice property. For example, you can set a title that is used in the drop-down menu associated with the Back and Forward buttons as follows:

```
This.navigationservice.Title = Main Photo Gallery  Page;
```

Or you can refresh the current Page as follows:

```
This.navigationservice.Refresh();
```

But Page also contains a few of its own properties that control the behavior of the parent container, such as windowheight, windowwidth, and windowtitle. Can also set in xaml. You can perform navigation in three main ways: Calling the Navigate method, Using Hyperlinks, Using the journal. Navigation containers support a Navigate method.

```
// Navigate to a page instance
Photopage  nextpage= new photopage();
This.navigationservice.Navigate(nextpage);
// Or navigate to a page via a URI
This.navigationservice.Navigate(new Uri("photopage.xaml", urikind.Relative));
```

Root of xaml must be “Page”. Or for html

```
This.navigationservice.Navigate(new Uri("http://www.adamnathan.net/wpf"));
```

Can also use 2 properties. Only useful from xaml.

```
This.navigationservice.Content = nextpage;
This.navigationservice.Source= new Uri("photopage.xaml", urikind.Relative);
```

Hyperlink element is used to link to xaml. Or handle its Click event and call navigate yourself to link from html towpf. Handle Navigating event and look for a sentinel HREF value can also set targetname to update a frame or use# and any named element in a page.

```
<textblock>
Click
<Hyperlink  navigateuri="photopage.xaml">here</Hyperlink> to view the photo.
</textblock >
```

Let’s discuss using Journal. Journal provides logic behind back and fwd. Internally two stacks. Back / fwd moves pages between stacks. Any other action empties the fwd stack. Back fwd can also nav. Containers goback and goforward and cangoback/fwd to avoid exception. Nav. Wnd. Always has journal but frame may depending on its journalownership=ownsjournal, usesparentjournal, Automatic (parent when hosted in wnd or frame). When frame has journal it has back/fwd buttons but can set navigationuivisibility=Hidden. When nav. With URI or hyperlink, always new instance. Can control when calling navigate with page so work to remember state. However, journalentry.keepalive attached prop. Can help for back fwd. Removefromjournal means a page is not in journal.

Other purposes of journal e.g. Application specific undo redo scheme. Nav. Container addbackentry and pass customcontentstate abstract class with a replay method that must be defined. Optionally journalentryname can be set. We use this in photo gallery for undoable image rotation.

```
[Serializable]
Class rotatestate : customcontentstate
{
Frameworkelement  element;
Double rotation;
Public  rotatestate(frameworkelement element, double rotation)
{
This.element = element;
This.rotation= rotation;
}
Public override string  journalentryname
{
Get{ return "Rotate " + rotation + "    "; }
}
Public override void Replay(navigationservice  navigationservice, navigationmode mode)
{
// Rotate the element by the specified amount
Element.layouttransform = new rotatetransform(rotation);
}
}
```

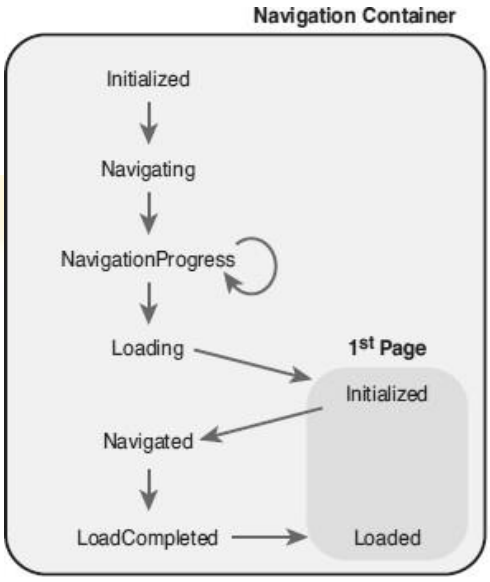


FIGURE 7.6 Navigation events that are raised when the first page is loaded.

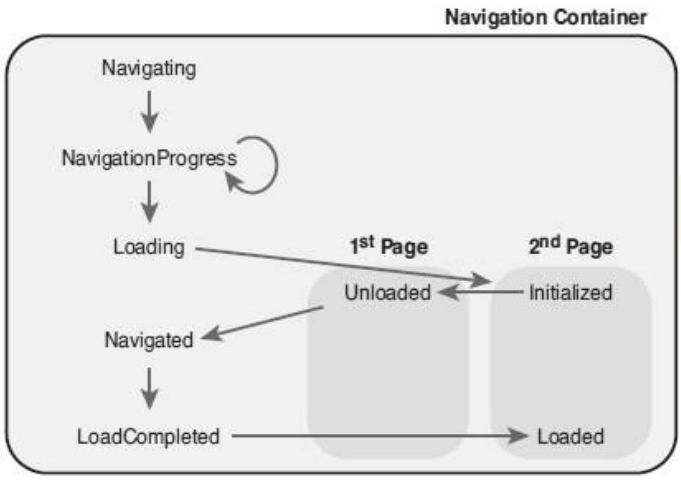


FIGURE 7.7 Navigation events that are raised when navigation occurs between two pages.

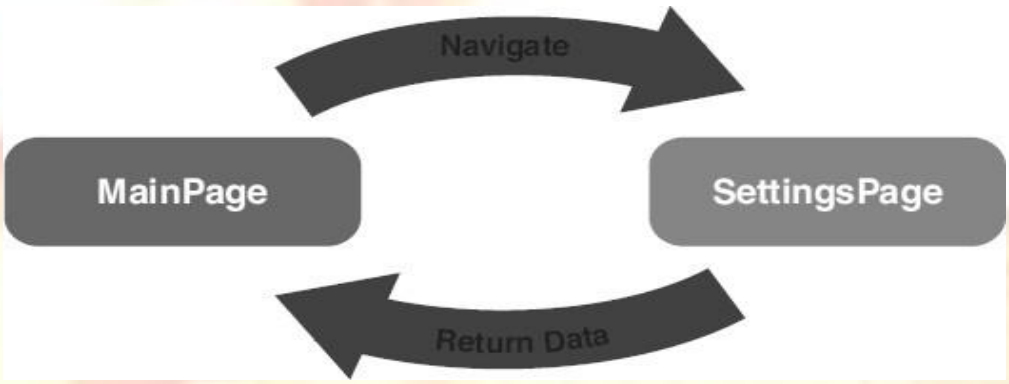
Navigationstopped is an event called instead of loadcompleted if an error occurs or nav. Is cancelled. These events also defined in "Application" to handle for any nav. Cont. Html to html nav. Not in journal, no event raised. To send data to pages, Navigate overloads with an extra "object" param. Target page receives it in loadcompleted.

```
Int photoid = 10;
// Navigate to a page instance
Photopage  nextpage= new photopage();
This.navigationservice.Navigate(nextpage, photoid);
// Or navigate to a page via a URI
This.navigationservice.Navigate(new Uri("photopage.xam", urikind.Relative),photoid);


This.navigationservice.loadcompleted += new
Loadcompletedeventhandler(container_loadcompleted);
//
Void  container_loadcompleted(object sender, navigationeventargs e)
{
If (e.extradata != null)
Loadphoto((int)e.extradata);
}


Int photoid = 10;
// Navigate to a page instance
Photopage nextpage= new photopage (photoid);
This.navigationservice.Navigate(nextpage);

Public photopage(int id)
{
Loadphoto(id);
}
```

```
// Navigate to a page by instance or URI
Application.Properties["photoid"] = 10;
This.navigationservice.Navigate(/* */);
If (Application.Properties["photoid"] != null)
Loadphoto((Application.Properties["photoid"]);
```

Let's see returning data with a pagefunction.

Pagefunction acts likea function.




```
<pagefunction
Xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Xmlns:sys="clr-namespace:System;assembly=mscorlib"
X:Class="myproject.pagefunction1"
X:typearguments="sys:String"
Title="pagefunction1">
<Grid>
</Grid >
</pagefunction >


Pagefunction1nextpage = new pagefunction1<string>();
This.navigationservice.Navigate(nextpage);
Nextpage.Return += new returneventhandler<string>(nextpage_Return);
//
Void nextpage_Return(object sender, returneventargs<string> e)
{
String  returnvalue= e.Result;
}
Onreturn(new returneventargs<string>("the data"));
```

Chapter 27

Lecture 27

Let's discuss XAML Browser apps (XBAPS). Silverlight more popular to deliver partial-trust wpf content in a browser. Differences are that not all features available by default, nav is integrated into the browser, deployment different. In VS, create WPF browser application. Create UI in Page and compile and run. Really just online-only clickonce applications, with some special wpf handling for browser-integrated experience.

```
// Whoops! Partially trusted code is not allowed to get this data!
Addfavorite(Environment.getfolderpath(Environment.specialfolder.mypictures));
```

Clickonce caching. VS increments. Or you can clear cache. Change some settings and its a xbap. But partial trust restricts many api. E.g. Above call exception. Requires fileiopermission which is not by default. Hit n trial to find what works. Local registry and file system, new windows (popups ok), unmanaged, but others dependent on browser, or on implementation details. Browserinterophelper.isbrowserhosted property to check if its a XBAP.

```
String  filecontents = null;
Openfiledialog ofd = new openfiledialog ();
If(ofd.showdialog() == true) // Result could be true, false, or null
{
Using(Stream s= ofd.openfile())
Using(streamreader sr= new streamreader(s))
{
Filecontents= sr.readtoend();
}
}
```

But u can still use rich text and media, isolated storage upto 512kb, arbitrary files on host web server, use browser file open dialog for local files. How parameters passed in. Either browserinterophelper.Source to retrieve the complete URL or browser cookie retrieved by Application.getcookie method. Any assembly marked with allowpartiallytrustedcallers and placed in GAC can be called by partial-trust code and it can be security loophole too.

There are also full-trust browser apps. In the project file you change

```
<targetzone>Internet</targetzone>
```

To this:

```
<targetzone>Custom</targetzone>
```

And in clickonce application manifest you add:

```
<permissionset class="System.Security.permissionset" version="1"
ID="Custom" samesite="site" Unrestricted="true"/>
```

You get Integrated Navigation in XBAP. Many XBAP don't take advantage of navigation. Such applications can set shownavigationui=false on Page. IE 7 and later merge the journal and provide a more streamlined interface. However if it appears in an iframe, there is still a separate nav. Bar. To publish, use VS publishing wizard or mage tool in SDK and copy files to webserver which must be configured to serve it. Users can install and run by just navigating to a URL. No security prompt if you dont need non. Stdandard permissions.

Let's discuss downloading files on demand. We will assign a set of loose files to a download group in VS. Under publish-¿application files in proj. Prop. Then use api in System.Deployment.Application to prompt download and benotified e.g. Progress bar while app loads. Page1 starts, loads mygroup and shows page2.

```
Using System;
Using System.Windows.Controls;
Using System.Windows.Threading;
Using System.Deployment.Application;
Public partial class Page1: Page
{
Public Page1()
{
Initializecomponent();
}
Protected  override void oninitialized(eventargs e)
{
Base.oninitialized(e);
If (applicationdeployment.isnetworkdeployed)
{
Applicationdeployment.currentdeployment.downloadfilegroupcompleted +=
Delegate {
Dispatcher.begininvoke(dispatcherpriority.Send,
New dispatcheroperationcallback(gotopage2), null);
};
Applicationdeployment.currentdeployment.downloadfilegroupasync("mygroup");
}
Else
{
Gotopage2(null);
}
}
Private object  gotopage2(object o)
{
Return  navigationservice.Navigate(new Uri("Page2.xaml", urikind.Relative));
}
}
```

Additional events can be used for fine grained progress. Loose xaml is powerful sometimes instead of html. In summary, XBAP have similar impl. From desktop app to web app. Deployment just needs the right .net installed. Wpf 3.5 by default on win7, wpf4 now.

Let's discuss resources e.g. Bitmap, fonts, string tables. Wpf builds on top, has binary resources and logical resources. Binary resources are what rest of .net framework considers a resource. Even compiled xaml stored as a resource. Can be embedded in assembly, loose file that may or may not be known at compile time. Can be localizable or not. To define a binary resource, add a file and set the build action resource or content (loose file). Dont use embeddedresource. Its misleading name but wpf doesnt fully support. Adding as content and loading directly are kind of equal but resource is neat. Should be embed if localizable or single binary file benefits. Let's see how to assign binary resource whether embed or loose. Wont work if not added to proj. If not added be explicit about the path.

```
<stackpanel Grid.Column="1" Orientation="Horizontal" horizontalalignment="Center">
<Button x:Name="previousbutton" tooltip="Previous(Left Arrow)">
<Image Height="21" Source="previous.gif"/>
</Button >
```



```
<Button x:Name="slideshowbutton" tooltip="Play Slide Show (F11)">
<Image Height="21" Source="slideshow.gif"/>
</Button >
<Button x:Name="nextbutton" tooltip="Next (Right Arrow)">
<Image Height="21" Source="next.gif"/>
</Button >
</stackpanel >

<Image Height="21" Source="pack://siteoforigin:,,,/slideshow.gif"/>

Image image = new Image();
Image.Source= new bitmapimage(new Uri("pack://application:,,,/logo.jpg"));
```

Can use subfolders for embedded resources. From procedural code, cant use xaml spec. So there are shortcuts.

Chapter 28

Lecture 28

Let's discuss localizing binary resources. Can partition into satellite assembly and use locbaml to manage string localization. To spec. A default culture and auto. Build a satellite assembly, you can to set uiculture. Need to open project file in a text ed. Add under debug, release etc. Or where it effects all prop. If you rebuild your project with this setting in place, youll find an en-US folder alongside your assembly, containing the satellite assembly named assemblyname.resources.dll. Also mark assembly with the neutral resource language matching.

```
<Project >
<propertygroup>
<uiculture>en-US</uiculture>
[assembly: neutralresourceslanguage("en-US",
Ultimateresourcefallbacklocation.Satellite)]
```

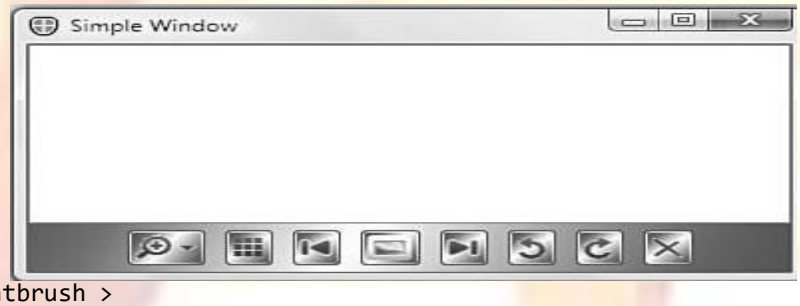
Next, apply Uid directive to any element needing localization. Msbuild /t:updateuid projectname.csproj. Locbaml /parse projectname.g.en-US.resources /out:en-US.csv. Now edit and localize all strings. Locbaml /generate Project-Name.resources.dll /trans:fr-CA.csv /cul:fr-CA. Then copy the assembly with a name matching the locale. To test, System.Threading.Thread.currentthread.currentuiculture (and System.Threading.Thread.currentthread.currentculture) to an instance of the desired cultureinfo.

Logical resources are introduced by wpf. Arbitrary .net object stored and named in an elements Resources prop. Typically meant to be shared by multiple child objects. Frameworkelement and framework content element both have a Resources prop. Often are style or data providers. First examples using simple brushes. Much like css, share objects.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Title="Simple Window" Background="Yellow">
  <dockpanel>
    <stackpanel  dockpanel.Dock="Bottom"
    Orientation="Horizontal"
    Horizontalalignment="Center">
      <Button Background="Yellow"
      Borderbrush="Red" Margin="5">
        <Image Height="21" Source="zoom.gif"/>
      </Button >
      <Button  Background="Yellow"
      Borderbrush="Red"
      Margin="5">
        <Image Height="21"
        Source="defaultthumbnailsize.gif"/>
      </Button >
      <Button  Background="Yellow"
      Borderbrush="Red"
      Margin="5">
        <Image Height="21"
        Source="previous.gif"/>
      </Button >
      <Button  Background="Yellow"
      Borderbrush="Red"
      Margin="5">
        <Image Height="21"
```

```
        Source="slideshow.gif"/>
      </Button >
      <Button  Background="Yellow"
      Borderbrush="Red"
      Margin="5">
        <Image Height="21"
        Source="next.gif"/>
      </Button >
      <Button  Background="Yellow"
      Borderbrush="Red"
      Margin="5">
        <Image Height="21"
        Source="counterclockwise.gif"/>
      </Button >
      <Button  Background="Yellow"
      Borderbrush="Red"
      Margin="5">
        <Image Height="21"
        Source="clockwise.gif"/>
      </Button >
      <Button  Background="Yellow"
      Borderbrush="Red"
      Margin="5">
        <Image Height="21"
        Source="delete.gif"/>
      </Button >
    </stackpanel >
    <listbox/>
  </dockpanel >
</Window >

<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Simple Window">
  <Window.Resources>
    <solidcolorbrush x:Key="backgroundbrush">Yellow</solidcolorbrush>
    <solidcolorbrush x:Key="borderbrush">Red</solidcolorbrush>
  </Window.Resources >
  <Window.Background>
    <staticresource  resourcekey="backgroundbrush"/>
  </Window.Background >
  <dockpanel>
    <stackpanel  dockpanel.Dock="Bottom"
    Orientation="Horizontal"
    Horizontalalignment="Center">
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="zoom.gif"/>
      </Button >
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="defaultthumbnailsize.gif"/>
      </Button >
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="previous.gif"/>
      </Button >
```

```
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="slideshow.gif"/>
      </Button >
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="next.gif"/>
      </Button >
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="counterclockwise.gif"/>
      </Button >
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="clockwise.gif"/>
      </Button >
      <Button Background="{staticresource backgroundbrush}"
      Borderbrush="{staticresource  borderbrush}" Margin="5">
        <Image Height="21" Source="delete.gif"/>
      </Button >
    </stackpanel >
    <listbox/>
  </dockpanel >
</Window >
<lineargradientbrush x:Key="backgroundbrush" startpoint="0,0" endpoint="1,1">
  <gradientstop Color="Blue" Offset="0"/>
  <gradientstop Color="White" Offset="0.5"/>
  <gradientstop Color="Red" Offset="1"/>
```




```
</lineargradientbrush >
```

Now u can change brushes in one place and have different effects. Staticresource markup extension walks the logical tree or even application level or system resources. Root level or app level resource dict for max sharing. Closest to element chosen. For multi-threaded either Frozen or x:Shared=false.

Let's discuss static vs Dynamic Resources. Dynamicresource reapplied every time it changes nothing special about the resources. Only if you want to see updates or not. Dynamic more overhead but demand loaded. Dynamic can only be used to set dep. Prop. Values. Static can even abstract whole controls.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
<Image Height="21" Source="zoom.gif"/>
</Window >
```

Is equivalent to this Window :

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Window.Resources>
    <Image x:Key="zoom" Height="21" Source="zoom.gif"/>
```

```
  </Window.Resources >
  <stackpanel>
    <staticresource  resourcekey="zoom"/>
  </stackpanel >
</Window >
```

Can be a way to factor but Image can only have one parent so cant share last, static doesnt support fwd. Ref. With dynamic

The following is possible.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Simple Window" Background="{dynamicresource  backgroundbrush}">
  <Window.Resources>
    <solidcolorbrush x:Key="backgroundbrush">Yellow</solidcolorbrush>
    <solidcolorbrush x:Key="borderbrush">Red</solidcolorbrush>
  </Window.Resources >
</Window >
```

Without sharing, x:Shared=false e.g. New instance of image multiple times.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Window.Resources>
    <Image x:Shared="False" x:Key="zoom" Height="21" Source="zoom.gif"/>
  </Window.Resources >
  <stackpanel>
    <!--  Applying  the  resource  multiple  times  works! -->
    <staticresource  resourcekey="zoom"/>
    <staticresource  resourcekey="zoom"/>
    <staticresource  resourcekey="zoom"/>
  </stackpanel >
</Window >
```

On in procedural code

```
Window.Resources.Add("backgroundbrush", new solidcolorbrush(Colors.Yellow));
Window.Resources.Add("borderbrush", new solidcolorbrush(Colors.Red));

Button button= new Button();
// The Button must descend from the Window before looking up resources:
Stackpanel.Children.Add(button);
Button.Background = (Brush)button.findresource("backgroundbrush");
Button.borderbrush = (Brush)button.findresource("borderbrush");

Button button= new Button();
Button.setresourcereference(Button.backgroundproperty, "backgroundbrush");
Button.setresourcereference(Button.borderbrushproperty, "borderbrush");
Button button= new Button();
Button.Background = (Brush)window.Resources["backgroundbrush"];
Button.borderbrush = (Brush)window.Resources["borderbrush"];
```

Following are some XAML and equivalent C# examples.

```
XAML:
<Button  Background="systemcolors.windowbrush"/>
C\#:
Button.Background = (Brush)new
Brushconverter().convertfrom("systemcolors.windowbrush");
XAML:
```

```
<Button Background=" {x:Static systemcolors.windowbrush}"/>
C\#:
Button.Background = systemcolors.windowbrush;

XAML:
<Button Background=" { staticresource systemcolors.windowbrushkey}"/>
C\#:
Button.Background = (Brush)findresource("systemcolors.windowbrushkey");
XAML:
<Button Background=" { staticresource{x:Static systemcolors.windowbrush}}"/>
C\#:
Button.Background = (Brush)findresource(systemcolors.windowbrush);

XAML:
<Button Background=" { staticresource{x:Static systemcolors.windowbrushkey}}"/>
C\#:
Button.Background = (Brush)findresource(systemcolors.windowbrushkey);

XAML:
<Button Background=" { dynamicresource {x:Static systemcolors.windowbrushkey}}"/>
C\#:
Button.setresourcereference(
Button.backgroundproperty, systemcolors.windowbrushkey);
```

In summary, resources really imp. In professional apps, enable localization, increase productivity as it consolidates / shares. Most interesting use is with styles and data binding.

Chapter 29

Lecture 29

We will discuss data binding now. Data means an arbitrary .net obj. Data binding, data templates, data triggers are related concepts. Data can be collection obj, xml file, web service, db table, custom obj, even wpf element eg button. So data binding is typing together arbitrary objects. Classic scenario is a visual rep. (e.g. Listbox or datagrid) of items in an xml file, db, or in-memory collection. Instead of iterating and adding items, tell listbox to get its data from another source, keep them up to date, format them etc.

Binding binds two properties together and keeps a communication channel open. Setup Binding once and let it handle all the sync. Eg. In proc. Code.

```
<textblock x:Name="currentfolder" dockpanel.Dock="Top"
Background="aliceblue" fontsize="16" />

Void  treeview_selecteditemchanged(object sender,
Routedpropertychangedeventargs<object> e)
{
Currentfolder.Text =
(treeview.selecteditem as treeviewitem).Header.tostring();
Refresh();
}
Public  mainwindow()
{
Initializecomponent();
Binding binding= new Binding();
// Set source object
Binding.Source = treeview;
// Set source property
Binding.Path= new propertypath("selecteditem.Header");
// Attach to target property
Currentfolder.setbinding(textblock.textproperty, binding);
}
```

When an item with no header selected, then a default value (in this case) returned. No exception is raised. Binding has a source prop and a target prop. Source is an object and a prop. Path. Can also use bindingoperations.setbinding. Bindingoperations.clearbinding. Bindingoperations.clearallbindings. Or set the target prop to a new val. (but clear would allow receiving values from lower prop. Sources).

There is a binding markup extension as well.

```
<textblock x:Name="currentfolder" dockpanel.Dock="Top"
Text="{Binding elementname=treeview, Path=selecteditem.Header}"
Background="aliceblue" fontsize="16" />
<textblock x:Name="currentfolder" dockpanel.Dock="Top"
Text="{Binding Source={x:Reference  treeview}, Path=selecteditem.Header}"
Background="aliceblue" fontsize="16" />
<textblock Text="{Binding targetnullvalue=Nothing is selected.}" />
<Label x:Name="numitemslabel"
Content="{Binding Source={staticresource  photos}, Path=Count}"
Dockpanel.Dock="Bottom"/>
System.componentmodel.inotifypropertychanged
<Label x:Name="numitemslabel"
```

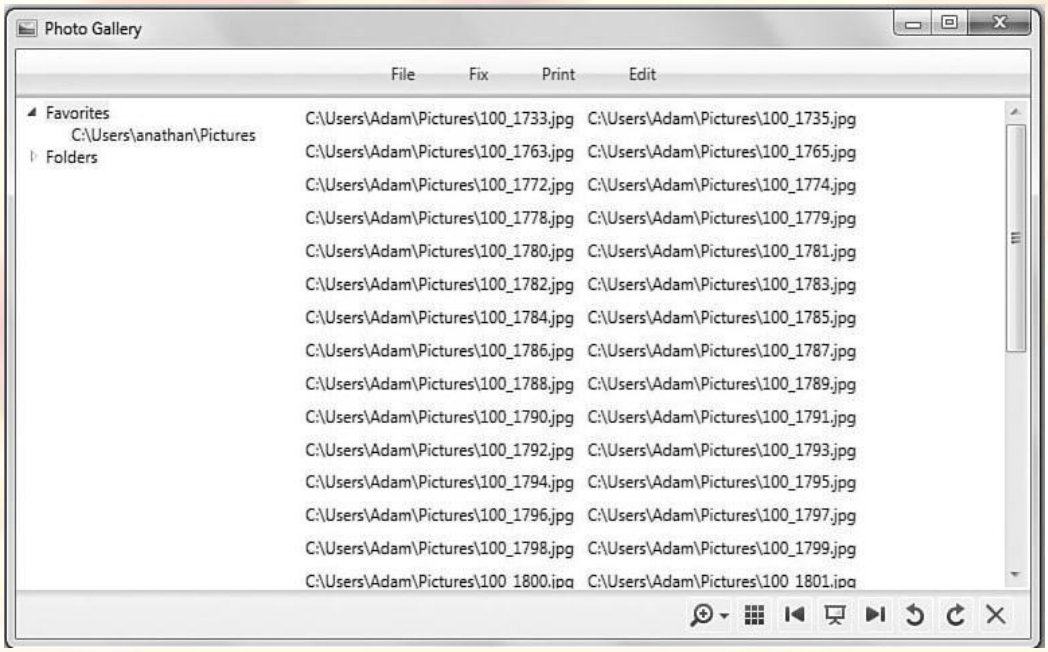
```
Content="{Binding Source={staticresource  photos}}"
Dockpanel.Dock="Bottom"/>
```

Could also set Source but using elementname is easier. Can also set value for nothing is selected.

We can also bind to plain .net properties. But the source obj should implement the System.component model. notify property interface, which has a single propertychanged event. (the right way) and Implement an xxxchanged event, where XXX is the name of the property whose value changed. Public class Photos : observablecollection¡Photo¿ target must be a dep. Prop. When binding to an object, Path is optional

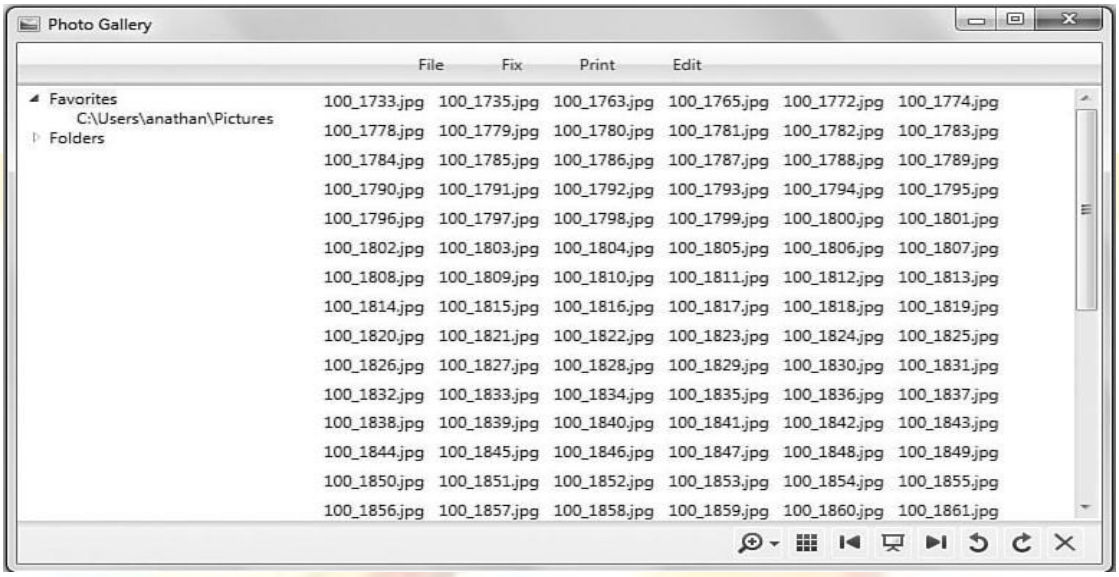

```
<listbox x:Name="picturebox"
Itemssource="{Binding Source={staticresource  photos}}">
</listbox >
```

We can also bind to a collection. How abt binding listbox in photo app to photo collection. When a new dir selected, clear lisbox and created listboxitem for each item. When user del or rename, an event (because of filesystemwatcher) and manually refresh. Raw binding i.e. Items is not dep. Prop but itemsource is. Source prop must impl. Inotifycollectionchanged (aka observablecollection). The simplest



The displaymemberpath property can be used to improve display.

```
<listbox x:Name="picturebox" displaymemberpath="Name"
Itemssource="{Binding Source={staticresource photos}}">
</listbox >
```



We cant mix Items and itemsource. But we always retrieve from Items. We could add an Image property but there are more flexible ways. 2 ways: data template and value converter. Selected item can also be synchronized but scrolling not synchronized, only first selection synchronized.

```
<listbox  issynchronizedwithcurrentitem="True" displaymemberpath="Name"
Itemssource="{Binding Source={staticresource  photos}}"></listbox>
<listbox issynchronizedwithcurrentitem="True" displaymemberpath="datetime"
Itemssource="{Binding Source={staticresource  photos}}"></listbox>
<listbox  issynchronizedwithcurrentitem="True" displaymemberpath="Size"
Itemssource="{Binding Source={staticresource  photos}}"></listbox>
```

Implicit data source is provided by a Data Context. We set datacontext of a parent and then dont specify Source or elementname or set parent.datacontext = photos;. It is useful when plugging in resources: usage context or decl.context.

```
<stackpanel  datacontext="{staticresource  photos}">
<Label x:Name="numitemslabel"
Content="{Binding Path=Count}"/>
<listbox x:Name="picturebox" displaymemberpath="Name"
Itemssource="{Binding}">
</listbox >
</stackpanel >
```

Control rendering is easy without binding but there are benefits of data binding. WPF gives three ways. String formatting which only works if the target prop is a string. {} to escape. Can add 1000 seperator etc.

```
<textblock x:Name="numitemslabel"
Text="{Binding  stringformat={}{0} item(s),Source={staticresource
photos},Path=Count}"dockpanel.Dock="Bottom"/>
<textblock x:Name="numitemslabel" dockpanel.Dock="Bottom">
<textblock.Text>
<Binding Source="{staticresource  photos}" Path="Count">
<Binding.stringformat>{0} item(s)</Binding.stringformat>
</Binding >
</textblock.Text >
</textblock >
```

String formatting can be used even without data binding.

```
<listbox itemstringformat="{}{0:C}" Xmlns:sys="clr-
namespace:System;assembly=mscorlib">
  <sys:Int32>-9</sys:Int32>
  <sys:Int32>9</sys:Int32>
  <sys:Int32>1234</sys:Int32>
  <sys:Int32>1234567</sys:Int32>
</listbox >
```

TABLE 13.1 String Format Properties Throughout WPF

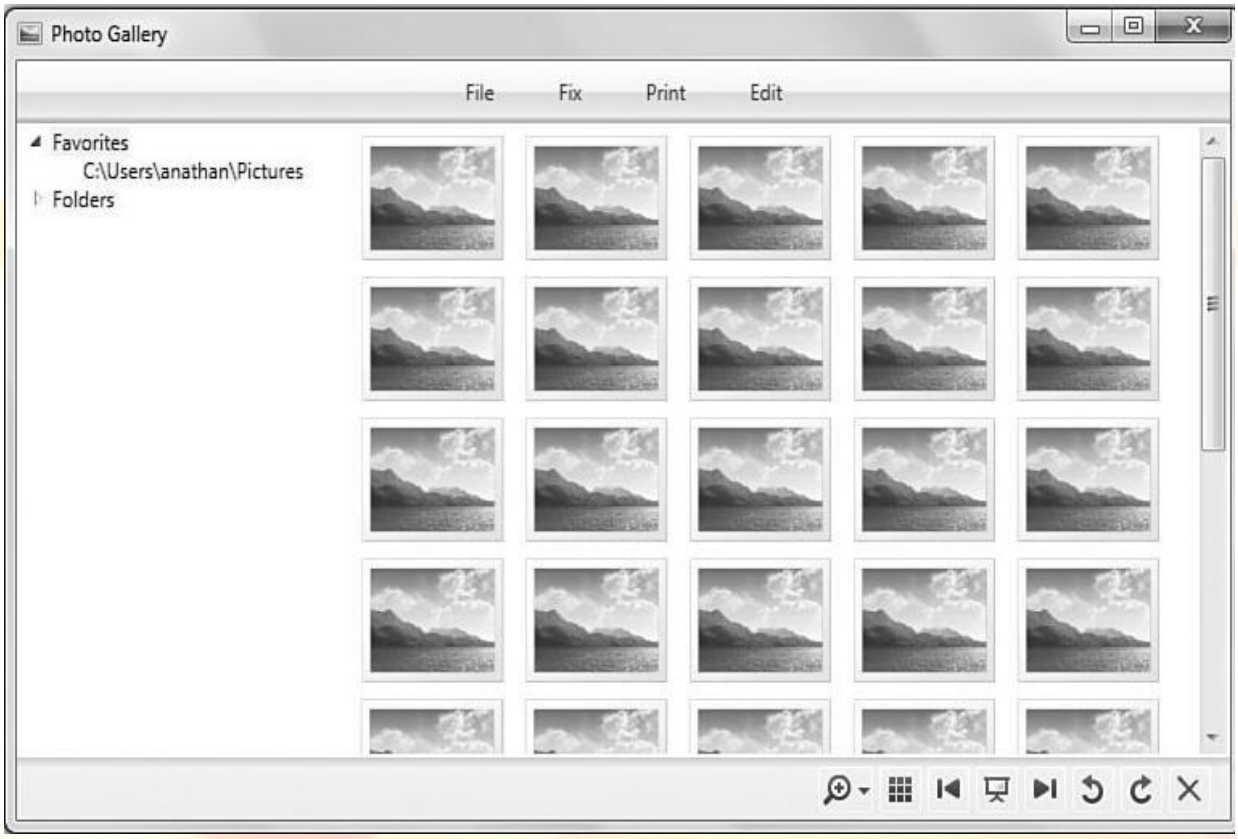
Property	Classes
StringFormat	BindingBase
ContentStringFormat	ContentControl, ContentPresenter, TabControl
ItemStringFormat	ItemsControl, HierarchicalDataTemplate
HeaderStringFormat	HeaderedContentControl, HeaderedItemsControl, DataGridColumn, GridViewColumn, GroupStyle
ColumnHeaderStringFormat	GridView, GridViewHeaderRowPresenter

With Data Template, UI auto-applied to arbitrary .net object when it is rendered. By setting these prop. You can swap in a complete new visual tree easy in xaml, cumbersome in proc. Code try with photo gallery

TABLE 13.2 Properties of Type DataTemplate Throughout WPF

Property	Classes
ContentTemplate	ContentControl, ContentPresenter, TabControl
ItemTemplate	ItemsControl, HierarchicalDataTemplate
HeaderTemplate	HeaderedContentControl, HeaderedItemsControl, DataGridRow, DataGridColumn, GridViewColumn, GroupStyle
SelectedContentTemplate	TabControl
DetailsTemplate	DataGridRow
RowDetailsTemplate	DataGrid
RowHeaderTemplate	DataGrid
ColumnHeaderTemplate	GridView, GridViewHeaderRowPresenter
CellTemplate	DataGridTemplateColumn, GridViewColumn
CellEditingTemplate	DataGridTemplateColumn

```
<listbox x:Name="picturebox"
Itemssource="{Binding Source={staticresource  photos}}">
<listbox.itemtemplate>
<datatemplate>
<Image Source="placeholder.jpg" Height="35"/>
</datatemplate >
</listbox.itemtemplate >
</listbox >
```



With a template there is implicit datacontext. Template can be used on non-data bound but inside the element it almost always make sense to use data binding. Template can be shared as resources. Template can even be auto-applied to some type by setting its datatype prop. There is also hierarchicaldatatemplate that understands hierarchies. Can be used with Treeview or menu.

```
<listbox x:Name="picturebox"
Itemssource="{Binding Source={staticresource  photos}}">
<listbox.itemtemplate>
<datatemplate>
<Image Source="{Binding Path=fullpath}" Height="35"/>
</datatemplate >
</listbox.itemtemplate >
</listbox >
```

Value Convertors morph source value into something else. Introduce custom logic and can change type etc. E.g.brush based on some enumeration. Can be used to customize display.

```
<Window.Resources>
<local:counttobackgroundconverter x:Key="myconverter"/>
</Window.Resources >

<Label Background="{Binding Path=Count, Converter={staticresource myconverter},
Source={staticresource photos}}" />
Public class  counttobackgroundconverter : ivalueconverter
{
Public object Convert(object value, Type targettype,
            object parameter, cultureinfo  culture)
{
If (targettype != typeof(Brush))
Throw new invalidoperationexception("The target must be a Brush!");
// Let Parse throw an exception if the input is bad
Int num = int.Parse(value.tostring());
Return(num == 0 ? Brushes.Yellow: Brushes.Transparent);
}
Public object convertback(object value, Type targettype,
Object parameter, cultureinfo culture)
{
Return  dependencyproperty.unsetvalue;
}
}
```

Chapter 30

Lecture 30

In last lecture, we discused data binding, binding object, markup extension, bind to any prop with inotifypropertychanged, target must be dependency property, bind to obj. So no property path, binding to collection, issynchronized with current item, datacontext, displaymemberpath, stringformat, datatemplate, valueconverter.

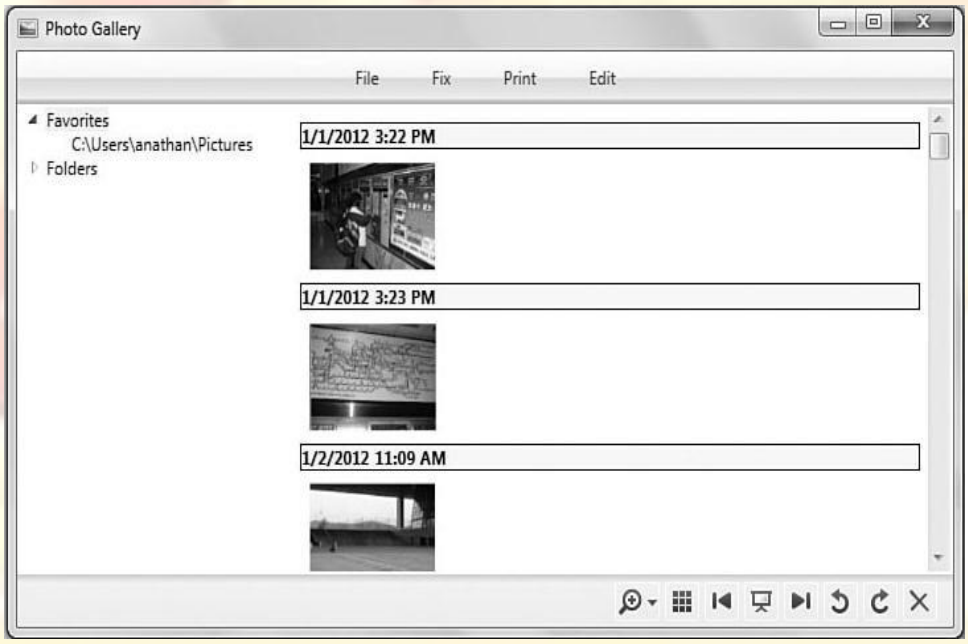
Let's discuss customizing collection view. When issynchronizedwithcurrentitem= true, where is the current item. A default "view" inserted. Object implementing icollectionview. Has also support for sorting, grouping, filtering, and navigating. Let's discuss these four and multiple views for the same source obj.

Sortdescriptions prop. Is a collection of sortdescription which chooses a field and order, sort by datetime, then by name. A clear method to return to unsorted. Eg. Three buttons to sort and toggle.

```
Sortdescription sort = new sortdescription("Name", listsortdirection.Ascending);

View.sortdescriptions.Add(new sortdescription("datetime", listsortdirection.Descending);
View.sortdescriptions.Add(new sortdescription("Name", listsortdirection.Ascending);

// Click event handlers for three different Buttons:
Void sortbyname_Click(object sender, routedeventargs e)
{
Sorthelper("Name");
}
Void sortbydatetime_Click(object sender, routedeventargs e)
{
Sorthelper("datetime");
}
Void sortbysize_Click(object sender, routedeventargs e)
{
Sorthelper("Size");
}
Void sorthelper(string  propertyname)
{
// Get the default view
Icollectionview view = collectionviewsource.getdefaultview(
This.findresource("photos"));
// Check if the view is already sorted ascending by the current  property
If (view.sortdescriptions.Count > 0
&& view.sortdescriptions[0].propertyname == propertyname
&& view.sortdescriptions[0].Direction == ", listsortdirection.Ascending)
{
// Already sorted ascending, so t o g g l e by sorting descending
View.sortdescriptions.Clear();
View.sortdescriptions.Add(new sortdescription(
Propertyname, listsortdirection.Descending));
}
Else
{
// Sort ascending
View.sortdescriptions.Clear();
View.
```

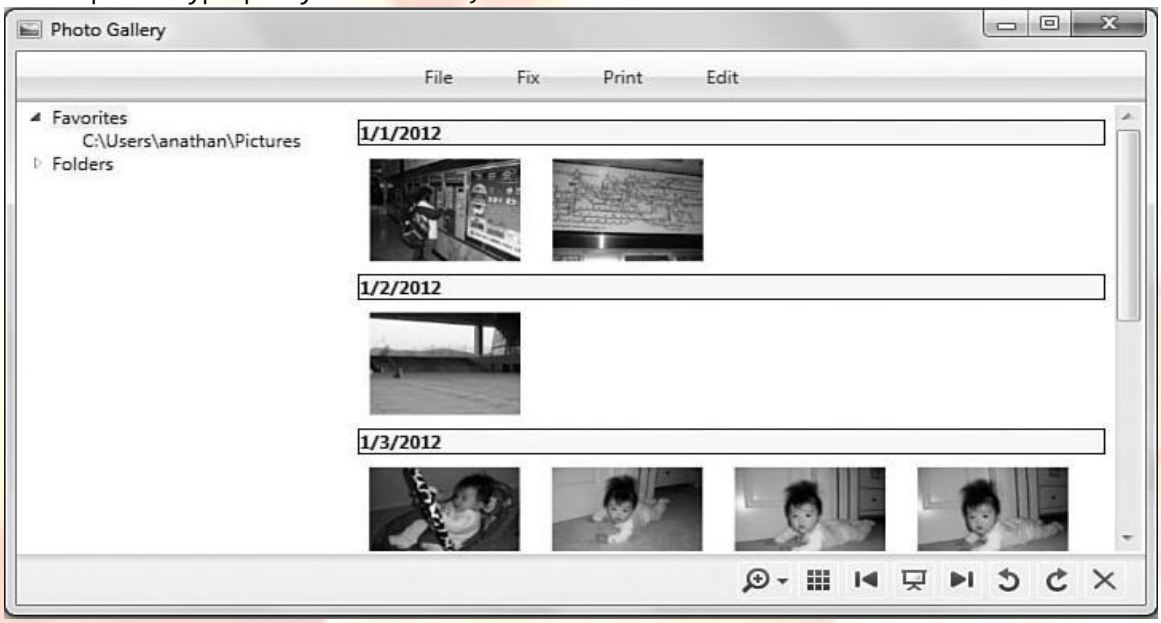
```
Tdescriptions.Add(new sortdescription(
Propertyname, listsortdirection.Ascending));
}
}
// Get the default view
Icollectionview view  = collectionviewsource.getdefaultview(
This.findresource("photos"));
// Do the grouping 46 view.groupdescriptions.Clear();
View.groupdescriptions.Add(new propertygroupdescription("datetime"));
<liszbox x:Name="picturebox"
Itemssource="{Binding Source={staticresource  photos}}">
<listbox.groupstyle>
<groupstyle>
<groupstyle.headertemplate>
<datatemplate>
<Border  borderbrush="Black" borderthickness="1">
<textblock Text="{Binding Path=Name}" fontweight="Bold"/>
</Border >
</datatemplate >
</groupstyle.headertemplate >
</groupstyle >
</listbox.groupstyle >
</listbox >
```

There is no explicit relationship with listbox. If additional controls bound, they would sort with it.Now groupdescriptions prop. Containing propertygroupdescription obj. But it has no effect without the groupstyle prop.




```
<listbox x:Name="picturebox"
Itemssource="{Binding Source={staticresource  photos}}">
<listbox.groupstyle>
<x:Static Member="groupstyle.Default"/>
</listbox.groupstyle >
</listbox >
// Get the default view
```

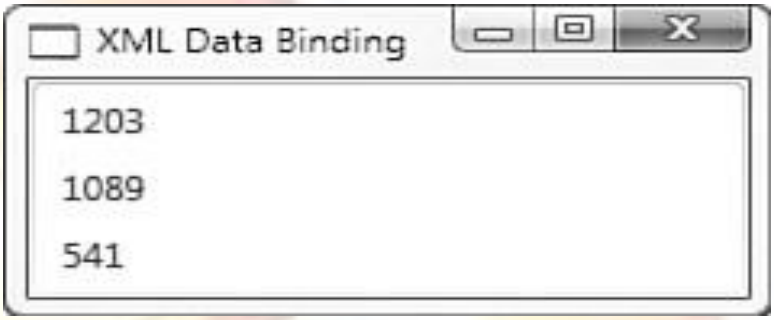
```
Icollectionview view = collectionviewsource.getdefaultview(
This.findresource("photos"));
// Do the grouping
View.groupdescriptions.Clear();
View.groupdescriptions.Add(
New propertygroupdescription("datetime", new datetimetodateconverter()));
Public class datetimetodateconverter : ivalueconverter
{
Public object Convert(object value, Type targettype, object parameter,
Cultureinfo  culture)
{
Return((datetime)value).tostring("MM/dd/yyyy");
}
Public object  convertback(object value, Type targettype, object parameter,
Cultureinfo  culture)
{
Return  dependencyproperty.unsetvalue;
}
}
```




```
Icollectionview view = collectionviewsource.getdefaultview(
This.findresource("photos"));
View.Filter = delegate(object o) {
Return((o as Photo).datetimedatetime.Now).Days <= 7;
};
Icollectionview view = collectionviewsource.getdefaultview(
This.findresource("photos"));
View.Filter = (o)=> { return (
(o as Photo).datetime datetime.Now).Days <= 7;
};
Void previous_Click(object sender, routedeventargs e)
{
// Get the default view
Icollectionview view = collectionviewsource.getdefaultview(
This.findresource("photos"));
// Move backward
View.movecurrenttoprevious();
// Wrap around to the end
```

```
If (view.iscurrentbeforefirst) view.movecurrenttolast();
}
Void next_Click(object sender, routedeventargs e)
{
// Get the default view
Icollectionview view = collectionviewsource.getdefaultview(
This.findresource("photos"));
// Move forward
View.movecurrenttonext();
// Wrap around to the beginning
If (view.iscurrentafterlast) view.movecurrenttofirst();
}
"{Binding Path=/}"
"{Binding Path=/datetime}"
"{Binding Path=Photos/}"
"{Binding Path=Photos/datetime}"
Collectionviewsource viewsource= new collectionviewsource();
Viewsource.Source = photos;
<Window.Resources>
<local:Photos x:Key="photos"/>
<collectionviewsource x:Key="viewsource" Source="{staticresource  photos}"/>
</Window.Resources >
<listbox x:Name="picturebox"
Itemssource="{Binding Source={staticresource  photos  viewsource}}" >
</listbox >
<collectionviewsource x:Key="viewsource"
Filter="viewsource_Filter"
Source="{staticresource  photos}">
<collectionviewsource.sortdescriptions>
<componentmodel:sortdescription  propertyname="datetime"
Direction="Descending"/>
</collectionviewsource.sortdescriptions >
<collectionviewsource.groupdescriptions>
<propertygroupdescription  propertyname="datetime"/>
</collectionviewsource.groupdescriptions >
</collectionviewsource >
Void viewsource_Filter(object sender, filtereventargs e)
{
E.Accepted = ((e.Item as Photo).datetime datetime.Now).Days <= 7;
}
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="XML Data Binding">
<Window.Resources>
<xmldataprovider x:Key="dataprovider" xpath="gamestats">
<x:xdata>
<gamestats xmlns="">
<!--  One  stat  per  game  type-->
<gamestat Type="Beginner">
<highscore>1203</highscore>
</gamestat >
<gamestat Type="Intermediate">
<highscore>1089</highscore>
</gamestat >
<gamestat Type="Advanced">
<highscore>541</highscore>
</gamestat >
```

```
</gamestats >
</x:xdata >
</xmldataprovider >
</Window.Resources >
<Grid>
<listbox itemssource="{Binding
Source={staticresource dataprovider},
Xpath=gamestat/highscore}"/>
</Grid >
</Window >
```



Sorting is applied before grouping. First sorting criteria should be same as grouping otherwise output doesnt make much sense. Can pass null to get total custom control in the value converter.

Filtering gives a property Filter of type Predicate¡Object¿. Its null by default e.g. Show only photos from last 7 days.

Navigation means managing the current item, not the other kind of nav. Icollectionview has currentitem currentposition and also methods for changing them. E.g. Prev/next photo buttons are handled like this view initializes to 0 and first item listbox initializes to -1 and null (unselected).

Prop. Paths in Bindings are useful for master/detail interfaces. Sorting, grouping, filtering automatic. But navigation only when issynchronizedwithcurrentitem=true. Otherwise selecteditem and currentitem are separate. Collectionviewsource can be used to create new views and applied to targets. Collectionviewsource has its own sortdescriptions and groupdescriptions properties and a Filter event to be used from xaml. Must include another namespace. Issynchronizedwithcurrentitem=true by default for custom views. Have to explicitly set false. Kind of a bit inconsistent. But we acknowledging view existance by custom view.

Let's discuss data providers. Source obj can be arbitrary. You could bind to db, registry, excel spreadsheet etc. With enough code. An obj. That exposes right props and notifications and handles messy details work involved might overweight benefits if writing all logic yourself two generic data-binding-friendly way to expose common items. Xmldataprovider and objectdataprovider. Starting with wpf 3.5 sp1 data binding works with LINQ (language independent query) you can set Source or datacontext to a LINQ and the enumerable result is used. Now with LINQ to SQL and LINQ to XML, is an easy way than using wpf data providers.

We use xdata to avoid compiler errors. Namespace pollution avoided with xmlns=. Bindings xpath prop used rather than Path. If external file, even easier. "gamestat/@Type" would fill list with each gamestats Type attribute. Data provided in xmlnode etc. Objects from System.Xml so you can use Path and xpath together. Needs hierarchical data template if you want to bind to xml tree.

```
<xmldataprovider x:Key="dataprovider" xpath="gamestats" Source="gamestats.xml"/>
<Label Content="{Binding Source={staticresource dataprovider},
Xpath=gamestat/highscore, Path=outerxml}"/>
```

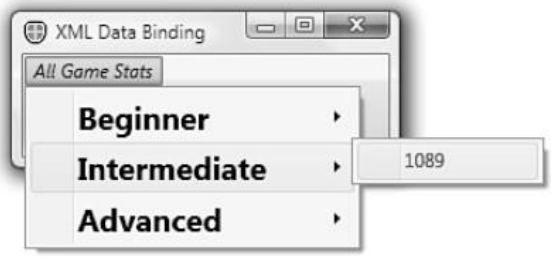
Chapter 31

Lecture 31

In the last lecture, we discussed the view between binding source and target sorting, grouping, filtering, navigating. Creating additional views, data providers (xml and object), used xml data provider to fill a listbox, now use hierarchical data template.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="XML Data Binding">
  <Window.Resources>
    <hierarchicaldatatemplate datatype="gamestats"
    Itemssource="{Binding xpath=*}">
      <textblock fontstyle="Italic"
      Text="All Game Stats"/>
    </hierarchicaldatatemplate >
    <hierarchicaldatatemplate  datatype="gamestat"
    Itemssource="{Binding xpath=*}">
      <textblock fontweight="Bold" fontsize="20"
      Text="{Binding xpath=@Type}"/>
    </hierarchicaldatatemplate >
    <datatemplate  datatype="highscore">
      <textblock  Foreground="Blue" Text="{Binding xpath=.}"/>
    </datatemplate >
    <xmldataprovider x:Key="dataprovider" xpath="gamestats">
      <x:xdata>
        <gamestats xmlns="">
          <!--  One  stat  per  game  type-->
          <gamestat Type="Beginner">
            <highscore>1203</highscore>
          </gamestat >
          <gamestat Type="Intermediate">
            <highscore>1089</highscore>
          </gamestat >
          <gamestat Type="Advanced">
            <highscore>541</highscore>
          </gamestat >
        </gamestats >
      </x:xdata >
    </xmldataprovider >
  </Window.Resources >
  <Grid>
    <treeview
    Itemssource="{Binding Source={staticresource dataprovider},
Xpath=.}"/>
  </Grid >
</Window >
```

The TreeView in Listing 13.3



Changing TreeView to Menu

Hierarchicaldatatemplate for every node and datatemplate for a leaf node. Hierarchicaldatatmplate allow specifyingchildren using itemsource prop. =all children. Datatype means effect all instances within the scope. Datatype corresponds to xmlnode name. No key. Internally datatype value used as key.

```
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom"
Xmlns:georss="http://www.georss.org/georss">
</rss >
Xmldataprovider  Source="http://twitter.com/statuses/user_timeline/24326956.rss"
Xmlnamespacemanager="{staticresource  namespacemapping}"
Xpath="rss/channel" x:Key="dataprovider"/>
<xmlnamespacemappingcollection x:Key="namespacemapping">
  <xmlnamespacemapping Uri="http://www.w3.org/2005/Atom" Prefix="atom"/>
  <xmlnamespacemapping Uri="http://www.georss.org/georss" Prefix="georss"/>
</xmlnamespacemappingcollection >
"{Binding xpath=atom:link}"
```

Now Let's discuss objectdataprovider. .net object as a data source. So what does that add against binding directly to a .net obj. Declaratively instantiate the source object with a parameterized constructor. Bind to a method on the source object. Have more options for asynchronous data binding. When binding not quick, async soui doesnt stuck. Wpf has two ways to mark async. Isasync prop of Binding, xmldataprovider and objectdataprovider have an isasynchronous prop. False by default on objdp, true by default for xmldp. Create source obj on background thread. When isasync ture (false by default), source prop invoked on background thread property getters supposedto be fast so shouldn't need.

First example wraps photos. Its the same, even binding path because binding "unwraps". It can also create the object given its type.

```
<Window.Resources>
<local:Photos x:Key="photos"/>
<objectdataprovider x:Key="dataprovider"
Objectinstance="{staticresource  photos}"/>
</Window.Resources >
<Window.Resources>
<!--  The  collection  object  is  instantiated  internally  by  objectdataprovider:-->
<objectdataprovider x:Key="dataprovider" objecttype="{x:Type local:Photos}"/>
</Window.Resources >
```

Binding to a method is useful for classes that are not designed for data binding. Imagine photos class had getfoldername method. Can use methodparameters. If Path used. It would apply to the obj returned.

```
<objectdataprovider x:Key="dataprovider" objecttype="{x:Type local:Photos}">
```

```
<objectdataprovider.constructorparameters>
<sys:Int32>23</sys:Int32>
</objectdataprovider.constructorparameters >
</objectdataprovider >
<objectdataprovider x:Key="dataprovider"
Objecttype="{x:Type local:Photos}"
Methodname="getfoldername"/>
```

Binding.Mode can be oneway i.e. The target is updated whenever the source changes. Twoway i.e. A change to either the target or source updates the other. Onewaytosource. This is the opposite of oneway. The source is updated whenever the target changes. Onetime. This works just like oneway, except changes to the source are not reflected at the target. The target retains a snapshot of the source at the time the Binding is initiated.

Useful for editable datagrid and textbox (by default) thats why convertback method. Only convertback on oneway-tosource.

Updatesourcetrigger. Do you want the two way source to be updated on every key stroke. Propertychanged. Changed when target prop value changes. Lostfocus. Changed when focus lost. Explicit. Call bindingexpression.updatesource which you can get from any frameworkelement.getbindingexpression

We can also have validation rules. Want to validate as soon as possible for user feedback. Without binding you could insert custom logic but now the data is auto-pushed to target. Let's say we want valid .jpg file. 2 ways. Validation rule or use source exceptions. Binding has validationrules prop that can be set to one or more validationrule derived objects.

```
<textbox>
<textbox.Text>
<Binding >
<Binding.validationrules>
<local:jpgvalidationrule/>
</Binding.validationrules >
</Binding >
</textbox.Text >
</textbox >
Public class  jpgvalidationrule : validationrule
{
Public override  validationresult  Validate(object value, cultureinfo  cultureinfo)
{
String filename = value.tostring();
If(!File.Exists(filename))
Return new validationresult(false, "Value is not a valid file.");
If (!Filename.endswith(".jpg", stringcomparison.invariantcultureignorecase))
Return new validationresult(false, "Value is not a .jpg file.");
Return new validationresult(true, null);
}
}
```

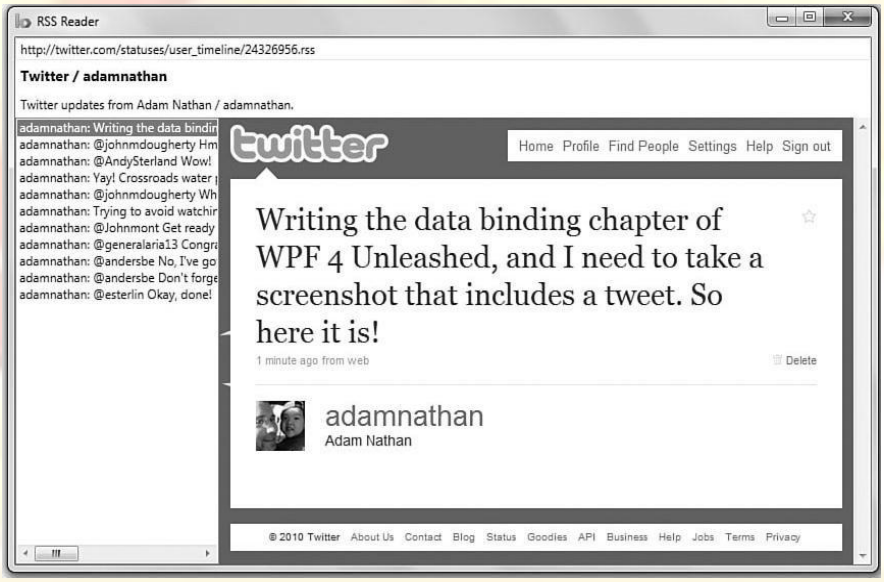
Validation check whenever attempt to update target. In this case on lostfocus because it is default. When invalid, an error adorner rendered on top of element. By default thin red border but can use Validation.errortemplate attached prop on target element to customize it, also other properties for custom control.

```
<textbox>
<textbox.Text>
<Binding >
```

```
<Binding.validationrules>
<exceptionvalidationrule/>
</Binding.validationrules >
</Binding >
</textbox.Text >
</textbox >
<textbox>
<textbox.Text>
<Binding >
<Binding.validationrules>
<exceptionvalidationrule/>
<dataerrorvalidationrule/>
</Binding.validationrules >
</Binding >
</textbox.Text >
</textbox >
<textbox>
<textbox.Text>
<Binding  validatesonexceptions="True" validatesondataerrors="True" />
</textbox.Text >
</textbox >
```

But what if already exceptions thrown by target. Also the source can implement System.componentmodel.idataerrorinfo. Easier way in wpf 3.5 sp1.

Let's write an RSS reader without code. A two-way textbox so that user can change feed address. Bindsdirectlytosource so path does not refer to the rss feed. Updatesourcetrigger of propertychanged so update with every keystroke. Listbox/frame share the source. Master detail.




```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="RSS Reader">
  <Window.Resources>
    <xmldataprovider x:Key="Feed"
    Source="http://twitter.com/statuses/user_timeline/24326956.rss"/>
  </Window.Resources >
  <dockpanel
```

```
  Datacontext="{Binding Source={staticresource Feed},
Xpath=/rss/channel/item}">
    <textbox  dockpanel.Dock="Top"
    Text="{Binding Source={staticresource Feed},
Bindsdirectlytosource=true, Path=Source,
Updatesourcetrigger=propertychanged}"/>
    <Label dockpanel.Dock="Top"
    Content="{Binding xpath=/rss/channel/title}"
    Fontsize="14" fontweight="Bold"/>
    <Label dockpanel.Dock="Top"
    Content="{Binding xpath=/rss/channel/description}"/>
    <listbox  dockpanel.Dock="Left" displaymemberpath="title"
    Itemssource="{Binding}"
    Issynchronizedwithcurrentitem="True" Width="300"/>
    <Frame Source="{Binding xpath=link}"/>
  </dockpanel >
</Window >
```

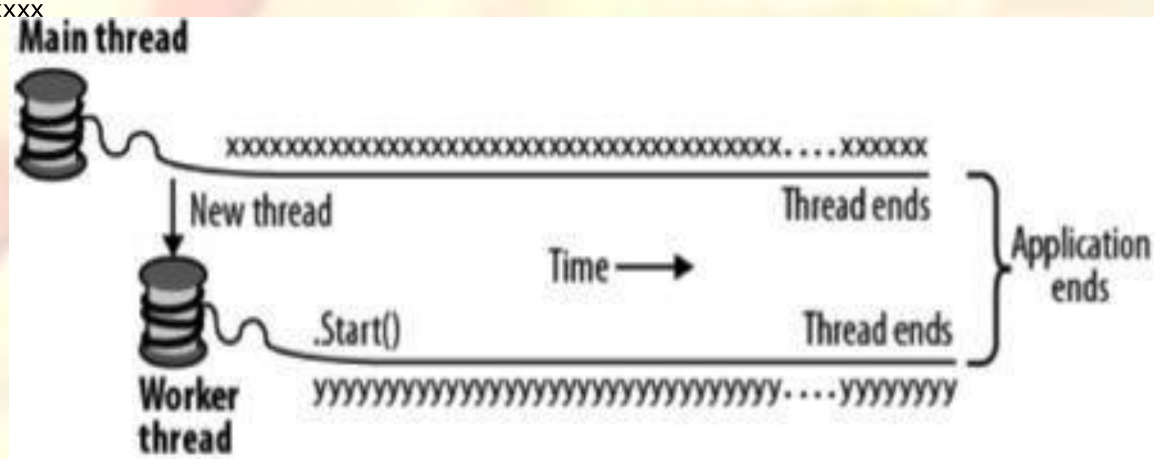
Chapter 32

Lecture 32

Concurrency is More than one thing happening at the same time. Comparison with events. Why still concurrency. Responsive user interface. Simultaneous requests. Parallel programming.

Thread: execution path that can proceed independently of others. Normal processes (programs in exec.) Have one thread. Multithreaded programs have more and can share data. Let's create one.

```
Class threadtest
{
Static void Main()
{
Thread t = new Thread(writey);
T.Start();

For(int i = 0; i < 1000; i++) Console.Write("x");
}

Static void writey()
{
For(int i = 0; i < 1000; i++) Console.Write("y");
}
}

Typical Output:
Xxxxxxxxxxxxxyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyxxxxxxxxxxxxxxxxxxx
xxxxxxx
```



Interleaved because of time slicing. Real parallel on multicore but still repeated blocks cuz of the way Console handles it. A thread is pre-empted because of time slicing. A threads isalive=true once starts and until ends. Name property, Thread.currentthread is the currently executing thread.

Join and Sleep. A thread is "blocked" when waiting. Can use threadstate property.

```
Static void Main()
{
Thread t = new Thread(Go);
```

```
T.Start();
T.Join();

Console.writeline("Thread t has ended!");
}
Static void Go(){ for(int i= 0; i < 1000; i++) Console.Write("y"); }

Thread.Sleep(timespan.fromhours(1));
Thread.Sleep(500);
```

Local vs Shared state.

```
Static void Main()
{
New Thread(Go).Start();
Go();
}
Static void Go()
{
For(int cycles = 0; cycles< 5; cycles++) Console.Write('?');
}
Class threadtest
{
Bool _done;
Static void Main()
{
Threadtest tt= new threadtest();
New Thread(tt.Go).Start();
Tt.Go();
}
Void Go()
{
If(!_done) { _done = true; Console.writeline("Done");}
}
}

Class threadtest
{
Static bool _done;
Static void Main()
{
New Thread(tt.Go).Start();
Go();
}
Static void Go()
{
If(!_done) { _done = true; Console.writeline("Done");}
}
}

Class threadtest
{
Static void Main()
{
Bool done= false;
Threadstart action= () =>
{
If(!Done) { done= true; Console.writeline("Done"); }
```

```
};
New Thread(action).Start();
Action();
}
}
```

Let's discuss thread safety. In this code, we can output Done twice. And the odds increase if statements are reversed.

```
Class threadsafe
{
Static bool _done;
Static readonly object_locker = new object();
Static void Main()
{
New Thread(Go).Start();
Go();
}
Static void Go()
{
Lock (_locker)
{
If(!_done){ Console.writeline("Done"); _done = true; }
}
}
}
```

Passing Data to Threads. Use lambda expression (easiest) or use object param.

```
Static void Main()
{
Thread t = new Thread(() => Print("Hello from t!"));
T.Start();
}
Static  void  Print(string  message) {  Console.writeline(message);}

New Thread (()=>
{
Console.writeline ("I'm running on another thread!");
Console.writeline ("This is so easy!");
}).Start()

Static void Main()
{
Thread t = new Thread(Print);
T.Start("Hello from t!");
}
Static void Print(object  messageobj)
{
String message = (string)messageobj;
Console.writeline(message);
}

Public delegate void threadstart();
Public delegate void  parameterizedthreadstart(object obj);
```

Lambda expressions and captured vars

```
For(int i= 0; i < 10; i++)
New Thread (()=> Console.Write(i)).Start();
Typical Output:
0223557799
```

```
For(int i= 0; i < 10; i++)
{
Int temp = i;
New Thread (()=> Console.Write(temp)).Start();
}
```

Exceptions

```
Public static void Main()
{
Try
{
New Thread(Go).Start();
}
Catch (Exception ex)
{
Console.writeline("Exception!");
}
}
Static void Go(){ throw null;}
```

```
Public static void Main()
{
New Thread (Go).Start();
}
Static void Go()
{
Try
{
//...
Throw null;
//...
}
Catch (Exception ex)
{
//...
}
}
```

There are isbackground and Priority properties. Signaling between threads using a manualresetevent (simplest).

Waitone and Set. Reset closes the signal.

```
Var signal = new manualresetevent(false);
New Thread (()=>
{
Console.writeline ("Waiting for signal...");
Signal.waitone();
Signal.Dispose();
Console.writeline ("Got signal!");
}).Start();
Thread.Sleep(2000);
Signal.Set();
```

Chapter 33

Lecture 33

Long rnning operations make application unresponsive. Because main thread used for rendering UI and responding to events. Start up worker thread and update UI when finished. But UI updation usually possible only on UI thread. So, forward the request to UI thread.(or marshal it). Low level way is to call begininvoke or Invoke on the elements Dispatcher object. It takes a delegate and queues it on the UI thread. Invoke does same but then blocks until it is done. So you can return a value. But if you dont need begininvoke is better.

```
Partial class mywindow: Window
{
Public mywindow()
{
Initializecomponent();
New Thread(Work).Start();
}
Void Work()
{
Thread.Sleep(5000);
Updatemessage("The answer");
}
Void updatemessage(string message)
{
Action action= ()=> txtmessage.Text = message;
Dispatcher.begininvoke(action);
}
}
```

Synchronizationcontext can be used for generalized thread marshaling.

```
Partial class mywindow: Window
{
Synchronizationcontext _uisynccontext;
Public mywindow()
{
Initializecomponent();
_uisynccontext = synchronizationcontext.Current;
New Thread(Work).Start();
}
Void Work()
{
Thread.Sleep(5000);
Updatemessage("The answer");
}
Void updatemessage(string message)
{
_uisynccontext.Post(_ => txtmessage.Text = message);
}
}
```

Threadpool save time of thread creation. Cant name thread and difficult debugging. Always background. Blocking can degrade performance. Thread.currentthread.isthreadpoolthread property. Threadpool creates or reduces real threads using a hillclimbing algo to maximize cpu usage and reduce slicing.

```
Threadpool.queueuserworkitem (notused => Console.writeline ("Hello"));
Task.Run (()=> Console.writeline ("Hello from the thread pool"));
```

Let's discuss Tasks. No easy way to get return value from a thread. We can join and use shared data. Exceptions difficult too. Cant tell to start something else when finished. Task: a higher level abstraction. Tasks can be chained using continuations. Can use threadpool. With taskcompletionsource callback approach.

Starting a task is like creating a thread. Except started right away (hot) and in a thread pool. Task.wait is like Join. Task has generic subclass Task¡treturn¿. Task.Result blocks.

```
Task.Run (()=> Console.writeline ("Foo"));
New Thread (()=> Console.writeline ("Foo")).Start();

Task task= Task.Run (()=>
{
Thread.Sleep (2000);
Console.writeline ("Foo");
});
Console.writeline (task.iscompleted);
Task.Wait();

Task<int> task = Task.Run(()=>{ Console.writeline("Foo"); return3; });
// ...
```

Tasks propagate exception to whoever calls wait or accesses Result

```
Int result = task.Result;
Console.writeline (result);
Task task= Task.Run (()=>{ throw null;});
Try
{
Task.Wait();
}
Catch (aggregateexception aex)
{
If (aex.innerexception is nullreferenceexception)
Console.writeline ("Null!");
Else
Throw;
}
```

Continuations says when finished continue with something else. Two ways.

```
Task<int> primenumbertask = Task.Run (()=>
Enumerable.Range(2,3000000).Count (n =>
Enumerable.Range(2,(int)Math.Sqrt(n)-1).All (i => n % i > 0)));

Primenumbertask.continuewith (antecedent =>
{
Int result = antecedent.Result;
Console.writeline (result);
});
Var awaiter = primenumbertask.getawaiter();
Awaiter.oncompleted(() =>
{
```

```
Int result = awaiter.getresult();
Console.writeline (result);
});
```

If synchronization context present, run on UI thread task completion source. Any operation that start and finishes some time later. Slave task you manually drive. Mark finish. Ideal for i/o bound work. All benefits of tasks (returns, exceptions, continuations) without blocking thread. Create a task which you can wait and attach continuations. Controlled by these operations.

```
Public class taskcompletionsource< tresult>
{
Public void setresult (tresult result);
Public void setexception (Exception exception);
Public void setcanceled();
Public bool trysetresult (tresult result);
Public bool trysetexception (Exception exception);
Public bool trysetcanceled();
//...
}
```

Calling any function signals the task. Call exactly once. Following example prints 42 after 5s. Calling this is equivalent to creating long running task. Let's do the same without a thread. Then attaching continuation.

```
Var tcs = new taskcompletionsource<int>();
New Thread (()=>{
Thread.Sleep(5000); tcs.setresult (42);
}).Start();
Task<int> task = tcs.Task;
Console.writeline (task.Result)

Task<tresult> Run<tresult>(Func<tresult> function)
{
Var tcs = new taskcompletionsource<tresult>();
New Thread (()=>
{
Try{ tcs.setresult (function());}
Catch (Exception ex) { tcs.setexception (ex);}
}).Start();
Return tcs.Task;
}
//...
Task<int> task = Run (()=>{ Thread.Sleep (5000); return 42;});

Task<int> getanswertolife()
{
Var tcs = new taskcompletionsource<int>();
Var timer= new System.Timers.Timer(5000) { autoreset = false };
Timer.Elapsed+= delegate { timer.Dispose(); tcs.setresult(42); };
Timer.Start();
Return tcs.Task;
}
```

Thread used only when continuation running. Task.delay available.

```
Var awaiter = getanswertolife().getawaiter();
Awaiter.oncompleted(()=> Console.writeline (awaiter.getresult()));

Task Delay(int milliseconds)
```

```
{
Var tcs = new taskcompletionsource<object>();
Var timer= new System.Timers.Timer(milliseconds){ autoreset = false };
Timer.Elapsed+= delegate { timer.Dispose(); tcs.setresult(null);};
Timer.Start();
Return tcs.Task;
}
Delay(5000).getawaiter().oncompleted (()=> Console.writeline (42));
Delay(5000).continuewith (ant => Console.writeline (42));
Task.Delay(5000).getawaiter().oncompleted (()=> Console.writeline (42));
Task.Delay(5000).continuewith(ant => Console.writeline (42));
```

Chapter 34

Lecture 34

In last lecture we discussed Threadpool to avoid thread creation time. Task concept one thread many tasks. Continuations / Exceptions / Returns. Task Completion Source. Discussed Task.Delay.

Synchronous vs. Asynchronous. Async typically return quickly. Called non-blocking. Thread.start, task.run, at-taching continuations. For io bound we can usually work without thread. For cpu bound we can start and return task. Cpu bound async tasks.

```
Int getprimescount(int start, int count)
{
Return
Parallelenumerable.Range(start, count).Count(n =>
Enumerable.Range(2,(int)Math.Sqrt(n) - 1).All(i => n % i> 0));
}

Void displayprimecounts()
{
For(int i= 0; i < 10; i++)
Console.writeline(getprimescount(i * 1000000 + 2, 1000000) +
" primes between " + (i * 1000000) + " and "+ ((i+ 1) * 1000000 - 1));
Console.writeline("Done!");
}
Output:
78498 primes between 0 and999999
70435 primes between 1000000 and1999999
67883 primes between 2000000 and2999999
66330 primes between 3000000 and3999999
65367 primes between 4000000 and4999999
64336 primes between 5000000 and5999999
63799 primes between 6000000 and6999999
63129 primes between 7000000 and7999999
62712 primes between 8000000 and8999999
62090 primes between 9000000 and9999999
```

How to make async. Course grained.

```
Task.Run (()=> displayprimecounts());

Task<int> getprimescountasync(int start, int count)
{
Return Task.Run(() =>
Parallelenumerable.Range(start, count).Count(n =>
Enumerable.Range(2,(int)Math.Sqrt(n) - 1).All(i => n % i> 0)));
}
For(int i= 0; i < 10; i++)
{
Var awaiter = getprimescountasync (i*1000000 + 2, 1000000).getawaiter();
Awaiter.oncompleted(()=>
Console.writeline (awaiter.getresult() + " primes between... "));
}
Console.writeline ("Done");
```

Back to sequential. What if we want displayprimecounts async as well.

```
Void displayprimecounts()
{
Displayprimecountsfrom(0);
}
Void displayprimecountsfrom(int i)
{
Var awaiter = getprimescountasync(i * 1000000 + 2, 1000000).getawaiter();
Awaiter.oncompleted(()=>
{
Console.writeline(awaiter.getresult() + " primes between...");
If(i++ < 10) displayprimecountsfrom(i);
Else Console.writeline("Done");
});
}
```

Simple but lot of code.

```
Task  displayprimecountsasync()
{
Var machine = new primesstatemachine();
Machine.displayprimecountsfrom (0);
Return machine.Task;
}
Class  primesstatemachine
{
Taskcompletionsource <object> _tcs = new taskcompletionsource<object>();
Public Task Task{ get{ return _tcs.Task; } }
Public void displayprimecountsfrom (int i)
{
Var awaiter = getprimescountasync (i*1000000+2,1000000).getawaiter();
Awaiter.oncompleted(()=>
{
Console.writeline (awaiter.getresult());
If(i++ < 10) displayprimecountsfrom (i);
Else { Console.writeline("Done"); _tcs.setresult (null); }
});
}
}
```

Its pretty simple in C# 5.0.

```
Async Task displayprimecounts()
{
For(int i= 0; i < 10; i++)
Console.writeline(await  getprimescountasync(i * 1000000 + 2, 1000000) +
" primes between " + (i * 1000000) + " and "+ ((i+ 1) * 1000000 - 1));
Console.writeline("Done!");
}
```

C# 5.0 introduces async and await keywords. Eliminates plumbing for async code. As simple as sync code. “await” simplifies attaching continuations. E.g. Also code to handle if sync completion. And some other details.

```
Var result= await expression;
Statement(s);
Var awaiter = expression.getawaiter();
```

```
Awaiter.oncompleted(()=>
{
Var result= awaiter.getresult();
Statement(s);
);
```

Returning to example. We can call it with await and use “async” keyword so compiler treats await specially. Async can be applied to methods returning void or Task or Task¡tresult¿.

```
Task<int> getprimescountasync(int start, int count)
{
Return Task.Run(() =>
Parallelenumerable.Range(start, count).Count(n =>
Enumerable.Range(2,(int)Math.Sqrt(n) - 1).All(i =>  n %  i> 0)));
}
Int result = await getprimescountasync (2,1000000);
Console.writeline (result);
Async void displayprimescount()
{
Int result = await getprimescountasync(2, 1000000);
Console.writeline(result);
}
```

Async doesnt change method so no need to list in interface. Can add async when implementing. Methods with async are called asynchronous. When await. Method returns to caller. But before returning a continuation attached to the task. If exception rethrown, otherwise return value assigned to await statement. Let’s see the expansion again. Await expression returns int because Task¡int¿ also possible to wait on void Task.

```
Void displayprimescount()
{
Var awaiter  = getprimescountasync(2,
Awaiter.oncompleted(()  =>
{
Int result = awaiter.getresult();
Console.writeline(result);
});
}
Await Task.Delay (5000);
(1000000).getawaiter();
Console.writeline ("Five seconds passed!");
```

Capturing local state in await. Real power of await. When execution resumes in continuation, local vars have the same values compiler translates into state machines. A general method. If we do ourself we may be doing the complex code we showed before. If sync context, same thread e.g. UI. Otherwise whatever free thread Let’s write a UI program that remains responsive with a cpu bound task. Let’s start with the sync version.

```
Async void displayprimecounts()
{
For(int i = 0; i < 10; i++)
Console.writeline(await  getprimescountasync(i
}
```

Synchronous version. Unresponsive. 2step in making async.

```
Class testui  : Window
{
Button  _button  = new Button  { Content  = "Go"  };
Textblock  _results  = new textblock();
Public testui()
{
    Var panel = new stackpanel();
    Panel.Children.Add(_button);
    Panel.Children.Add(_results);
    Content = panel;
    _button.Click += (sender, args) => Go();
}
Void Go()
{
For(int  i= 1;  i < 5;  i++)
    _results.Text+= getprimescount(i*1000000,1000000) +" primes between "+(i*1000000)+ "
and "+((i+1)*1000000-1) +Environment.newline;
}
Int getprimescount(int start, int count)
{
    Return parallelenumerable.Range(start, count).Count(n => Enumerable.Range(2,
 (int)Math.Sqrt(n) - 1).All(i => n % i> 0));
}
}
```

Asynchronous version. Simplicity. Call async functions. And await them. Go leases time on UI thread. Task.Run is
The real parallel.

```
Task<int> getprimescountasync(int start, int count)
{
Return Task.Run (()=>
Parallelenumerable.Range (start, count).Count (n =>
Enumerable.Range(2,(int)Math.Sqrt(n) - 1).All(i =>  n %  i> 0)));
}

Async void Go()
{
_button.isenabled = false;
For(int  i= 1;  i < 5;  i++)
    _results.Text+= await  getprimescountasync(i*1000000,1000000)+" primes between "+(i
 *1000000)+ " and "+((i+1)*1000000-1)+Environment.newline;
_button.isenabled = true;
}
```

Simple but have to handle re-entrancy. And real parallel code does not use shared state. Another example. Download web pages and sum their lengths. System.Net.webclient.downloaddatataskasync method. Returns Task¡byte[]¿ so await returns byte[].

```
Async void Go()
{
_button.isenabled = false;
String[] urls= "www.albahari.com www.oreilly.com www.linqpad.net".Split();
Int totallength = 0;
Try
{
Foreach (string url in urls)
```

```
{
Var uri = new Uri("http://" + url);
Byte[] data = await new webclient().downloaddatataskasync(uri);
_results.Text+= "Length of "+ url + " is "+ data.Length +Environment.newline;
Totallength += data.Length;
}
_results.Text+= "Total length: "+ totallength;
}
Catch (webexception ex)
{
_results.Text+= "Error: "+ ex.Message;
}
Finally { _button.isenabled = true; }
}
```

Mirrors how we would write it sync. Even though control returns to caller after first await, the finally blocks is not run until method is logically completed. Let's see underneath on the message loop.

```
While (!Thisapplication.Ended)
{
Wait for something to appear in message queue
Got something: what kind of message is it?
Keyboard/mouse message -> fire an event handler
User begininvoke/Invoke message -> execute  delegate
}
```

Event handler run via this loop. Go runs as far as await and then returns to this loop. Compilers expansion of await ensures a continuation is setup. Because we awaited on a UI thread, the continuation is posted on the sync context, which ensures it runs via message loop. Go keeps running pseudo-concurrently with UI thread. True concurrency occurs while downloaddatataskasync is running.vs. Course grained where the loop itself on worker thread. Gets difficult when progress reporting.

```
_button.Click+= (sender, args) =>
{
_button.isenabled = false;
Task.Run (()=> Go());
};

Void Go()
{
For(int i= 1; i < 5; i++)
{
Int result = getprimescount(i * 1000000, 1000000);
Dispatcher.begininvoke(new Action(() =>
_results.Text+= result+ " primes between " + (i * 1000000) +
" and " + ((i+ 1) * 1000000 - 1) + Environment.newline));
}
Dispatcher.begininvoke(new Action(() => _button.isenabled = true));
}
```

Chapter 35

Lecture 35

Let's revise the downloading code from last lecture.

```
Async void Go()
{
_button.isenabled = false;
String[] urls= "www.albahari.com www.oreilly.com www.linqpad.net".Split();
Int totallength = 0;
Try
{
Foreach (string url in urls)
{
Var uri = new Uri("http://" + url);
Byte[] data = await new webclient().downloaddatataskasync(uri);
_results.Text+= "Length of "+ url + " is " + data.Length+
Environment.newline;
Totallength += data.Length;
}
_results.Text+= "Total length: "+ totallength;
}
Catch (webexception ex)
{
_results.Text+= "Error: "+ ex.Message;
}
Finally { _button.isenabled = true; }
}
```

We can return a Task from void function without explicitly return it. Enables async call chains. Compiler indirectly uses taskcompletionsource to implement methods returning Tasks. We can expand printanswertolife like this.
Nuances aside.

```
Async Task printanswertolife()
{
Await Task.Delay(5000);
Int answer = 21 * 2;
Console.writeline(answer);
}
Async Task Go()
{
Await  printanswertolife();
Console.writeline("Done");
}
```

Whenever task finishes, execution returns to whoever awaited for the task. You can return Task¡tresult¿ if the method returns tresult. Internally that results in TCS signaled with a value.

```
Task printanswertolife()
{
Var tcs = new taskcompletionsource<object>();
Var awaiter = Task.Delay(5000).getawaiter();
Awaiter.oncompleted(()=>
```

```
{
Try
{
Awaiter.getresult();
Int answer = 21 * 2;
Console.writeline(answer);
Tcs.setresult(null);
}
Catch (Exception ex) { tcs.setexception(ex); }
});
Return tcs.Task;
}


Async Task <int> getanswertolife()
{
Await Task.Delay(5000);
Int answer = 21 * 2;
Return answer;
}
```

Let's see the async version vs. The sync version.

```
Async Task Go()
{
Await  printanswertolife();
Console.writeline("Done");
}
Async Task printanswertolife()
{
Int answer =
Await getanswertolife();
Console.writeline(answer);
}
Async Task <int> getanswertolife()
{
Await Task.Delay(5000);
Int answer = 21 * 2;
Return answer;
}
Void Go()
{
Printanswertolife();
Console.writeline("Done");
}
Void printanswertolife()
{
Int answer = getanswertolife();
Console.writeline(answer);
}
Int getanswertolife()
{
Thread.Sleep(5000);
Int answer = 21 * 2;
Return answer;
}
```

Same ease of programming as sync. Intentional. So three steps. Write sync. Use async and await. Return Task and Task¡tresult¿ so awaitable. This means only Task.Run for real parallel cpu task and TCS for real parallel IO task. Rest of TCS are taken care of by compiler.

Async call graph exec. Brief sync exec on thread calling Go. Everyone await and returns. When Delay fires a thread. Remaining statements run. Eventually Gos task is marked completed.

```
Async Task Go()
{
Var task = printanswertolife();
Await task;
Console.writeline("Done");
}
Async Task printanswertolife()
{
Var task = getanswertolife();
Int answer = await task;
Console.writeline(answer);
}
Async Task <int> getanswertolife()
{
Var task = Task.Delay(5000);
Await task;
Int answer = 21 * 2;
Return answer;
}
```

Let's discuss Parallelism. Go is not awaited, allowed and required here. So Let's see how to run two tasks in parallel and then await them. True concurrency at bottom level operations. If a sync context, only pseudo-concurrency. So only switched on await. That means we can increment shared var without locking. But cant assume same value before and after await.

```
Var task1= printanswertolife();
Var task2= printanswertolife();
Await task1;
Await task2;
Async Task <int> getanswertolife()
{
    _x++;
    Await Task.Delay(5000);
    Return 21 * 2;
}
```

Async lambda expr.

```
Func<Task> unnamed = async ()=>
{
Await Task.Delay(1000);
Console.writeline("Foo");
};

Await namedmethod();
Await unnamed();
Mybutton.Click += async(sender, args)=>
```

```
{
Await Task.Delay(1000);
Mybutton.Content= "Done";
};
Mybutton.Click += buttonhandler;
```

```
Async void buttonhandler (object sender, eventargs args)
{
Await Task.Delay(1000);
Mybutton.Content= "Done";
};

Func<Task<int>> unnamed = async () =>
{
Await Task.Delay(1000);
Return123;
};
Int answer = await unnamed();
```

Cancellation. Cancel is on cancellationtokensource. Most have builtin cancellation support.

```
Class  cancellationtoken
{
Public bool  iscancellationrequested { get; private set;}
Public void Cancel() { iscancellationrequested = true; }
Public void  throwifcancellationrequested()
{
If (iscancellationrequested)
Throw new operationcanceledexception();
}
}
Async Task Foo(cancellationtoken  cancellationtoken)
{
For(int i= 0; i < 10; i++)
{
Console.writeline(i);
Await Task.Delay(1000);
Cancellationtoken.throwifcancellationrequested();
}
}
Async Task Foo(cancellationtoken  cancellationtoken)
{
For(int i= 0; i < 10; i++)
{
Console.writeline(i);
Await Task.Delay(1000, cancellationtoken);
}
}
Var cancelsource= new cancellationtokensource();
Task.Delay (5000).continuewith(ant => cancelsource.Cancel());
//...

Var cancelsource= new cancellationtokensource(5000);
Try{ await Foo (cancelsource.Token); }
Catch (operationcanceledexception ex)
{  Console.writeline("Cancelled"); }
```

Can even use cancellation with Task.Wait. I.e. Sync methods but have to call cancel from another task. Infact you can give a time interval after which auto cancelled. Useful for timeouts.

Progress reporting. Thread safety issues. Iprogress and progress. Progress ctor.

```
Async Task Foo(Action<int> onprogresspercentchanged)
{
Return Task.Run(() =>
{
For(int i= 0; i < 1000; i++)
{
If(i % 10 == 0)  onprogresspercentchanged(i / 10);
// Do something compute-bound...
}
});
}
Action<int> progress = i=> Console.writeline (i+ "%");
Await Foo(progress);

Var progress= new Progress<int>(i => Console.writeline (i+ " %"));
Await Foo(progress);

Public interface iprogress<in T>
{
Void Report(T value);
}

Task Foo (iprogress<int>onprogresspercentchanged)
{
Return Task.Run (()=>
{
For(int i= 0; i < 1000; i++)
{
If (i % 10 == 0)
Onprogresspercentchanged.Report(i / 10);
// Do something compute-bound...
}
});
}
```

Task-based Async Pattern (TAP): A TAP method Returns a “hot” (running) Taskor Task¡tresult¿. Has an “Async” suffix (except for special cases such as task combinators). Is overloaded to accept a cancellation token and/or iprogress¡T¿if it supports cancellation and/or progress reporting. Returns quickly to the caller (has only a small initial synchronous phase). Does not tie up a thread if I/O-bound.

Task Combinators.

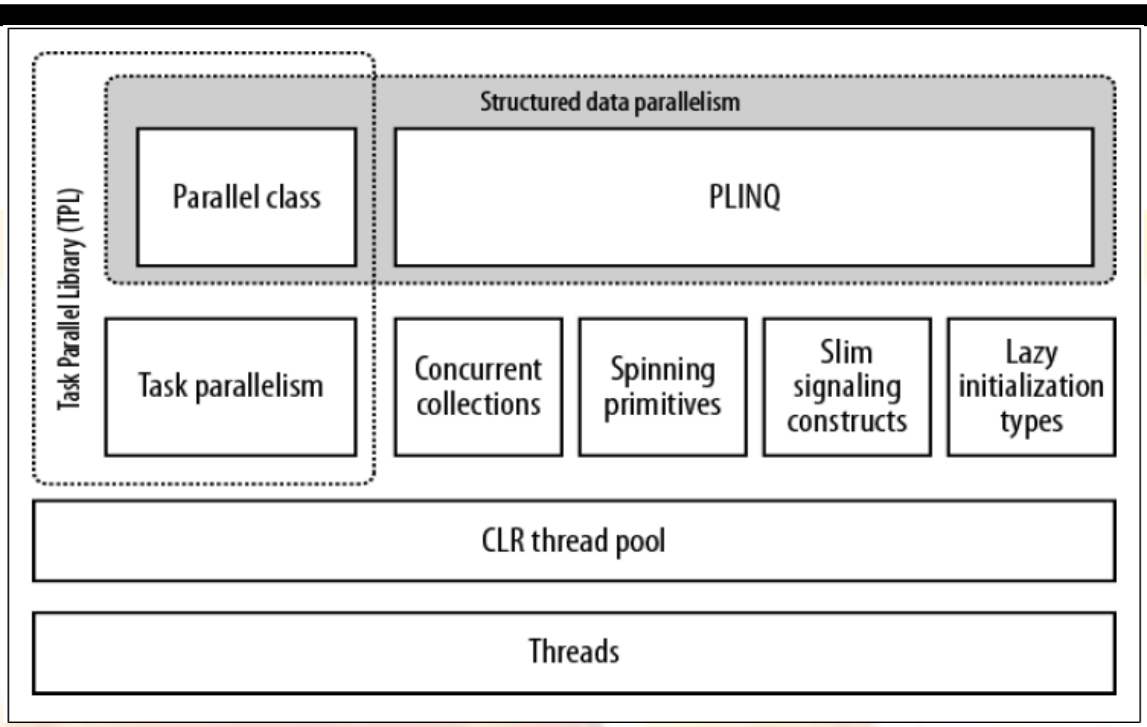
```
Task<int> winningtask = await Task.whenany
Console.writeline ("Done");
Console.writeline (winningtask.Result); 4
(Delay1(), Delay2(), Delay3());
Int answer = await await Task.whenany(Delay1(), Delay2(), Delay3());
Task <string> task= someasyncfunc();
Task winner = await (Task.whenany(someoperation, Task.Delay(5000)));
If(winner != task) throw new timeoutexception();
String result= await task;
```

Chapter 36

Lecture 36

We can imlement timeout with whenany task combinator. On the other hand whenall waits for all. Difference is that should task1 fault, we never get to.

```
Int answer = await await Task.whenany(Delay1(), Delay2(), Delay3());
Task<string> task = someasyncfunc();
Task winner = await (Task.whenany(someoperation, Task.Delay(5000)));
If(winner != task) throw new timeoutexception();
String result= await task;
Await Task.whenall (Delay1(), Delay2(), Delay3());

Task task1 = Delay1(), task2 = Delay2(), task3 = Delay3();
Await task1; await task2; await task3;

Task task1 = Task.Run (()=>{ throw null;} );
Task task2 = Task.Run (()=>{ throw null;} );
Task all = Task.whenall (task1, task2);
Try{ await all;}
Catch
{
Console.writeline (all.Exception.innerexceptions.Count);
}
Task <int> task1 = Task.Run(()=>1);
Task <int> task2 = Task.Run(()=>2);
Int[] results= await Task.whenall(task1, task2);
Async Task<int> gettotalsize(string[] uris)
{
Ienumerable<Task<byte[]>> downloadtasks= uris.Select(uri =>
New(webclient).downloaddatataskasync(uri));
Byte[][] contents = await Task.whenall(downloadtasks);
Return contents.Sum(c => c.Length);
}
Async Task<int> gettotalsize(string[] uris)
{
Ienumerable<Task<int>> downloadtasks = uris.Select(async uri =>
(await new webclient().downloaddatataskasync (uri)).Length);
Int[] contentlengths = await Task.whenall (downloadtasks);
Return  contentlengths.Sum();
}
```



Task Parallel Library exploit multicore for real parallel tasks. 3 steps partition into small chunks, process, collate the results in thread-safe manner. Lock contention and lot of cumbersome code. Data parallelism vs. Task parallelism. Data parallelism easier and scales well. It also structured. Same place in code where parallelism starts and ends.

Concurrent collections also useful when you want a thread-safe collection three static methods in the Parallel class Parallel.Invoke is not a shortcut for creating many threads. Actually batches to use processors efficiently. Combining on the user. Following is thread-unsafe. Locking may make slow. Concurrentbag.

```
Public static void Invoke(params Action[] actions);
Parallel.Invoke (
() => new webclient().downloadfile ("http://www.linqpad.net", "lp.html"),
() => new webclient().downloadfile ("http://www.jaoo.dk", "jaoo.html"));
Var data = new List<string>();
Parallel.Invoke (
() => data.Add(new webclient().downloadstring("http://www.foo.com")),
() => data.Add(new webclient().downloadstring("http://www.far.com")));
Public static  parallelloopresult For(
Int frominclusive, int toexclusive, Action<int> body)
Public static parallelloopresult foreach<tsource> (
Ienumerable<tsource> source, Action<tsource> body)
For(int i= 0; i < 100; i++)
Foo(i);
Parallel.For(0, 100, i=> Foo(i));
Parallel.For(0, 100, Foo);
Foreach (char c in "Hello, world")
Foo(c);
Parallel.foreach("Hello, world", Foo);
Var keypairs = new string[6];
Parallel.For(0, keypairs.Length,
I => keypairs[i] = RSA.Create().toxmlstring(true));
```

Next we discuss using loop counter, breaking out of loops, and using per-thread counters. Loop counters are easy with sequential. Use overloaded version. What is parallel loop state.

```
Int i = 0;
Foreach (char c in "Hello, world")
Console.writeline (c.tostring()+ i++);

Public static parallelloopresult foreach<tsource> (
Ienumerable<tsource> source,
Action<tsource,parallelloopstate,long> body)

Parallel.foreach("Hello, world",(c, state, i)=>
{ Console.writeline(c.tostring()+ i); });

Public class  parallelloopstate
{
Public void Break();
Public void Stop();
Public bool isexceptional{ get;}
Public bool isstopped{ get;}
Public long? Lowestbreakiteration { get;}
Public bool shouldexitcurrentiteration{ get;}
20          }
```

A normal break in parallel. Diff between Break and Stop. Break reaches atleast the sequential point. What about local value. Use overload.

```
Foreach (char c in "Hello, world")
If (c == ',')
Break;
Else
Console.Write(c);

// OUTPUT: Hello

Parallel.foreach("Hello, world",(c, loopstate)=>
{
If (c == ',')
Loopstate.Break();
Else
Console.Write(c);
});

// OUTPUT: Hlloe

Public static  parallelloopresult For<tlocal>(
Int frominclusive,
Int toexclusive,
Func<tlocal> localinit,
Func<int, parallelloopstate, tlocal, tlocal> body,
Action<tlocal> localfinally);

Object locker= new object();
Double total = 0;
Parallel.For(1,10000000,i =>{ lock(locker) total += Math.Sqrt(i); });
```

```
Object locker= new object();
Double grandtotal = 0;
Parallel.For(1,10000000,() => 0.0,(i, state, localtotal)=>
Localtotal + Math.Sqrt(i),
Localtotal =>
{ lock(locker) grandtotal+= localtotal; }
);
```

Concurrent collections include concurrent stack, concurrent queue, concurrent bag, concurrent dictionary. They are optimized for concurrent. But also useful for thread-safe. Conventional outperform in all but highly concurrent situations. Thread-safe collection doesnt guarantee thread safe code. You enumerate, another modifies, you get com-bination of old and new. No concurrent version of List. Stack queue bag implemented with linked list but less mem efficient as a consequence. E.g. The following code is 3 times slower but not with reads. Also some atomic test and act.like trypop.

Iproducerconsumercollection¡T¿ has no need to lock. Adds tryadd and trytake methods. With stack, most recent, queue oldest, and bag whatever. Concurrentbag is an unordered collection. Its kind of linked list for each thread. Add has almost no contention.

Blockingcollection¡T¿ wait instead of returning false. Wrapper class. Also Let's you limit the size. Called bounded blocking collection. Ctor takes a iproducerconsumer and limit. Default is to make a queue. Can also give multiple collecitons and use addtoany takefromany. Can also takegetconsumingenumerable. Let's make a producer consumer queue.

```
Public class pcqueue : idisposable
{
    Blockingcollection<Action> _taskq = new blockingcollection<Action>();
    Public pcqueue(int workercount)
    {
        For (int i = 0; i < workercount; i++)
            Task.Factory.startnew(Consume);
    }
    Public void Enqueue(Action action) { _taskq.Add(action); }
    Void Consume()
    {
        Foreach (Action action in _taskq.getconsumingenumerable())
            Action();
    }
    Public void Dispose() { _taskq.completeadding(); }
}

Public class pcqueue : idisposable
{
    Blockingcollection<Task> _taskq = new blockingcollection<Task>();
    Public pcqueue (int workercount)
    {
        For (int i = 0; i < workercount; i++)
            Task.Factory.startnew (Consume);
    }
    Public Task Enqueue (Action action, cancellationtoken canceltoken
        = default (cancellationtoken))
    {
        Var task = new Task(action, canceltoken);
        _taskq.Add(task);
        Return task;
    }
```

```
    Public Task<tresult> Enqueue<tresult>(Func<tresult> func,
        Cancellationtoken canceltoken = default (cancellationtoken))
    {
        Var task = new Task<tresult>(func, canceltoken);
        _taskq.Add(task);
        Return task;
    }
    Void Consume()
    {
        Foreach (var task in _taskq.getconsumingenumerable())
            Try
            {
                If (!Task.iscanceled) task.runsynchronously();
            }
            Catch (invalidoperationexception) { }
    }
    Public void Dispose() { _taskq.completeadding(); }
}
```

Chapter 37

Lecture 37

In the last lecture, finished discussion on tasks and multithreading. Important part of event driven and visual pro-gramming. To remain responsive. The task parallel library. Parallel.Invoke, For, Foreach. Loop counter, break, stop, per-thread counter. Concurrent collections, stack, queue, bag, dictionary. Ended with a producer consumer queue with tasks.

Now we'll discuss event driven programming on the web and mobile. Client side programming and event-handling. Server side web programming in other courses.

Static web sites were there initially. Html. Hypertext markup language. Display information and thats it. Today websites reach interactivity of desktop applications. Because of javascript. Animation, interactivity, dynamic visual effects. E.g. Immediately give error msg when wrong data, give total when add to shopping cart, slideshow instead of image list, expand collapse information, popup tooltips. Immediate feedback. No delay like server side. No constant loading reloading. So feels like desktop programs.

E.g. You visited google maps. JS (javascript) in action. Zoom in zoom out. Prev. Map sites used reloading.

Javascript was introduced in 95 by netscape. About as old as web. But mostly hobby things like flies with mouse or messages that move like stock ticker. Many scripts but didnt work in all browsers, even crashed them often. JS has nothing to do with Java, originally named livescript but renamed to associate with the then hot Java. Initial interoperability problems in netscape and IE. Often added incompatible features. MS introduced jscript. Their version of JS for IE. These days mostly handled. Standardization. Some quirks left. Called ecmascript. The official standardization name. Refueled by high profile sites like google using JS in last decade. Now JS even used by some non-web scripting. Even flash actionscript based on it. Or write widgets, phone apps etc.

JS is a prog language. Can be hard for some. Also browser incompatibilities make testing difficult. Jquery is a JS library intended to make JS programming easier. Jquery solves JS complexity and browser incompatibilities. Can do things in single LOC that would take hundreds. Many advanced features come as jquery plugins. Used on millions of websites.

HTML: structural layer and CSS: presentation layer and JS: behavioral layer.

HTML has simple commands called tags. Doctype is of html5 here. Tells browser how to render the page. What standards. Five types of html in use: HTML 4.01 Transitional, HTML 4.01 Strict, XHTML 1.0 Transitional, XHTML 1.0 Strict, and HTML5. All current browsers understand them all. Starting and closing tags like XML. At least three tags. Html root tag, head tag containing title etc, and body tag containing all parts to be rendered in browser window. ¡p¿¡/p¿ is paragraph ¡strong¿¡/strong¿ is emphasis¡a href=”http:...”¿¡/a¿ is a hyperlink. XML attribute and value. Validating html means checking if all tags appropriately closed etc.

```
<!DOCTYPE  html >
<html>
<head>
<meta charset=utf-8>
<title>Hey, I am the title of this web page.</title>
</head >
<body>
```

```
Hey, I am some body text on this web page.
</body >
</html >
```

Originally there was only HTML. CSS is a formatting language. Html only to structure, so ¡h1¿ ¡h2¿ are both headings with diff imp ¡ul¿ is an unordered list. CSS adds design. CSS style is a rule telling web browser how to display an element. E.g. You can make CSS rule to make all ¡h1¿ tags 36 px tall, in courier font, and in orange. CSS can do more powerful stuff, like add border, change margins, and even control exact placement on web page. JS can add/remove/change CSS properties based on input or mouse clicks. Even animate from one property to another. Like yellow to red. Or animate across screen by changing position.

A single CSS style is a rule that tells how to format. Make “this” look like “that”. Selector and declaration block. E.g. Selector can be headline, paragraph of text, photo etc. Declaration block can turn text blue, add red border around a paragraph, position the photo at center of page etc. E.g. P color: red; font-size: 1.5em; selector, declaration block has declarations. Each is a property value pair and then “;”.

JS Let’s a page re-act. Smart web forms. Let’s user know when they miss important info, make elements appear or disappear, or move around a webpage. Even load new content from web server without reloading. More engaging and effective web sites.

Client side vs. Server side. Prog lang for the web browser. Alternate is a server prog lang. Php, .net, asp, cold fusion, ruby on rails, etc. They run on web server. Log of intelligence by accessing DB, process CC, send emails. Visitors wait until response comes. Client side lang can re-act immediately. Responsive. Other client side technologies are applet’s, silverlight, flash. Often requires a plugin or start slow because of downloading. Sometimes even diff to see if flash or JS. Once yahoo maps was flash. Then re-written. Right click and see if About the Flash Player Ajax brings client-side server-side together. JS talks to server, downloads content, and update webpage. Google maps Let’s you move to new areas. JS is a prog lang and can be used on server side. E.g. Node.js supports JS on server-side.

Compiled vs scripted languages (interpreted). JS interpreter is in web browser. Let’s write a prog to ask visitor name, get response, and print a welcome msg. Web browser has layout or rendering engine (understanding HTML and CSS) and a JS interpreter. Tell web browser about JS in a ¡script¿¡/script¿ tag. Here is a script in html 4.01 and in html5.

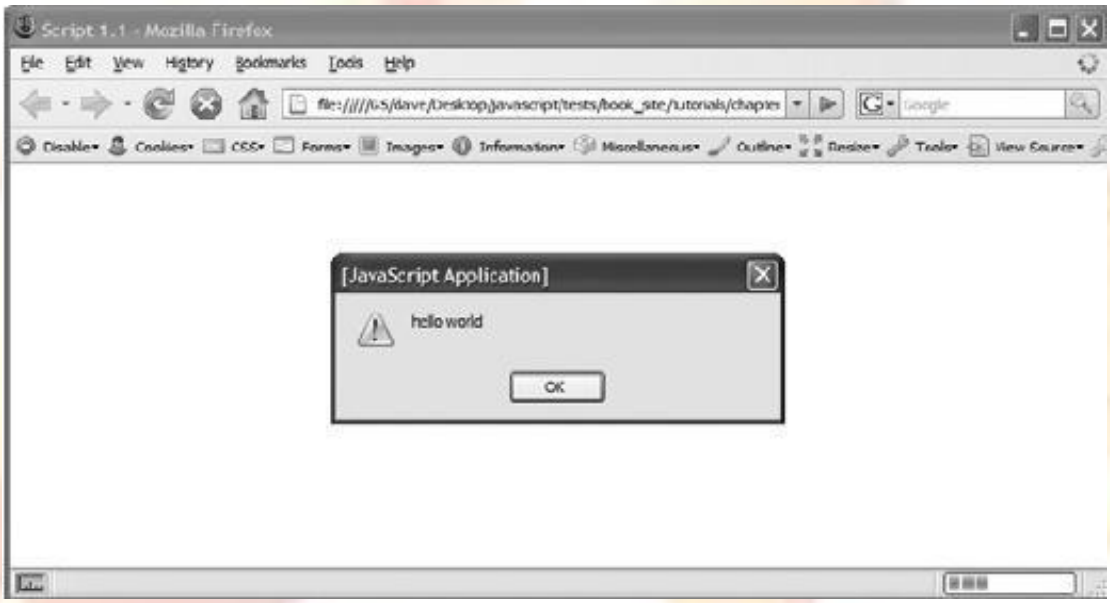
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>My Web Page</title>
<script type="text/javascript">
</script>
</head>
<!Doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>My Web Page</title>
<script>
</script>
</head>
4.01//EN"  "http://www.w3.org/TR/html4/strict.dtd">
```

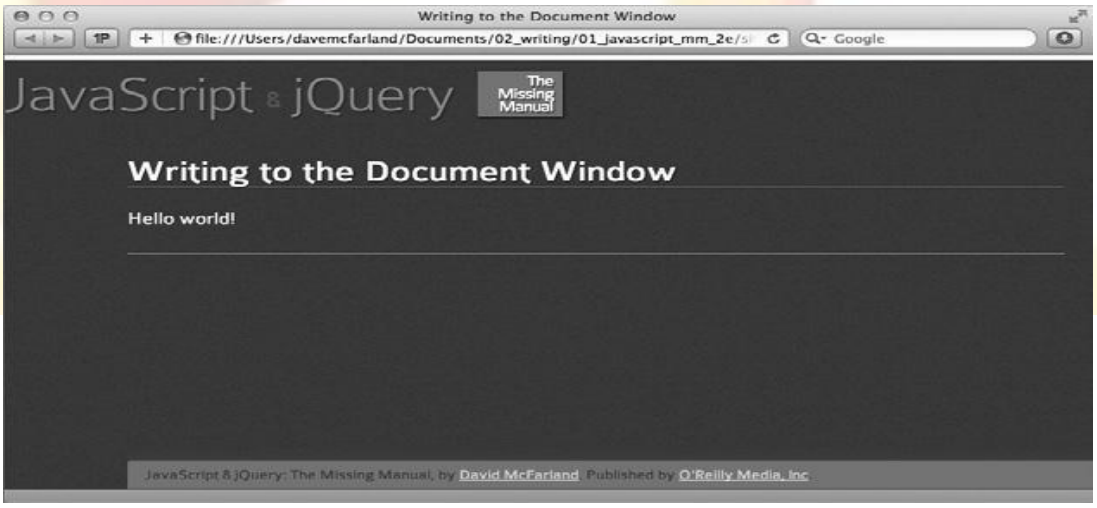
Usually in head section and at one place. But its ok to put it anywhere and in multiple tags. Script can also be placed after ¡/body¿ so script loaded after page displayed can also use external script files. Easy to share.

Separate tags if you want inline code AND “src” attrib for external file. Can and often use multiple external files.

```
<!Doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>My Web Page</title>
<script src="navigation.js"></script>
</head>
<!Doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>My Web Page</title>
<script src="navigation.js"></script>
<script src="slideshow.js"></script>
<script>Alert('hello world!');</script></head>
```

Let’s open hello.html in web browser.





```
<book = javascript \& jquery the Missing Manual page 30>
<script>
Document.write('<p>Hello world!</p>');
</script >
```

Before the web page because of placement. Web page appears after OK pressed. Use document.write and <script> tag anywhere.

```
<link href="../_css/site.css" rel="stylesheet">
<script src="../_js/jquery-1.6.3.min.js"></script>
<script>
$(function (){
$('body').hide().fadein(3000);
});
</script >
```

Lot of basic syntax like C++, C#. Create a variable using "var x", names begin with letter, $, or . Var days = ['Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun']; alert(days[0]); var playlist = []; var prefs = [1, 223, 'www.oreilly.com', false]; prefs.push('test'); prefs.push('test',test2); arrays grow prefs.unshift('test'); prefs.unshift('test',test2); insert at start shift() gets/removes the first element. Queue using push/shift pop() removes last. Stack alert and prompt functions.

```
Var TAX = .08;
Function calculatetotal(quantity, price){
Var total= quantity * price * (1 + TAX);
Var formattedtotal = total.tofixed(2);
Return  formattedtotal;
}

Var saletotal= calculatetotal(2,16.95);
Document.write('Total cost is: \$'+ saletotal);
```

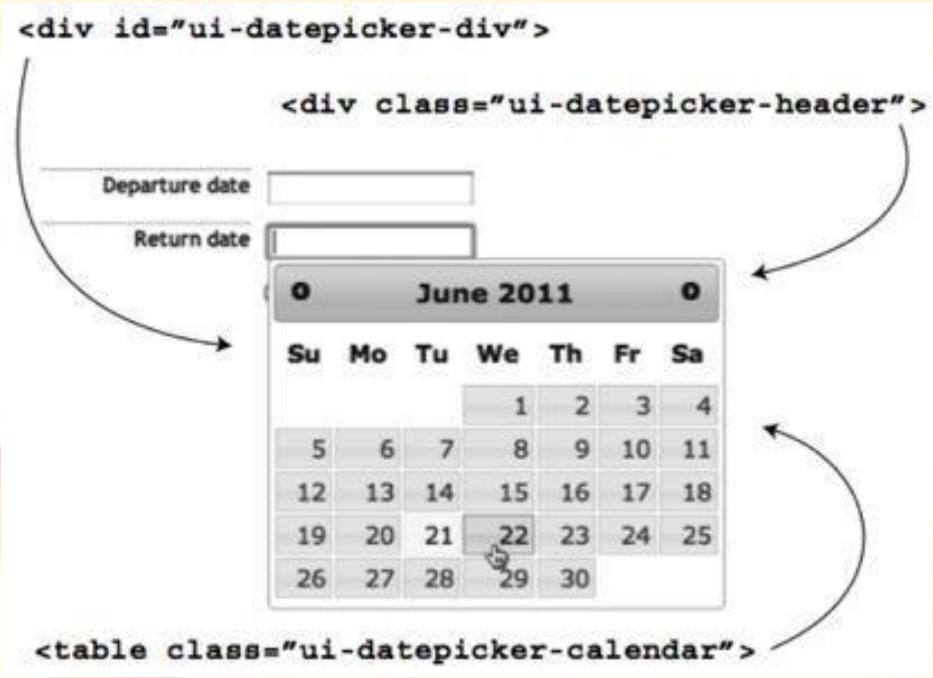
If, while, for, dowhile like C#. Function declarations. No types. Return value.

Let's discuss jquery now. Many JS programs select elements, add new content, hide and show content, modify tag's attributes, determine value of form fields, and react to user actions. The details complicated specially with browser interoperability. Libraries offer a set of functions to make these tasks easy. Only 30k compressed library size. Easy to learn. Used on millions of sites. Free. Dev community. Plugins !!! Where to get jquery.js. Use cdns dont need to host your own. Often cached. Google one is v popular.

```
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.6.3.min.js">
</script >

<script src="http://code.jquery.com/jquery-1.6.3.min.js"></script>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.3/jquery.min.js">
</script >

<script src="js/jquery-1.6.3.min.js"></script>
<script>
\$(document).ready(function (){
// your programming goes here
});
</script >

\$(function (){
```

```
// your programming goes here
});// end ready
```

Second script tag for jquery prog. $(document).ready() waits until HTML of webpage loads and then runs the function. Browser processes in order and to work with elements you want them downloaded. Shortcut for ready function.

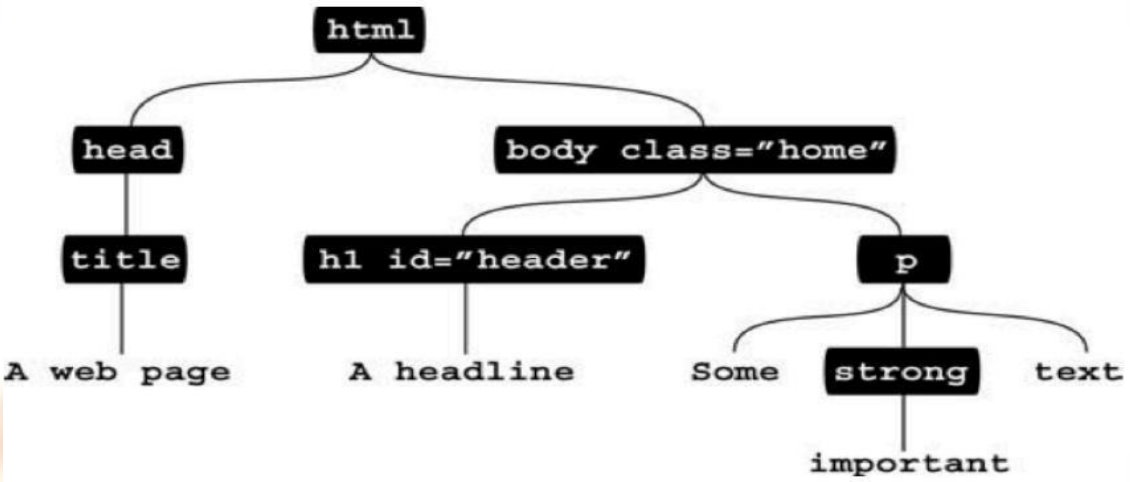


Dynamically changing webpage is key idea. Mouse over, click, detail info. E.g. Date picker is run by JS but itself is made of HTML / CSS. JS just makes presentation interactive. Two steps. Select sth, do sth with it. Do sth can be change prop, add/remove element, extra info, add/remove class attrib, or a combination of these.

Chapter 38

Lecture 38

HTML DOM is much like an XML DOM. JS provides ways to select e.g. Some browsers allow selecting by CSS sometimes not cross browser.




```
Document.getelementbyid('banner');
Document.getelementbytagname('a');
```

To select ¡a¿ tags with class navbutton, you have to select all ¡a¿ tags, then iterate and find the ones with the right class. In jquery $('selector') e.g. $('#banner') is the tag with id banner. $('#banner').html('¡h1¿javascript was here¡/h1¿');. Html is a jquery helper function.
Basic selectors are ID selectors, element selectors, class selectors.

```
<p id="message">Special message</p>
Var messagepara = document.getelementbyid('message');
Var messagepara = $('\#message');

Var linkslist= document.getelementsbytagname('a');
Var linkslist= $('a');

$('.submenu')
$('.submenu').hide();
```

Advanced selectors are

- Descendent selectors $('#navbar a') ,
- Child selectors $('body ¿ p') ,
- Adjacent sibling $('h2 + div') ,
- Attribute selectors $('img[alt]'), $('input[type="text"]'), $('a[href ="mailto:"]'), $('a[href$=".pdf"]'), $('a[href*="missingmanuals.com"]') , form element selectors later.
  Jquery filters are :even :odd $('.striped tr:even') :first :last :not $('a:not(.navbutton)'); :has $('li:has(a)') — diff from descendent :contains $('a:contains(Click Me!)') :hidden :visible $('div:hidden').show();

Jquery selection. Dont end up with DOM lists. Rather jquery equivalents automatic loops.$('#slideshowimg').hide();

Chaining functions...$('#popup').width(300).height(300);
$('#popup').width(300).height(300).text('Hi!').fadein(100

Let's see jquery functions to add content

Take this example

```
<div id="container">
<div id="errors">
<h2>Errors:</h2>
</div >
</div >
Alert(\$('\#errors').html());
$('#product101').replacewith(<p>Added to cart</p>');
$('\#errors').html('<p>There are four errors in this form</p>');
$('\#errors h2').text('No errors found');
$('\#errors').append('<p>There are four errors in this form</p>');
$('\#errors').prepend('<p>There are four errors in this form</p>');
$('\#username').after('<span class="error">User name required</span>');
$('\#popup').remove();
$('\#product101').replacewith(<p>Added to cart</p>');
$('a[href\^="http://"]').addclass('externallink');
<a href="http://www.oreilly.com/">
<a href="http://www.oreilly.com/" class="externallink">
```

Attributes can be manipulated with addclass removeclass toggleclass. And css. Var bgcolor = $('#main').css('background-color'); $('body').css('font-size', '200$('p.highlight').css('border', '1px solid black'); multiple css props can be changed together.

```
Var basefont = $('body').css('font-size');
Basefont = parseint(basefont,10);
$('body').css('font-size',basefont * 2);
```

```
$('#highlighteddiv').css('background-color','#FF0000','border','2px solid #FE0037');
```

For changing html attribute, css and addclass are just shortcuts general purpose attr() and removeattr() var image-File = $('#banner img').attr('src'); $('#banner img').attr('src','images/newimage.png'); $('body').removeattr('bgcolor');. Acting on each element in a selection. When you do want something special. Each() and anonymous function. Use "this" for current element as DOM obj. $(this) for current element as jquery selection.

```
Var imagefile= \$('\#banner img').attr('src');
\$('\#banner img').attr('src', 'images/newimage.png');
\$('body').removeattr('bgcolor');
\$('selector').each(function (){
// code goes in here
});
\$('a[href\^=http://]').each(function (){
Var extlink = \$(this).attr('href');
\$('\#biblist').append('<li>'+ extlink+ '</li >');
});
```

Events. Things happen to webpage. Page loading, mouse move, key press you respond to events. Mouse events: click, dblclick, mousedown, mouseup, mouseover, mouseout, mousemove. Doc/window events: load, resize, scroll, unload. Form events: submit, reset, change, focus, blur. Keyboard: keypress (over n over), keydown, keyup.

Step 1: select elements, step 2: assign an event, step 3: pass function to event.

```
$('\#menu').mouseover(function (){
$('\#submenu').show();
});// end mouseover
```

Ready() vs. Load event

```
\$(function(){
// do something on document ready
});
```

Jquery events. Hover. Toggle is like hover except worked on and off by clicks. Event object is passed to all functions handling events.

```
<script>
\$(document).ready(function (){
\$('html').dblclick(function (){
Alert('ouch');
});// end double click
\$('a').mouseover(function(){
Var message = "<p>You moused over a link</p>";
\$('.main').append(message);
});// end mouseover
\$('\#button').click(function() {
\$(this).val("Stop that!");
});// end click
});// end ready
</script >
\$('\#menu').hover(function (){
\$('\#submenu').show();
}, function (){
\$('\#submenu').hide();
});// end hover
```

Chapter 39

Lecture 39

Event properties. String.fromcharcode(evt.which) evt.preventdefault(); or return false; to stop normal behavior e.g. Links, form submit etc. Remove events $('.tabbutton').unbind('click'); default event bubbling and can stop it evt.stoppropagation(); generic way to bind events (click, mouseover etc. Special) $('#selector').bind('click', mydata, functionname); $('selector').bind('click', functionname); equivalent is $('selector').click(functionname); can bind multiple events

```
\$(document).click(function (evt){
Var xpos = evt.pagex;
Var ypos = evt.pagey;
```

Event property	Description
pageX	The distance (in pixels) of the mouse pointer from the left edge of the browser window.
pageY	The distance (in pixels) of the mouse pointer from the top edge of the browser window.
screenX	The distance (in pixels) of the mouse pointer from the left edge of the monitor.
screenY	The distance (in pixels) of the mouse pointer from the top edge of the monitor.
shiftKey	Is *true* if the shift key is down when the event occurs.
which	Use with the keypress event to determine the numeric code for the key that was pressed (see tip, next).
target	The object that was the "target" of the event—for example, for a *click()* event, the element that was clicked.
data	A jQuery object used with the *bind()* function to pass data to an event handling function (see page 177).

```
Alert('X:' + xpos+ ' Y:' + ypos);
});// end click


\$('selector').bind('click', functionname);
\$('\#selector').bind('click', mydata, functionname);
\$(document).bind('click keypress', function (){
\$('\#lightbox').hide();
});// end bind
\$('\#theelement').bind('click', function (){
// do something  interesting
});// end bind
\$('\#theelement').bind('mouseover', function (){
// do something else interesting
}
// end bind
\$('\#theelement').bind({
```

```
'Click'  : function()  {
// do something  interesting
}, // end click function
'Mouseover'  : function()  {
// do something  interesting
};// end mouseover  function
});// end bind
```

FAQ example.

```
<script src="../_js/jquery-1.6.3.min.js"></script>
<script>
\$(document).ready(function()  {
\$('.answer').hide();
\$('\#main h2').toggle(
Function()  {
\$(this).next('.answer').fadein();
\$(this).addclass('close');
},
Function()  {
\$(this).next('.answer').fadeout();
\$(this).removeclass('close');
}
);//end toggle
}); 16  </script>
```

Jquery animations. $('element').fadeout('slow');. Fast normal slow or number of ms . Default 200 400 600. Fadein, fadeout, fadetoggle. Slidedown, slideup, slidetoggle.

Login slider example

```
\$(document).ready(function (){
\$('\#open').toggle(
Function (){
\$('\#login form').slidedown(300);
\$(this).addclass('close');
},
Function (){
\$('\#login form').fadeout(600);
\$(this).removeclass('close');
}
);// end toggle
});// end ready
```

Generic animate. Any numeric css prop. Border-left-width becomes borderleftwidth cuz JS doesnt understand hyphen even += -=.

```
\$('\#message').animate(
{
Left: '650px',
Opacity: .5,
Fontsize:'24px'
},
1500
);
\$('\#moveit').click(function() {
\$(this).animate(
{
Left:'+=50px'
},
1000);
});
```

Easing. Linear or swing $('#element').slideup(1000,'linear'); can event pass a function to run when the animation finishes

```
\$('\#element').slideup(1000,'linear');
\$('\#photo').fadein(1000, function (){
\$('\#caption').fadein(1000);
});
\$('\#photo').width(0).height(0).css('opacity', 0);
\$('\#caption').hide();
\$('\#photo').animate(
{
Width:'200px',
Height: '100px',
Opacity: 1
},
1000,
Function() {
\$('\#caption').fadein(1000);
}
);// end animate
```

Can chain effects.

$('#photo').fadein(1000).delay(10000).fadeout(250);

Photo gallery example.

```
\$('\#gallery a').click(function(evt) {
Evt.preventdefault();
Var imgpath = \$(this).attr('href');
Var oldimage = \$('\#photo img');
Var newimage = \$('<img src="' + imgpath+'">');
Newimage.hide();
\$('\#photo').prepend(newimage);
Newimage.fadein(1000);
Oldimage.fadeout(1000,function(){
\$(this).remove();
});// end fadeout
});// end click
\$('\#gallery a:first').click();
```

Forms.

```
<input name="quantity" type="text" id="quantity">
<input name="total" type="text" id="total">
Var unitcost = 9.95;
Var amount = \$('\#quantity').val();
Var total= amount * unitcost;
Total = total.tofixed(2);
\$('\#total').val(total);
```

Selector	Example	What it does
:input	$(':input')	Selects all input, textarea, select, and button elements. In other words, it selects all form elements.
:text	$(':text')	Selects all text fields.
:password	$(':password')	Selects all password fields.
:radio	$(':radio')	Selects all radio buttons.
:checkbox	$(':checkbox')	Selects all checkboxes.
:submit	$(':submit')	Selects all submit buttons.
:image	$(':image')	Selects all image buttons.
:reset	$(':reset')	Selects all reset buttons.
:button	$(':button')	Selects all fields with type *button*.
:file	$(':file')	Selects all file fields (used for uploading a file).
:hidden	$(':hidden')	Selects all hidden fields.

Submit event.

```
\$(document).ready(function(){
\$('\#signup').submit(function(){
If(\$('\#username').val() == ''){
Alert('Please supply a name in the Name field.');
```

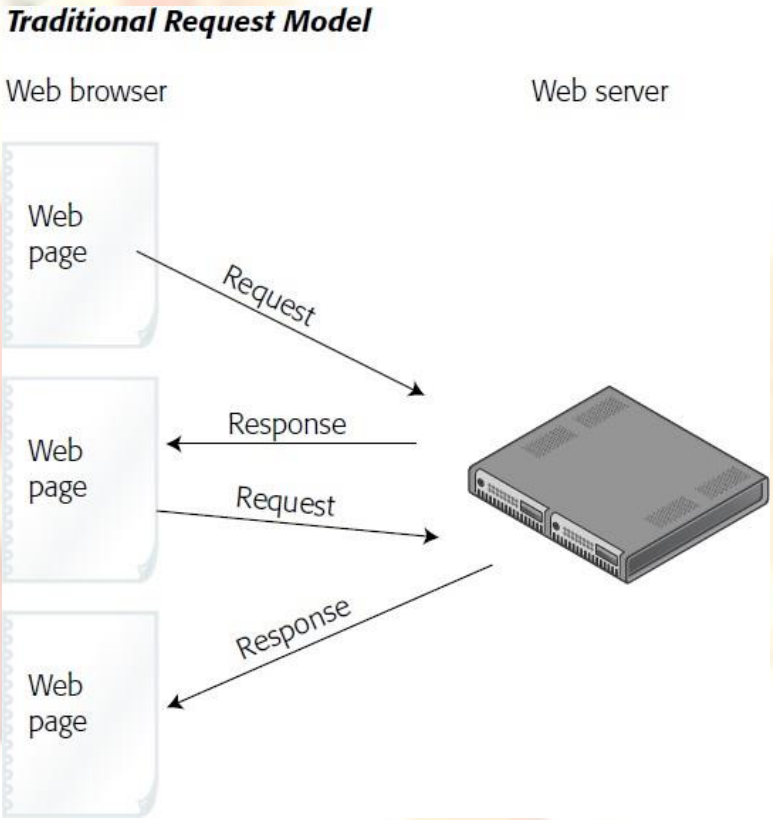
```
Return false;
}
});// end submit()
});// end ready()
```

Focus, blur, click, change (menus). React to choice from a list menu cant do everything at client. Page disappearing and reappearing. Ajax Let's webpage ask for info and update itself when it comes. Asynchronous JS and XML. Term coined in 2005 for interactive sites coming from google. Like google maps, gmail.

Chapter 40

Lecture 40

In Last Lecture, we talk about event object and its properties. Binding and unbinding events. Jquery animations, easing, chaining. FAQ, login slider, photo gallery. Forms and form selectors. Now the real power. Ajax. What can be done. Display NEW html content without reloading the page. Submit form and instantly display results. Login without leaving the page. E.g. Star rating widget. Browsing through database info like you scroll down on fb, twitter. Nothing radical. Except without loading a new page. Can achieve same with HTML and server side prog. Ajax make pages feel more responsive and desktop like.



JS, server-side programming, and web browser, all work together. Web browser: xmlhttprequest object. Makes ajax possible. Talks to web server and get response. JS: sends request, waits for response, process response, updates web page. Web server: receives request and responds as HTML, plain text, XML, JSON. Or application server for more complicated tasks. Need web server for ajax examples.

For talking to web server, we create xmlhttprequest (also called XHR in short) var newxhr = new XML-httprequest(); again browser incompatibilities call open to specify what kind of data and where it will go can GET or POST newxhr.open('GET', 'shop.php?Productid=34'); write a callback function it will remove, add, change el-ements send data newxhr.send(null); GET newxhr.send('q=javascript'); POST receive response callback invoked and XHR receives status, text response, and possibly an XML response

status = 200/304 all ok, 404 file not found,500 internal server error, 403 access forbidden responsetext has text of JSON or HTML, responsexml less commonly used.

The jquery simplifies all steps except that of changing the webpage simplest is load function which loads HTML into an area of web page e.g. Load news in a div from a web-server $('#headlines').load('todays news.html'); can only load from same site... Relative urls possible to add only a part $('#headlines').load('todays news.html #news');.

```
\$('\#headlines').load('todays_news.html');

\$('\#headlines').load('todays_news.html \#news');

\$('\#newslinks a').click(function (){
Var url = \$(this).attr('href');
\$('\#headlines').load(url+ ' \#newsitem');
Return false;
});
\$.get(url, data, callback);
\$.post(url, data, callback);
```

Get() and post(). Need server side to do anything else. Server may not return html e.g. Database records as xml or json. Jquery handles differences of GET and POST. No selector. Stand by themselves.

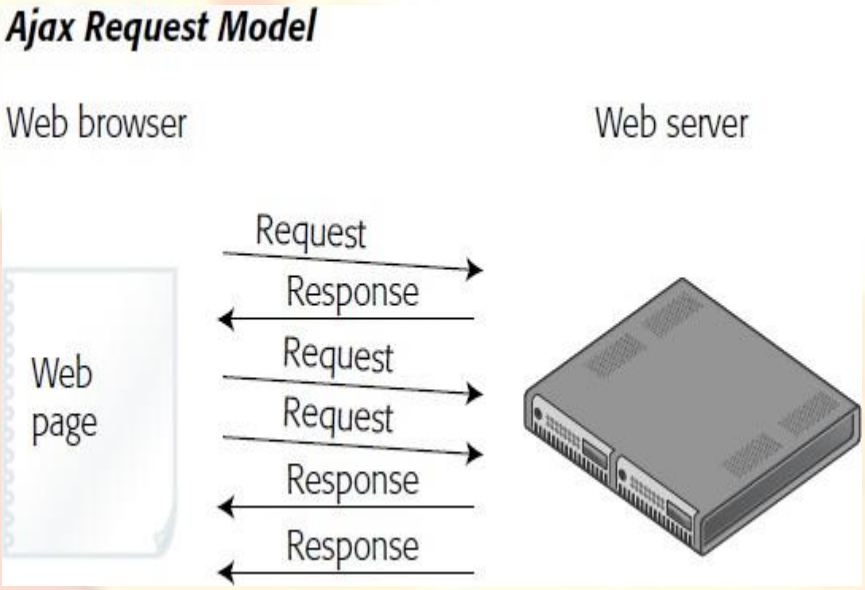
```
$.get('ratemovie.php', 'rating=5');
$.post('ratemovie.php', 'rating=5');
```

Formatting data. Can send a product number, entire form, signup. Format as query string of JS obj literal. URL http://www.chia-vet.com/prod- ucts.php?Prodid=18&sessid=1234. GET has limit. Often thousands of chars. $.get('ratemovie.php','rating=5'); $.post('ratemovie.php','rating=5'); $.post('ratemovie.php','rating=5&user=Bob'); 'favfood=Mac & Cheese' // incorrect 'favfood=Mac%20%26%20Cheese' // properly escaped var querystring = 'fav-Food=' + encodeuricomponent('Mac & Cheese'); $.post('foodchoice.php', querystring); better way is obj literal.

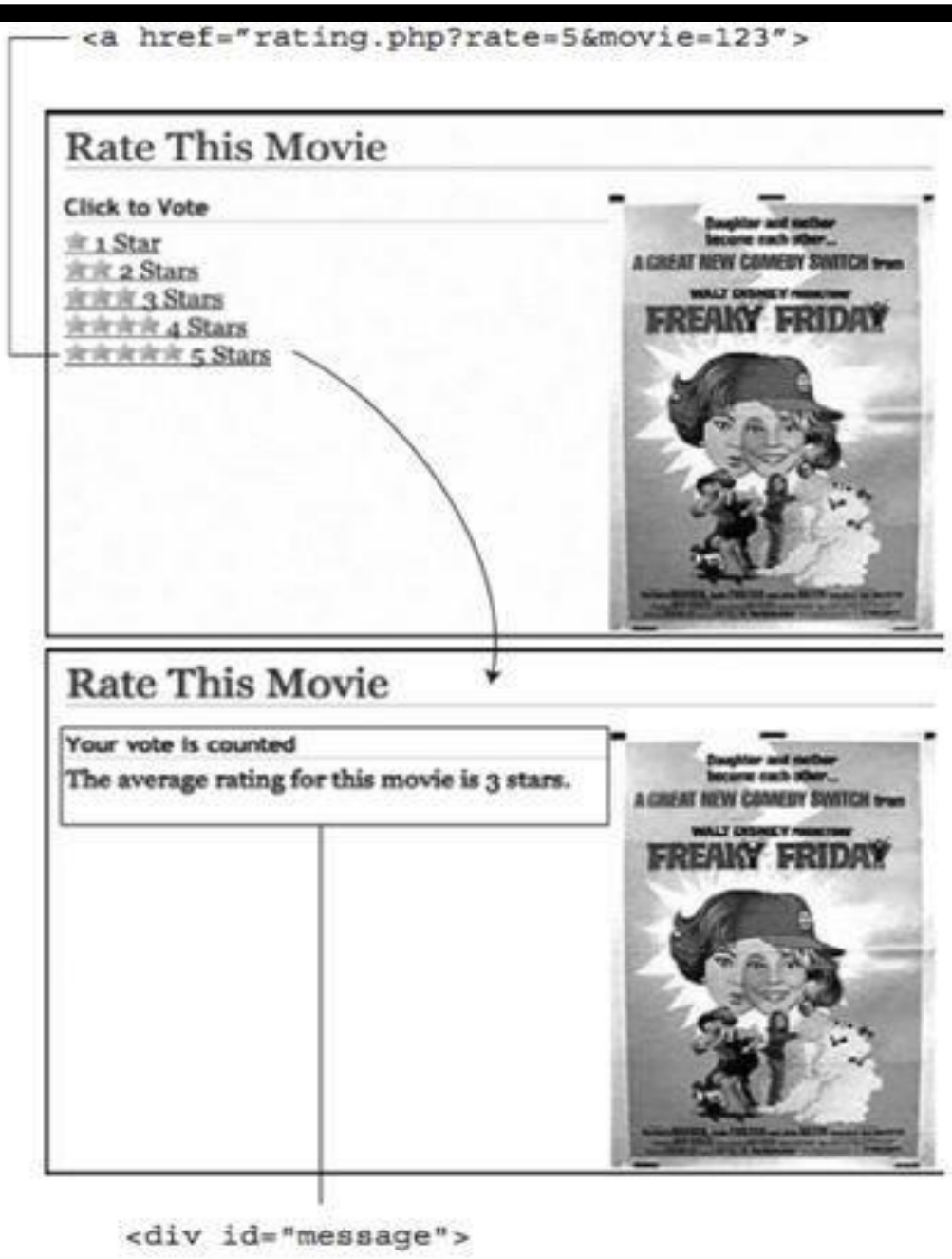
```
{
Name1:'value1',
Name2:'value2'
}

\$.post('rankmovie.php', { rating:5 });

Var data = { rating:5 };

\$.post('rankmovie.php', data);

Var data = {
Rating: 5,
User:'Bob'
}

\$.post('rankmovie.php', data);

Var data = \$.post('rankmovie.php',
{
Rating: 5,
User:'Bob' }
); // end post
```

Serialize using name/value of form elements var formdata = $('#login').serialize(); $.get('login.php',formdata,loginresults); processing data returned. Call back first arg is data. Servers often use XML of JSON. Second arg is string about status success. E.g. Movie rating.

```
Function processresponse(data, status){
Var newhtml;
Newhtml = '<h2>Your vote is counted</h2>';
Newhtml += '<p>The average rating for this movie is ';
Newhtml += data + '.</p>';
\$('\#message').html(newhtml);
//
}
```




```
\$('\#message a').click(function() {
Var href=\$(this).attr('href');
Var querystring=href.slice(href.indexof('?')+1);
\$.get('rate.php', querystring, processresponse);
Return false;// stop the link
});

Function processresponse(data) {
Var newhtml;
Newhtml = '<h2>Your vote is counted</h2>';
Newhtml += '<p>The average rating for this movie is ';
Newhtml += data + '.</p>';
\$('\#message').html(newhtml);
}
```



Error handler.

```
\$.get(url, data, successfunction).error(errorfunction);
\$.get('rate.php', querystring, processresponse).error(errorresponse);
Function  errorresponse(){
Var errormsg= "Your vote could not be processed right now.";
Errormsg += "Please try again later.";
\$('\#message').html(errormsg);
}
```

JS format. Method for exchanging data. JSON is JS so its quick n easy for JS. No XML like parsing. JSON is a JS obj literal. (MUST use quotations if names have spaces etc.)

```
{
Firstname: ’Frank’,
Lastname:’Smith’,
Phone:’503-555-1212’
}
```

```
{
’Firstname’:’Frank’,
’Lastname’:’Smith’,
’Phone’:’503-555-1212’
}
```

Server returns a string formatted like a JSON obj literal. Jquery getjson method. Callback will be a JSON object.

```
Var bday  = {
Person: ’Raoul’,
Date:  ’10/27/1980’
};
Bday.person // ’Raoul’
Bday.date  //’10/27/1980’
```

Obj literals can be composed of other obj literals

```
Var data  = {
Contact1:  {
Firstname: ’Frank’,
Lastname: ’Smith’,
Phone:  ’503-555-1212’
},
Contact2:  {
Firstname: ’Peggy’,
Lastname: ’Jones’,
Phone:  ’415-555-5235’
}
}; 13
Data.contact1.firstname
\$.each(JSON, function  (name, value)  { 16  });
```

Like get but data passed to

\$.getjson(’contacts.php’,’limit=2’, processcontacts);

Chapter 41

Lecture 41

Objective-C introduces smalltalk style messaging in C. Early 1980s. Used for next computer. It is simple extension of C. To create an object, send it an alloc message. Nsmutablearray *arrayinstance = [nsmutablearray alloc];. Initialize it. [arrayinstance init];. Can combine. Nested message send. Nsmutablearray *arrayinstance = [[nsmutablearray alloc] init];. Message = [receiver selector arguments]. E.g. Add obj to array. [arrayinstance addobject:anotherobject];.

Another message you can send to mutablearray. Replaceobjectsinrange:withobjectsfromarray:range:. Pairing of labels and arguments a feature of objective-C. In other languages, Arrayinstance.replace object sinrange with objects from array range (anotherarray, anotherrange); In objective-C. [arrayinstance replaceobjectsinrange:arange withobjectsfromar-ray:anotherarray range:anotherrange];.Destroy using [arrayinstance release];. Should also arrayinstance = nil;. Otherwise dangling. But sending message to nil is ok. Nil is like null.

```
Int main (int argc, const char* argv[])
{
Nsautoreleasepool *pool=[[nsautoreleasepool alloc] init];
Nsmutablearray *items= [[nsmutablearray alloc] init];
[items addobject:@"One"];
[items addobject:@"Two"];
[items addobject:@"Three"];
[items insertobject:@"Zero" atindex:0];
For(int i= 0;i< [items count]; i++){
Nslog(@"%@",[items objectatindex:i]);
}
[items release];
Items= nil;
[pool drain];
Return0;
}
```

Nsstring. [items addobject:@”One”]; @ is shortcut for creating nsstring. Nsstring *mystring = @”Hello, World!”;. Int len = [mystring length];. Len = [@”Hello, World!” Length];. Mystring = [[nsstring alloc] initwithstring:@”Hello, World!”];. Len = [mystring length];.Nslog. Format string. Int a = 1; float b = 2.5; char c = ’A’; nslog(@”Integer: Integer: 1 Float: 2.5 Char: A Nsarray and nsmutablearray. Also nsdictionary and nsset. Holds references. Cannot hold primitives and C structures. Can call [array count]. Int numberofobjects = [array count]; [array insertobject:object atindex:numberofobjects]; cant add beyond end. Exception. [array addobject:[nsnull null]]; // otherwise cannot hold nil. Nsstring *object = [array objectatindex:0];

Subclassing. Root class of entire hierarchy. Nsobject. Objective-C keywords start with @. Instance variables.

```
#import <uikit/uikit.h>
@interface  Possession: nsobject
{
}
@end
#import  <Foundation/Foundation.h>
@interface  Possession: nsobject
```

```
{
Nsstring *possessionname;
Nsstring  *serialnumber;
Int valueindollars;
Nsdate  *datecreated;
}
@end


#import  <Foundation/Foundation.h>
@interface  Possession: nsobject
{
Nsstring *possessionname;
Nsstring  *serialnumber;
Int valueindollars;
Nsdate  *datecreated;
}
- (void)setpossessionname:(nsstring  *)str;
- (nsstring  *)possessionname;

- (void)setserialnumber:(nsstring  *)str;
- (nsstring  *)serialnumber;
- (void)setvalueindollars:(int)i;
- (int)valueindollars;
- (nsdate  *)datecreated;
@end 17

#import "Possession.h"
@implementation  Possession
// Getter
- (nsstring  *)possessionname
{
// Returna pointer to the object this Possession calls its possessionname
Return  possessionname;
}
// Setter
-  (void)setpossessionname:(nsstring  *)newpossessionname 29 {
// Change the instance  variable to point at another string,
// this Possession will now call this new string its possessionname
Possessionname= newpossessionname;
}
// Createa new Possession instance
Possession *p=[[Possession alloc] init];
// Set possessionname to a new nsstring
[p setpossessionname:@"Red Sofa"];
// Get the pointer of the Possession's possessionname
Nsstring *str=[p possessionname];
// Print that object
Nslog(@"%@", str); // This would print "Red Sofa"
```

Getter setters can be made like above example. Let's see instance methods (and overriding).

```
- (nsstring*)description
{
Nsstring *descriptionstring=
[[nsstring alloc] initwithformat:@"\%@(\%@): Worth\$\%d, recorded on\%@",
```

```
Possessionname,
Serialnumber,
Valueindollars,
Datecreated];
Return  descriptionstring;
}
```

Initializers start with init naming convention. Id = any object. Every object has isa pointer to class and thats how methods are called. Like vtable. Self and super. Other initializers.

```
- (id)initwithpossessionname:(nsstring *)name
Valueindollars:(int)value
Serialnumber:(nsstring*)snumber;
- (id)initwithpossessionname:(nsstring *)name
Valueindollars:(int)value
Serialnumber:(nsstring*)snumber
{
// Call the superclass's designated  initializer
Self=[super init];
// Did the superclass's designated  initializer  succeed?
If(self){
// Give the instance  variables  initial values
[self setpossessionname:name];
[self setserialnumber:snumber];
[self setvalueindollars:value];
Datecreated=[[nsdate alloc] init];
}
// Return the address of the newly initialized  object
Return self;
}
- (id)init
{
Return[self initwithpossessionname:@"Possession"
Valueindollars:0
Serialnumber:@""];
}
```

Class methods

```
@interface  Possession: nsobject
{3 nsstring *possessionname;
Nsstring *serialnumber;
Int valueindollars;
Nsdate*datecreated;
}
+ (id)randompossession;
- (id)initwithpossessionname:(nsstring *)name
Valueindollars:(int)value
Serialnumber:(nsstring*)snumber;
+ (id)randompossession
{
// Create an array of three adjectives
Nsarray *randomadjectivelist=[nsarray arraywithobjects:@"Fluffy",
@"Rusty",
@"Shiny", nil];
// Create an array of three nouns
Nsarray *randomnounlist=[nsarray arraywithobjects:@"Bear",
@"Spork",
```

```
@"Mac", nil];
Int adjectiveindex= rand()\%[randomadjectivelist count];
Int nounindex= rand()\% [randomnounlist count];
Nsstring *randomname=[nsstring stringwithformat:@"\%@ \%@",
[randomadjectivelist objectatindex:adjectiveindex],
[randomnounlist  objectatindex:nounindex]];
Int randomvalue= rand()\%100;
Nsstring *randomserialnumber= [nsstring stringwithformat:@"\%c\%c\%c\%c\%c",
'0'+  rand()\% 10,
'A'+ rand()\%26,
'0'+  rand()\% 10,
'A'+ rand()\%26,
'0'+  rand()\% 10];
// Once again, ignore the memory problems with this method
Possession *newpossession=
[[self alloc] initwithpossessionname:randomname
Valueindollars:randomvalue
Serialnumber:randomserialnumber];
Return  newpossession;
}
```

Obj-C is dynamically typed. Nsmutablearray *items = [[nsmutablearray alloc] init]; [items dosomethingweird]; 2009-07-19 01:34:53.602 randompossessions[25326:10b]. *** -[nscfarray dosomethingweird]: unrecognized selector sent to instance 0x104b40. Objective-C has try-catch. Usually for runtime errors that the programmer made better for loop.

Properties, simplified accessors.

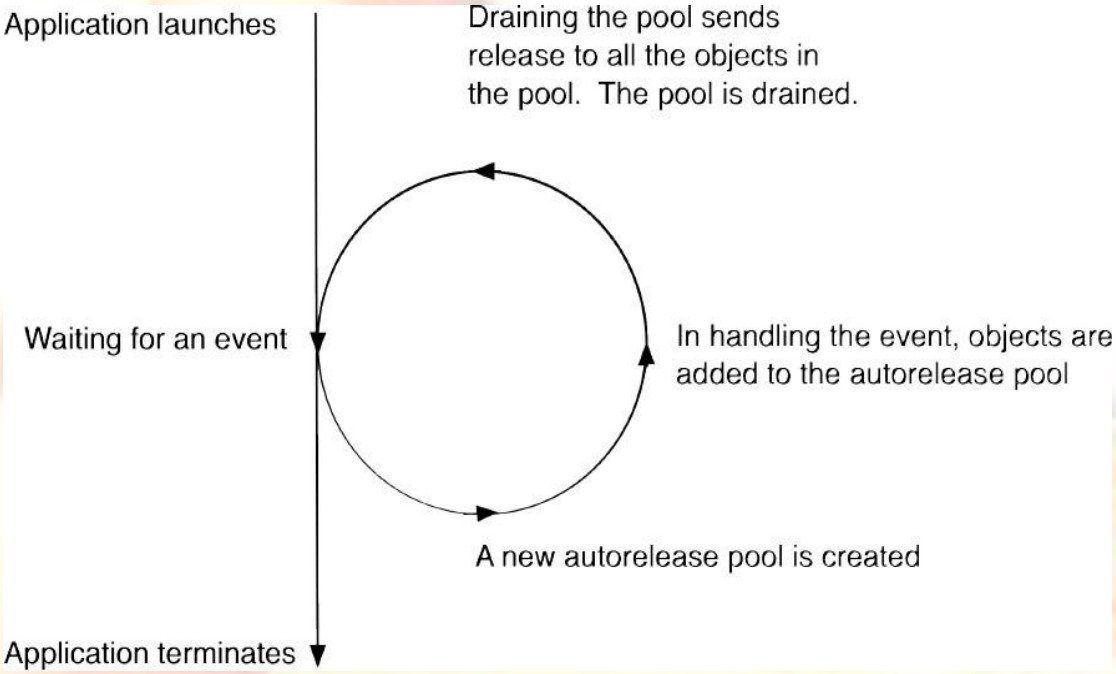
```
@interface  Possession: nsobject
{
Nsstring *possessionname;
Nsstring *serialnumber;
Int valueindollars;
Nsdate*datecreated;
}
+ (id)randompossession;
- (id)initwithpossessionname:(nsstring *)name
Valueindollars:(int)value
Serialnumber:(nsstring*)snumber;
- (id)initwithpossessionname:(nsstring *)name;
@property  nsstring*possessionname;
@property nsstring*serialnumber;
@property int valueindollars;
@property nsdate *datecreated;
@end
@implementation  Possession
@synthesize possessionname, serialnumber, valueindollars, datecreated;
@property(nonatomic, copy) nsstring*possessionname;
@property(nonatomic, copy) nsstring*serialnumber;
- (void)setpossessionname:(nsstring *)str
{
Id t = [str copy];
[possessionname  release];
Possessionname = t;
}
```

Copy and mutablecopy.

Chapter 42

Lecture 42

Alloc and dealloc methods. Manual reference counting. Obj knows owner count retaincount. Retain and release methods. Should you release a created object that is returned ?. Want to say don't release but i don't want to be the owner. Autorelease. Added to nsautoreleasepool. Nsobject *x = [[[nsobject alloc] init] autorelease];.




```
Nsobject *x = [[[nsobject alloc] init] autorelease];
- (nsstring*)description
{
Nsstring *descriptionstring=
[[nsstring alloc] initwithformat:@"\%@(\%@): Worth \$\%d, Recorded on \%@",
Possessionname,
Serialnumber,
Valueindollars,
Datecreated];
Return  [descriptionstring  autorelease];
}
 - (nsstring  *)description
{
Return  [nsstring  stringwithformat:@"\%@  (\%@): Worth \$\%d, Recorded on  \%@",
Possessionname,
Serialnumber,
Valueindollars,
Datecreated];
}
- (void)setpossessionname:(nsstring *)str
{
```

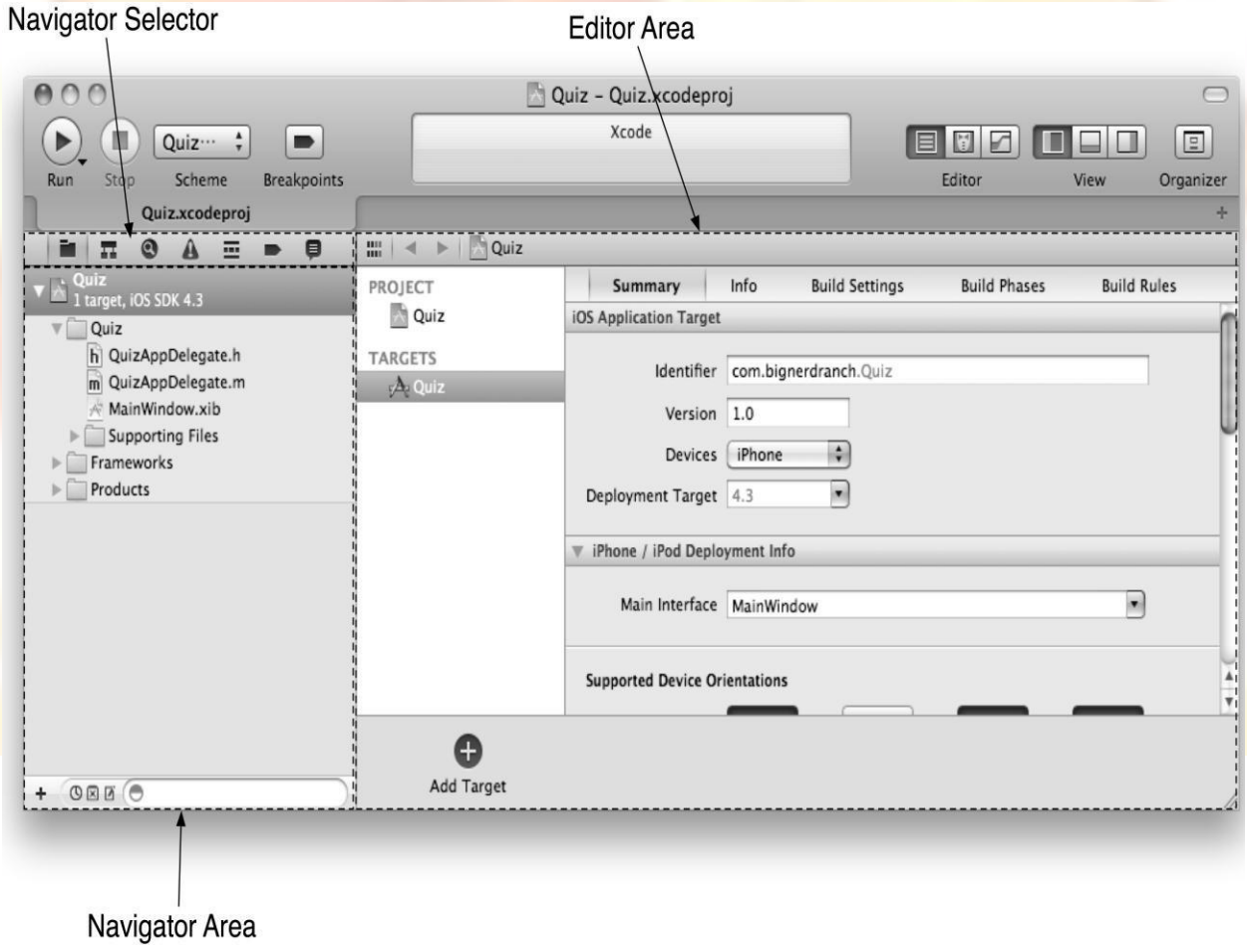
```
[str retain];
[possessionname  release];
Possessionname = str;
}
- (void)dealloc
{
[possessionname  release];
[serialnumber release];
[datecreated release];
[super dealloc];
}
```

Retain count rules. Init, new, copy in name. Assume you own. Any other means. Assume in autorelease. If you dont own and want to make sure, call retain. No longer need and own than release or autorelease. When 0 count, dealloc called.

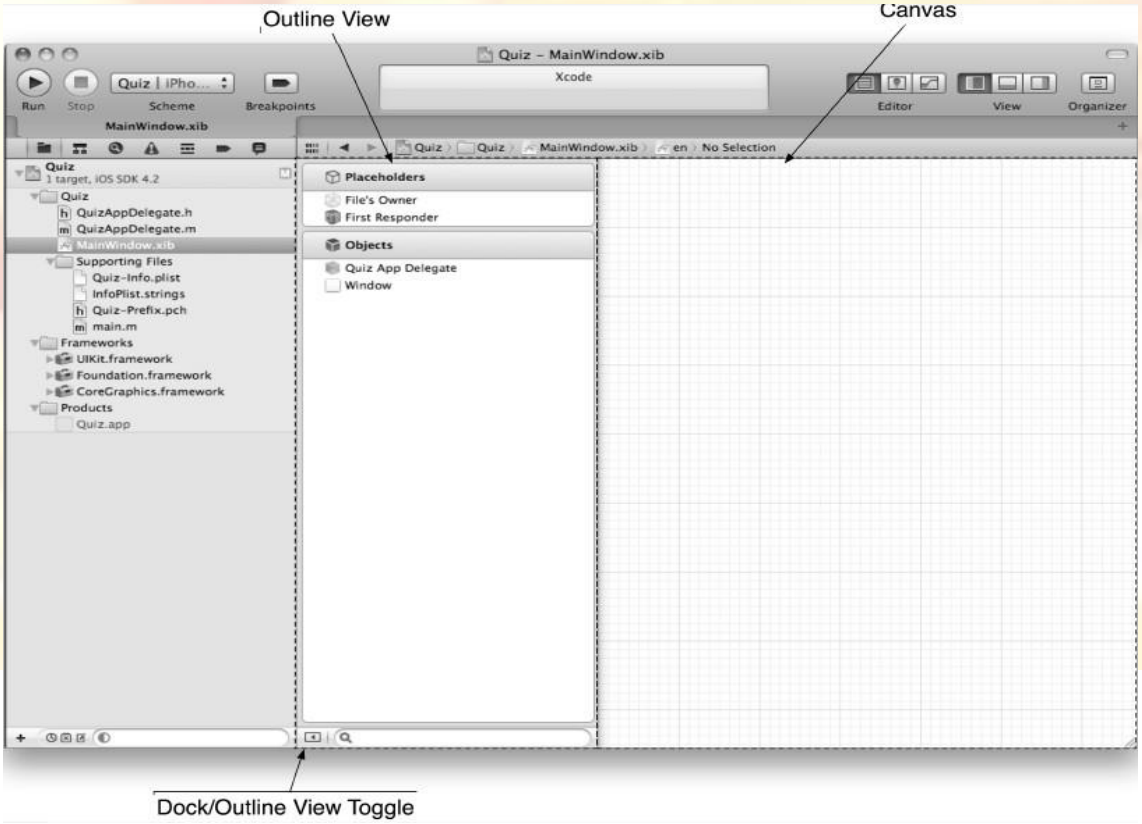
Protocols i.e. Interfaces.

Last lecture about ajax. Really learned how similar event driven programming is in JS than wpf. What about mobile. Same paradigm, different language. Same concepts, different incarnation. Ios programming. Learn objective-C. Event driven programming. Well see examples but you may not be able to try them. Need a mac computer and Xcode, teh ios simulator. Let's make a simple app.

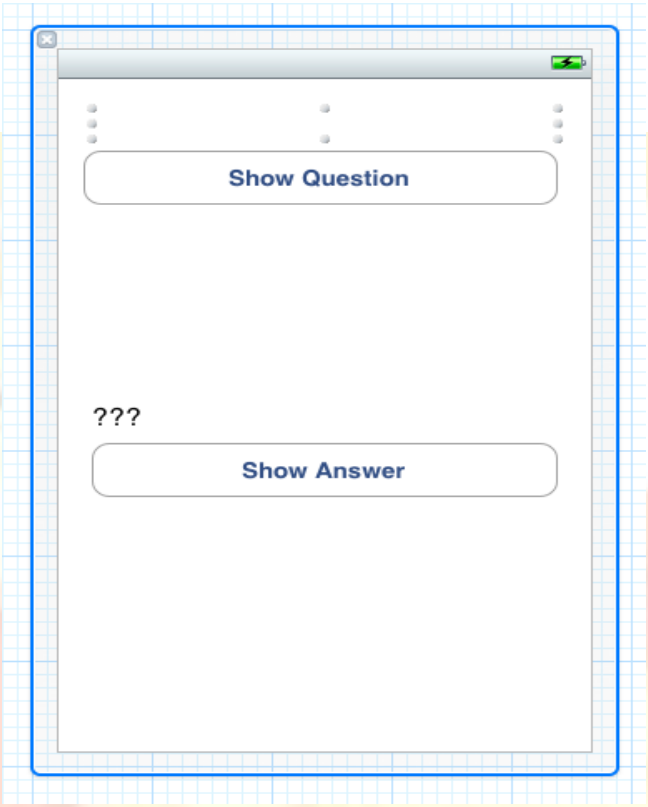
A quiz showing a question and revealing the answer. Create new project (window based applica.). Name it.



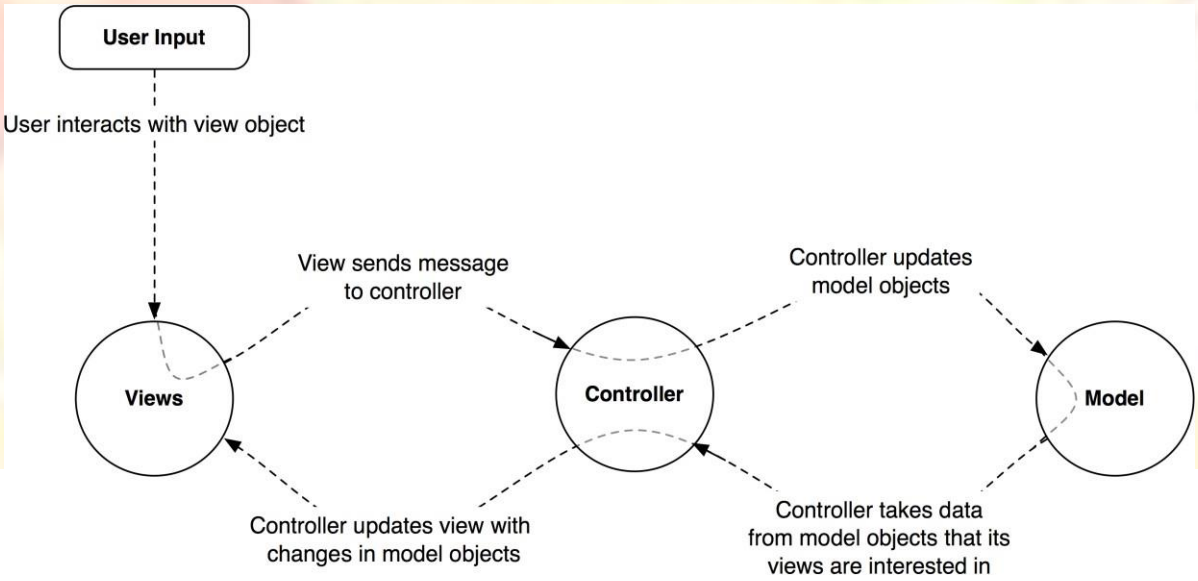
Similar to visual studio. The interface in xml. Xib file. Compiled to a nib file. Ios application a directory containing executables and resources. Sounds familiar.

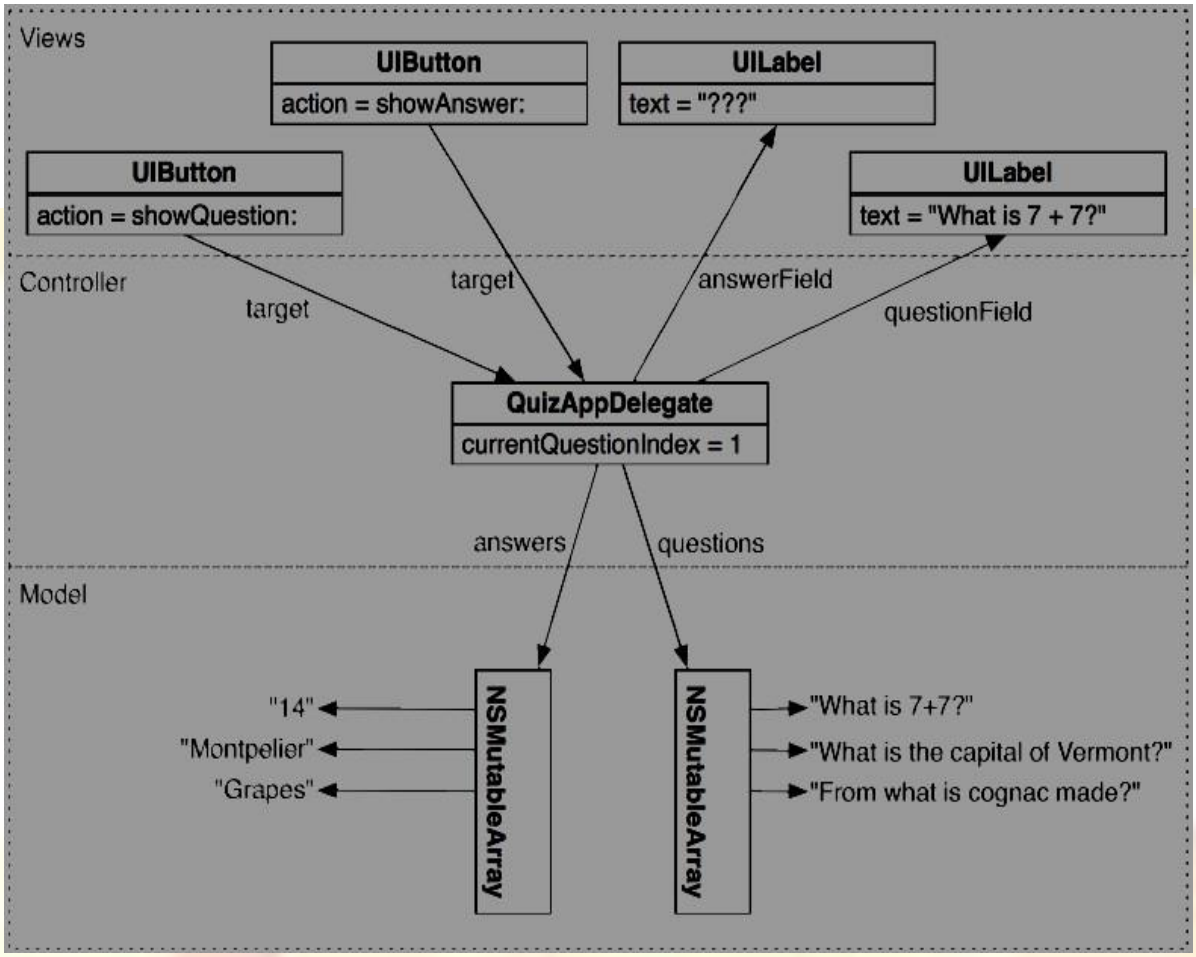


Xib editing. Select window to add controls. Drag buttons. Give them names.



Mvc pattern. View objects. Visible things. Uiview subclasses. Model objects. Hold data and know nothing about interface. Often use standard containers. Controllers keep things in sync.





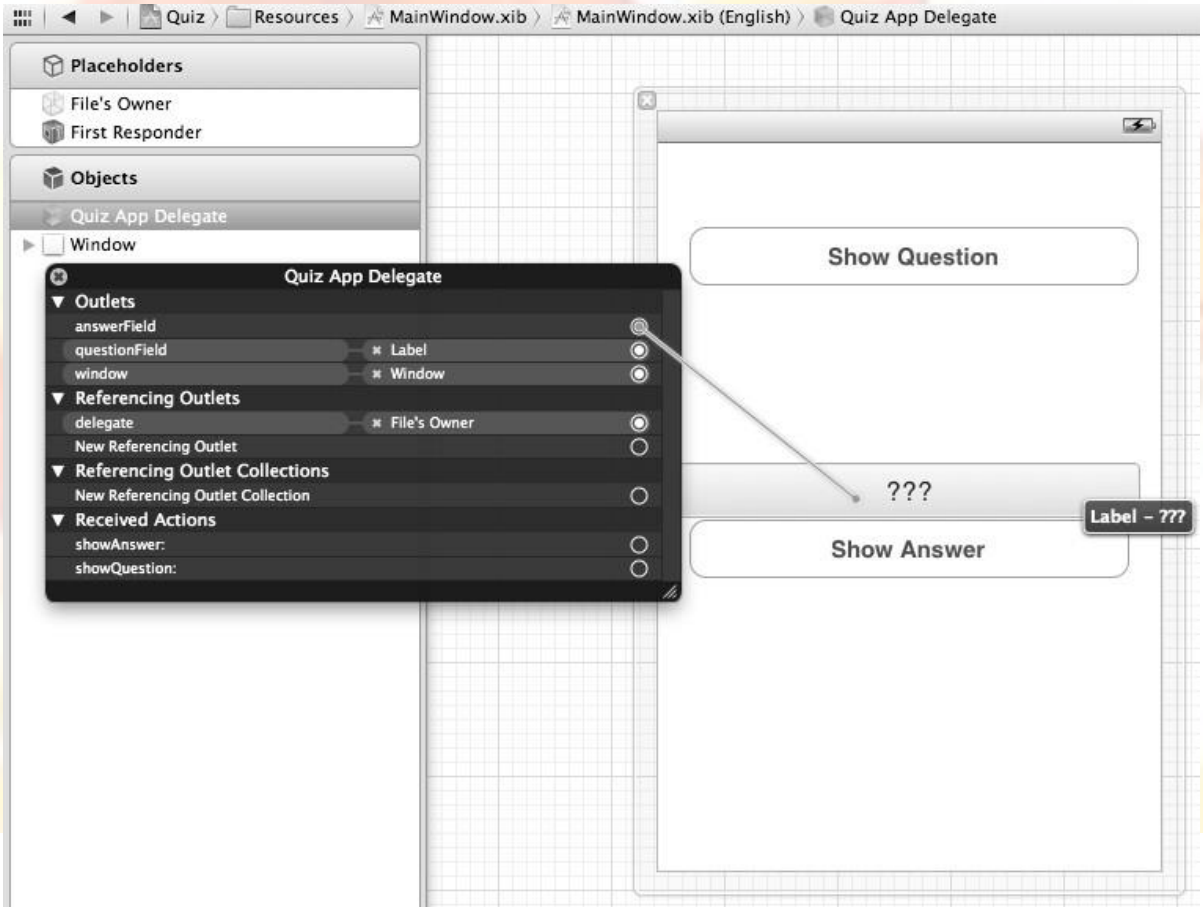
Iboutlet, ibaction.

```
@interface  quizappdelegate: nsobject
{
Int currentquestionindex;
// The model objects
Nsmutablearray  *questions;
Nsmutablearray  *answers; 7 // The view objects
<uiapplicationdelegate>

Iboutlet uilabel *questionfield;

Iboutlet uilabel *answerfield;

}

@property(nonatomic, retain) iboutlet  uiwindow *window;

- (ibaction)showquestion:(id)sender;

- (ibaction)showanswer:(id)sender;
```

```
@end
```

Setting connections. From the obj with pointer to the obj you want that pointer to point at. Control-drag from target to object. Check connection in connection inspector.

```
@interface  quizappdelegate: nsobject <uiapplicationdelegate>
{
Int currentquestionindex;
// The model objects
Nsmutablearray *questions;
Nsmutablearray *answers;
// The view objects
Iboutlet uilabel *questionfield;
Iboutlet uilabel *answerfield;
}
@property(nonatomic, retain) iboutlet  uiwindow *window;

- (ibaction)showquestion:(id)sender;
- (ibaction)showanswer:(id)sender;
@end
```




```
@implementation  quizappdelegate
- (id)init
{
// Call the init method  implemented by the superclass
Self  =  [super init]; 6  if(self){
// Create two arrays and make the pointers point to them
Questions= [[nsmutablearray alloc] init];
Answers= [[nsmutablearray  alloc] init];
// Add questions and answers to the arrays
[questions addobject:@"What is7 + 7?"];
[answers  addobject:@"14"];
[questions  addobject:@"What is the capital of Vermont?"];
[answers  addobject:@"Montpelier"];
[questions  addobject:@"From what is cognac made?"];
[answers  addobject:@"Grapes"];
}
// Return the address of the new object
Return self;
}
- (ibaction)showquestion:(id)sender
{
// Step to the next question
Currentquestionindex++;
// Am I past the last question?
If(currentquestionindex == [questions count]){
// Go back to the first question
```

```
Currentquestionindex= 0;
}

// Get the string at that index in the questions array
Nsstring *question= [questions  objectatindex:currentquestionindex];
// Log the string to the console
Nslog(@"displaying  question:%@", question);
// Display the string in the question field
[questionfield  settext:question];
// Clear the answer field
[answerfield  settext:@"???"];
}
- (ibaction)showanswer:(id)sender
{
// What is the answer to the current  question?
Nsstring *answer= [answers  objectatindex:currentquestionindex];
// Display it in the answer field
[answerfield  settext:answer];
}
```

Chapter 43

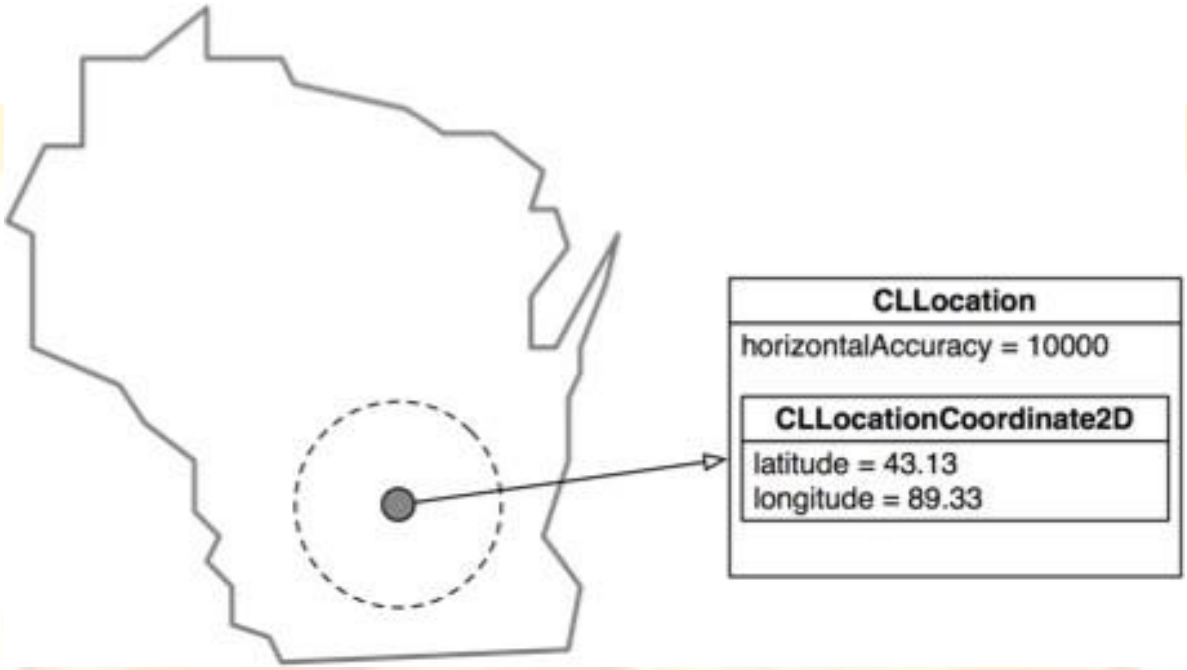
Lecture 43

In Last Lecture, we discussed objective C memory management rules, Wrote our first iphone app: a quiz app, xib and nib files and interface editor, MVC pattern, iboutlet ibaction, Connection Inspector, then wrote init, showquestion, and showanswer.

Core Location Framework. Let's make a whereami application. Classes that enable finding geographical position. Distancefilter and desiredaccuracy properties.


```
#import  <uikit/uikit.h>
#import <corelocation/corelocation.h> 3
@interface  whereamiappdelegate: nsobject
{ <uiapplicationdelegate>
Cllocationmanager *locationmanager;
}
@property(nonatomic, retain) iboutlet uiwindow *window;
@end
- (BOOL)application:(uiapplication*)application
Didfinishlaunchingwithoptions:(nsdictionary *)launchoptions
{
// Create location manager object
Locationmanager=[[cllocationmanager  alloc] init];
// We want all results from the location manager
[locationmanager  setdistancefilter:kcldistancefilternone];
// And we want it to be as accurate as possible
// regardless of how much time/power it takes
[locationmanager  setdesiredaccuracy:kcllocationaccuracybest];
// Tell our manager to start looking for its location  immediately
[locationmanager  startupdatinglocation];
// This line may say self.window, don't worry about that
[[self window] makekeyandvisible];
Return YES;
}
```

Delegation. Locationmanager:didupdatetolocation:fromlocation: sent to delegate which we want



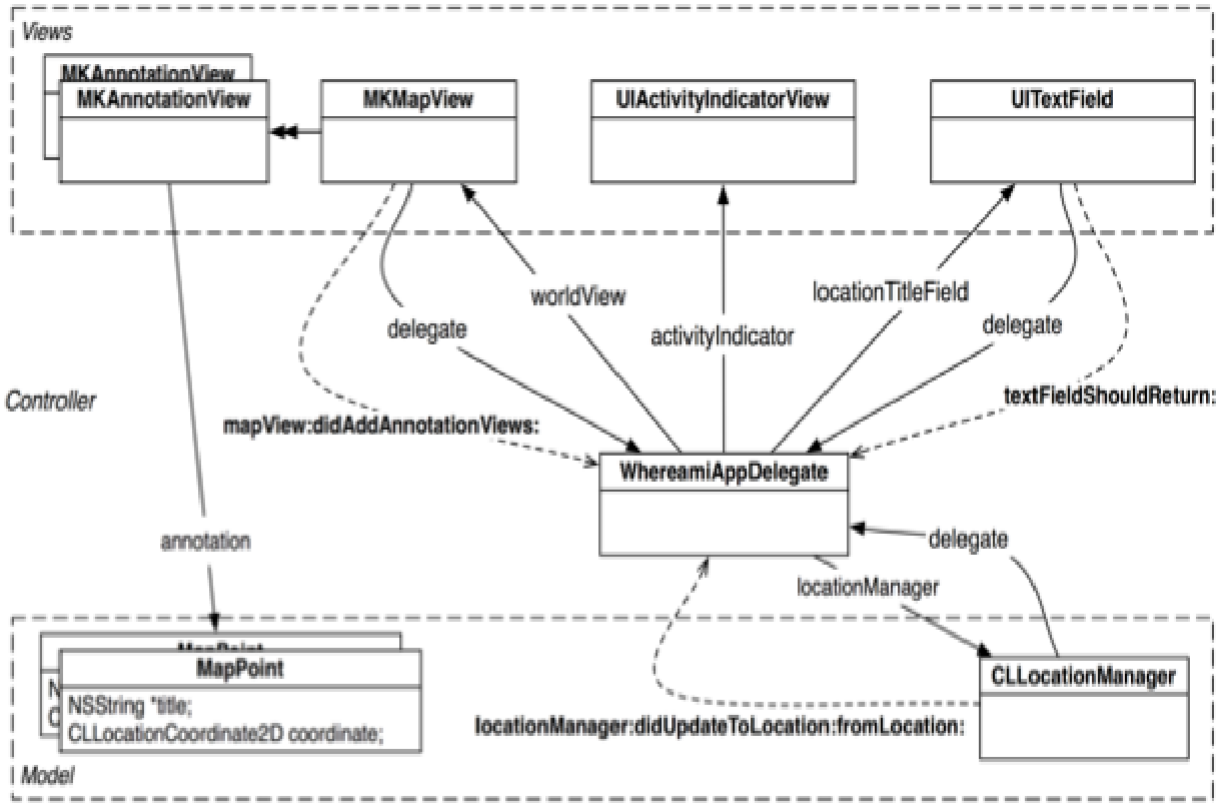
whereamiap-pdelegate to be.

```
- (void)locationmanager:(cllocationmanager *)manager
Didupdatetolocation:(cllocation *)newlocation
Fromlocation:(cllocation *)oldlocation
{
Nslog(@"%@", newlocation);
}
- (void)locationmanager:(cllocationmanager *)manager
Didfailwitherror:(nserror*)error
{
Nslog(@"Could not find location:%@", error);
}
```

Delegation is a design pattern. An OO approach to callbacks. Allows callback methods to share data. A delegatecan only be sent messages specified in its protocol. For every object that can have a delegate, there is a corresponding protocol.

```
@protocol cllocationmanagerdelegate <nsobject>
@optional
- (void)locationmanager:(cllocationmanager *)manager
Didupdatetolocation:(cllocation *)newlocation
Fromlocation:(cllocation *)oldlocation;
- (void)locationmanager:(cllocationmanager *)manager
Didupdateheading:(clheading *)newheading;
- (BOOL)locationmanagershoulddisplayheadingcalibration:(cllocationmanager*)manager;
- (void)locationmanager:(cllocationmanager *)manager
Didenterregion:(clregion *)region;
- (void)locationmanager:(cllocationmanager *)manager
Didfailwitherror:(nserror*)error;
```

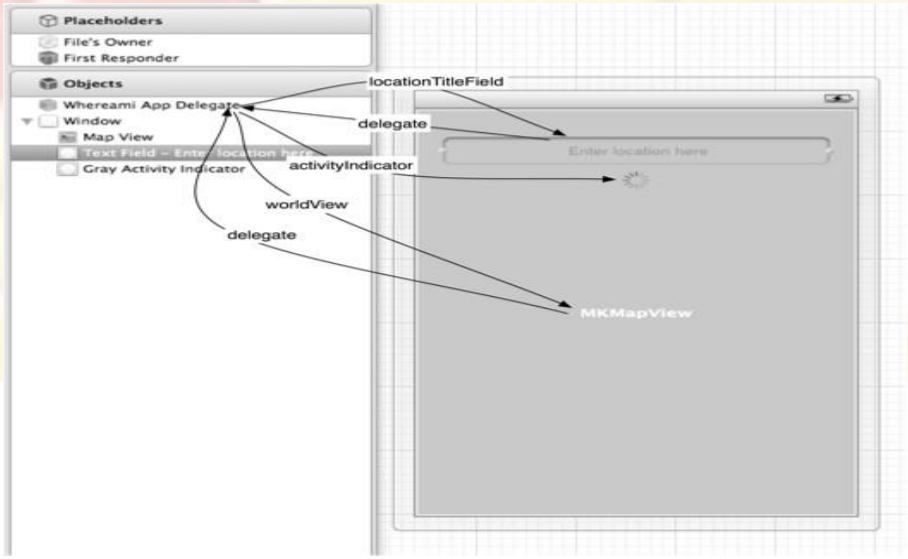
```
@end

 (void)finishedfindinglocation:(cllocation *)newlocation

EL updatemethod=@selector(locationmanager:didupdatetolocation:fromlocation:);
F ([[self delegate] respondstoselector:updatemethod]){
/ If the method is implemented, then we send the message.
[self delegate] locationmanager:self
Idupdatetolocation:newlocation
Romlocation:oldlocation];
Interface whereamiappdelegate: nsobject
Uiapplicationdelegate, cllocationmanagerdelegate>
(void)dealloc

F([locationmanager delegate]== self)
Locationmanager setdelegate:nil];
Locationmanager  release];
Window release];
Super dealloc];
```



Protocols are like interfaces in other langs i.e. No implementation. Protocols for delegations delegate protocols. Can be used to ask things from the delegate or inform of updates. Can be @optional methods. By default required. Here all were optional. Every object implements respondstoselector: something like this (above). Class has to declare protocols it implements like other langs. No warning anymore.

Memory and delegation. Delegates are usually controllers. Delegates never retained. Retain cycle. Assign attrib. Weak reference.

Let's display a map of current location.



Several instances of mkannotationview appear as icons on the mkmapview. An mkmapview displays the map and the labels for the recorded locations. A uiactivityindicatorview indicates that the device is working and not stalled. A uitextfield allows the user to input text to label the current location on the map.

Drag the map view onto uiwindow.

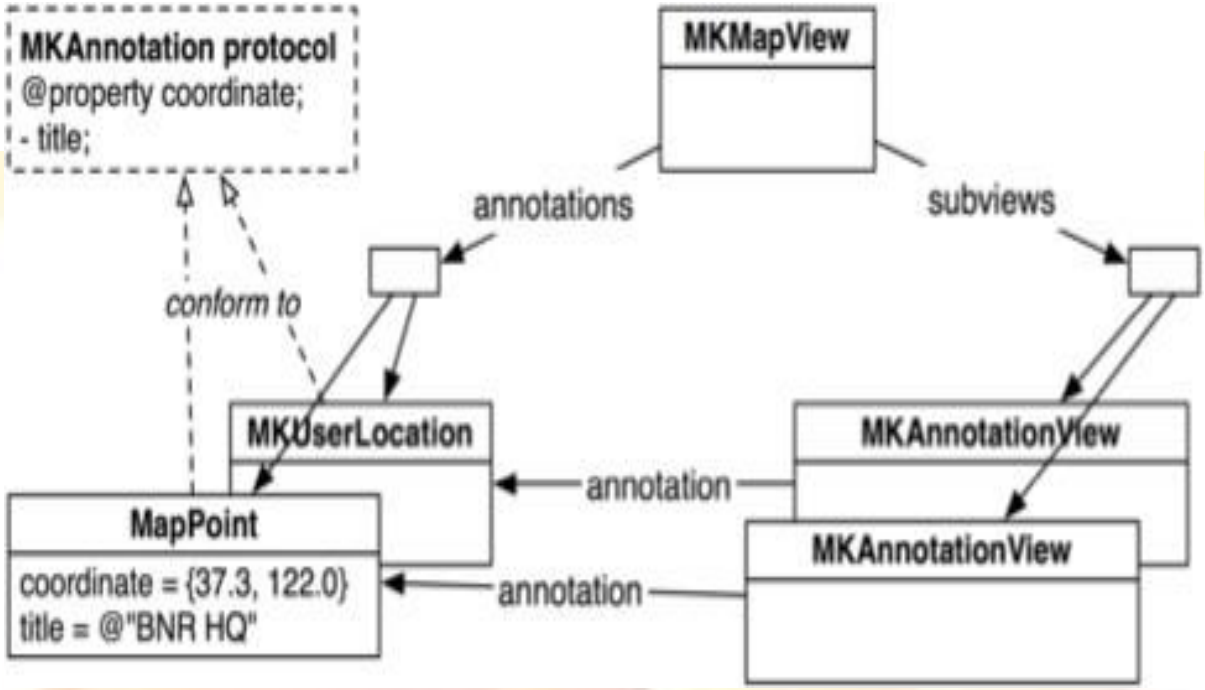
```
#import <mapkit/mapkit.h>

@interface  whereamiappdelegate: nsobject
<uiapplicationdelegate, cllocationmanagerdelegate>
{
Cllocationmanager *locationmanager;
Iboutlet  mkmapview*worldview;
Iboutlet uiactivityindicatorview *activityindicator;
Iboutlet uitextfield *locationtitlefield;
}
@property(nonatomic, retain) iboutlet uiwindow *window;
@end
```

All we do is set showsuserlocation property of mkmapview and it will show users location.

```
- (BOOL)application:(uiapplication*)application
Didfinishlaunchingwithoptions:(nsdictionary *)launchoptions
{
Locationmanager=[[cllocationmanager alloc] init];
[locationmanager  setdelegate:self];
[locationmanager setdistancefilter:kcldistancefilternone];
[locationmanager  setdesiredaccuracy:kcllocationaccuracybest];
//[locationmanager startupdatinglocation];
[worldview  setshowsuserlocation:YES];
// This line may say self.window, don't worry about that
[[self window] makekeyandvisible];
Return YES;
}
- (void)mapview:(mkmapview *)mv didupdateuserlocation:(mkuserlocation*)u
{
Cllocationcoordinate2d loc= [u coordinate];
Mkcoordinateregion  region= mkcoordinateregionmakewithdistance(loc,250,250);
[worldview  setregion:region animated:YES];
}
```

Too big the dot on world map. Want to zoom in. When to send a zoom in message. Instead mkmapview has delegate.

```
#import <Foundation/Foundation.h>
#import <corelocation/corelocation.h>
#import <mapkit/mapkit.h>
@interface  mappoint: nsobject <mkannotation>
{

Nsstring *title;
Cllocationcoordinate2d  coordinate;
}
//A new designated  initializer for instances of mappoint
- (id)initwithcoordinate:(cllocationcoordinate2d)c title:(nsstring *)t;
// This isa required  property from mkannotation
```




```
@property(nonatomic, readonly) cllocationcoordinate2d  coordinate;
// This is an optional  property from mkannotation
@property(nonatomic, copy) nsstring*title;
@end
```

Let's add an annotation. Mkannotation protocol Let's create a new mappoint class.

```
#import "mappoint.h"
@implementation mappoint
@synthesize coordinate, title;
- (id)initwithcoordinate:(cllocationcoordinate2d)c title:(nsstring *)t
{
Self=[super init];
If (self){
Coordinate= c;
[self settitle:t];
}
Return self;
}
- (void)dealloc
{
[title release];
[super dealloc];
}
@end
```

In xib file set text fields delegate to be the instance of whereamiappdelegate. This methods from uitextfielddelegate.

```
@interface  whereamiappdelegate: nsobject
<uiapplicationdelegate, cllocationmanagerdelegate,
Mkmapviewdelegate, uitextfielddelegate> 6
- (BOOL)textfieldshouldreturn:(uitextfield  *)tf{
// This method isn't implemented yet- but will be soon.
```

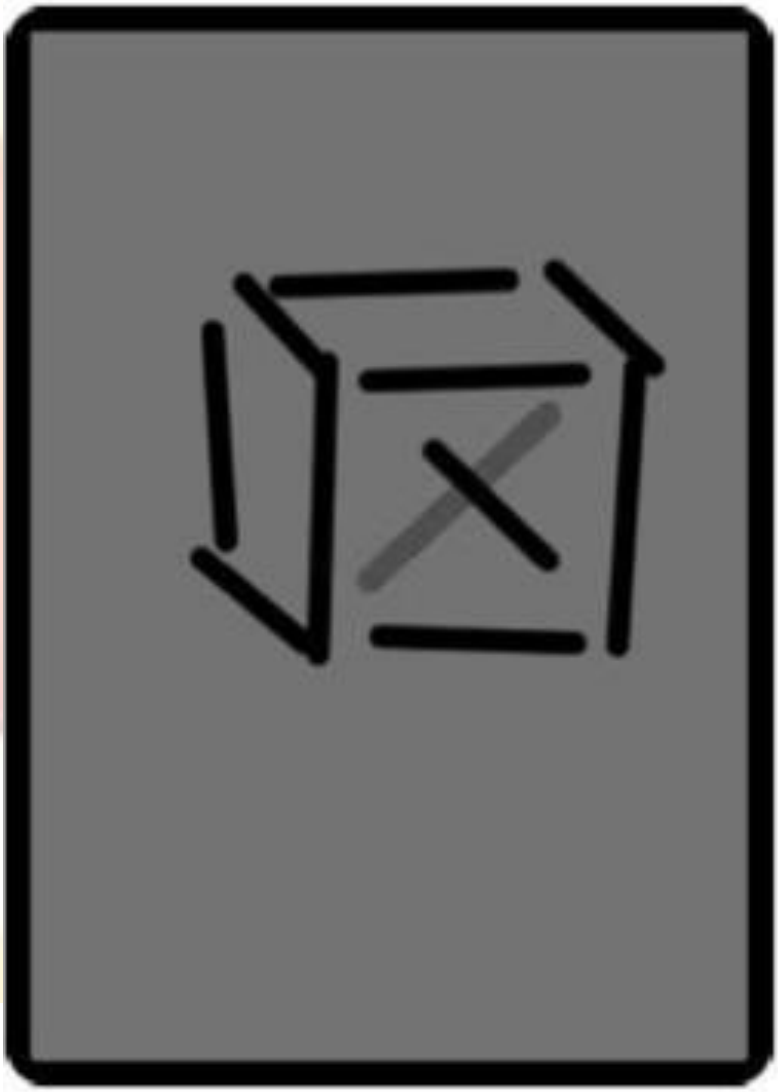
```
[self findlocation];
[tf resignfirstresponder];
Return YES;
} 14
@interface  whereamiappdelegate: nsobject
<uiapplicationdelegate, cllocationmanagerdelegate,
Mkmapviewdelegate, uitextfielddelegate>
{
Cllocationmanager  *locationmanager;
Iboutlet  mkmapview  *worldview;
Iboutlet uiactivityindicatorview *activityindicator;
Iboutlet  uitextfield *locationtitlefield;
}
@property  (nonatomic, retain) iboutlet  uiwindow  *window;
- (void)findlocation;
- (void)foundlocation:(cllocation  *)loc;
@end 28
#import "whereamiappdelegate.h"
#import "mappoint.h"
@implementation  whereamiappdelegate 33
- (void)findlocation
{
[locationmanager startupdatinglocation];
[activityindicator startanimating];
[locationtitlefield  sethidden:YES];
} 40
- (void)foundlocation:(cllocation  *)loc
{
Cllocationcoordinate2d  coord=  [loc coordinate];
// Create an instance of mappoint with the current data
Mappoint  *mp=  [[mappoint alloc] initwithcoordinate:coord
Title:[locationtitlefield text]];
// Add it to the map view
[worldview  addannotation:mp];
// mkmapview  retains its annotations, we can release
[mp release]; 51 // Zoom the region to this location means we can implement
Mkcoordinateregion region= mkcoordinateregionmakewithdistance(coord, 250, 250);
[worldview  setregion:region animated:YES];
[locationtitlefield settext:@""];
[activityindicator stopanimating];
[locationtitlefield sethidden:NO];
[locationmanager  stopupdatinglocation];
} 59
- (void)locationmanager:(cllocationmanager  *)manager
Didupdatetolocation:(cllocation *)newlocation
Fromlocation:(cllocation  *)oldlocation
{
Nslog(@"\%@", newlocation); 65 // How many seconds ago was this new location created?
Nstimeinterval t= [[newlocation timestamp] timeintervalsincenow];
// cllocationmanagers will return the last found location of the
// device first, you don’t want that data in this case.
// If this location was made more than 3 minutes ago, ignore it.
If (t<-180){
// This is cached data, you don’t want it, keep looking
Return;
}
[self foundlocation:newlocation];}
```

Chapter 44

Lecture 44

Let's discuss touch events. Touch events are hallmark of mobile devices Let's make a drawing app like brushes. A finger or fingers touches the screen - (void)touchesbegan:(nsset *)touches withevent:(uievent *)event; a finger or fingers move across the screen (This message is sent repeatedly as a finger moves.) - (void)touchesmoved:(nsset *)touches withevent:(uievent *)event;

A finger or fingers is removed from the screen - (void)touchesended:(nsset *)touches withevent:(uievent *)event; a system event, like an incoming phone call, interrupts a touch before it ends - (void)touchescancelled:(nsset *)touches withevent:(uievent *)event;



Events added to event queue. A uitouch object created and tracked for a finger. Set of uitouches passed. One per finger. Only the moving, begining, or ending event passed. Let's make our app.

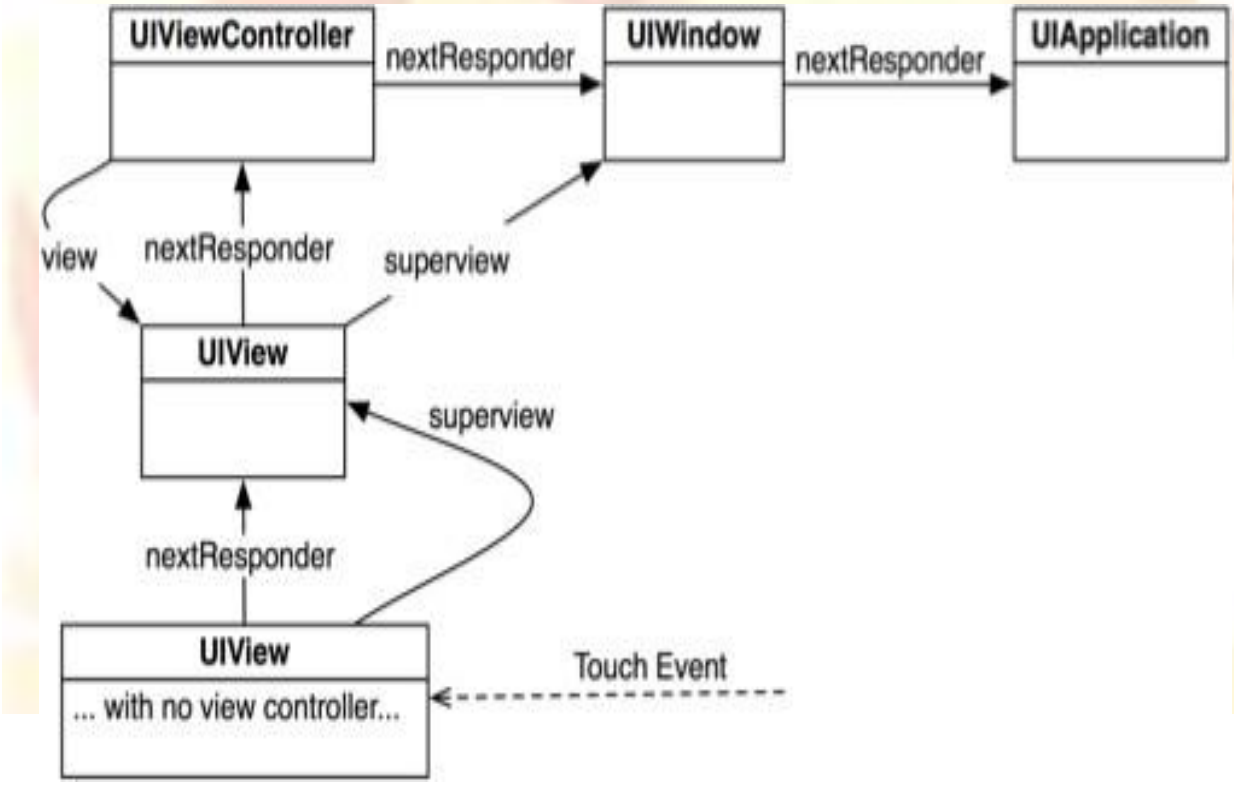
```
#import <Foundation/Foundation.h>
@interface Line: nsobject{
Cgpoint begin;
Cgpoint end;
}
@property(nonatomic) cgpoint begin;
@property(nonatomic) cgpoint end;
@end
#import "Line.h"
@implementation Line
@synthesize begin, end;
@end
#import <Foundation/Foundation.h>
#import <uikit/uikit.h>
@interface  touchdrawview: uiview
{
Nsmutabledictionary*linesinprocess;
Nsmutablearray *completelines;
}
- (void)clearall;
@end
```

Use IB to set the view to default

```
#import "touchdrawview.h"
#import "Line.h"
@implementation  touchdrawview
- (id)initwithcoder:(nscoder*)c
{
Self = [super initwithcoder:c];
If(self){
Linesinprocess=[[nsmutabledictionary alloc] init];
Completelines= [[nsmutablearray   alloc] init];
[self  setmultipletouchenabled:YES];
}
Return self;
}
- (void)dealloc
{
[linesinprocess  release];
[completelines release];
[super dealloc];
}
- (void)clearall
{
// Clear the collections
[linesinprocess  removeallobjects];
[completelines  removeallobjects];
// Redraw
[self setneedsdisplay];
}
- (void)drawrect:(cgrect)rect
{
Cgcontextref context= uigraphicsgetcurrentcontext();
Cgcontextsetlinewidth(context, 10.0);
Cgcontextsetlinecap(context, kcglinecapround);
```

```
// Draw complete lines in black
[[uicolor blackcolor] set];
For(Line *line  in  completelines){
Cgcontextmovetopoint(context,[line begin].x,[line begin].y);
Cgcontextaddlinetopoint(context,[line end].x,[line end].y);
Cgcontextstrokepath(context);
}
// Draw lines in process in red
[[uicolor redcolor] set];
For(nsvalue *v in linesinprocess){
Line *line=[linesinprocess objectforkey:v];
Cgcontextmovetopoint(context,[line begin].x, [line begin].y);
Cgcontextaddlinetopoint(context,[line end].x,[line end].y);
Cgcontextstrokepath(context);
}
}
- (void)touchesbegan:(nsset *)touches
Withevent:(uievent *)event
{
For(uitouch *t in touches){
// Is this a double tap?
If ([t tapcount]> 1){
[self clearall];
Return;
}
// Use the touch object (packed in an nsvalue) as the key
Nsvalue *key= [nsvalue valuewithpointer:t];
// Create a line for the value
Cgpoint loc=[t locationinview:self];
Line *newline=[[Line alloc] init];
[newline  setbegin:loc];
[newline setend:loc];
// Put pair in dictionary
[linesinprocess  setobject:newline forkey:key];
// There isa memory leak in this method
// You will find it using Instruments in the nextchapter
}
}
- (void)touchesmoved:(nsset *)touches
Withevent:(uievent *)event
{
// Update  linesinprocess with moved touches
For(uitouch *t in touches){
Nsvalue *key= [nsvalue  valuewithpointer:t];
// Find the line for this touch
Line *line=[linesinprocess  objectforkey:key];
// Update the line
Cgpoint loc=[t locationinview:self];
[line setend:loc];
}
// Redraw
[self setneedsdisplay];
}
- (void)touchesended:(nsset *)touches
Withevent:(uievent *)event
{
[self endtouches:touches];
```

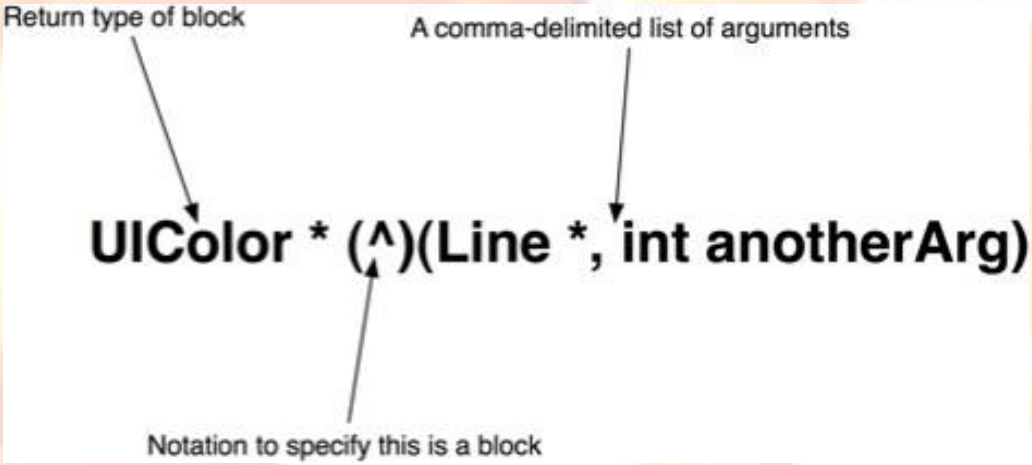
```
}
 - (void)touchescancelled:(nsset*)touches
 Withevent:(uievent *)event
 {
 [self endtouches:touches];
 }
 - (void)endtouches:(nsset *)touches
 {
 // Remove ending touches from dictionary
 For(uitouch *t in touches){
 Nsvalue *key= [nsvalue valuewithpointer:t];
 Line *line=[linesinprocess objectforkey:key];
 // If this is a double tap, 'line' will be nil,
 // so make sure not to add it to the array

 If(line){
 [completelines addobject:line];
 [linesinprocess  removeobjectforkey:key];
 }
 }
 // Redraw
 [self setneedsdisplay];
}
```

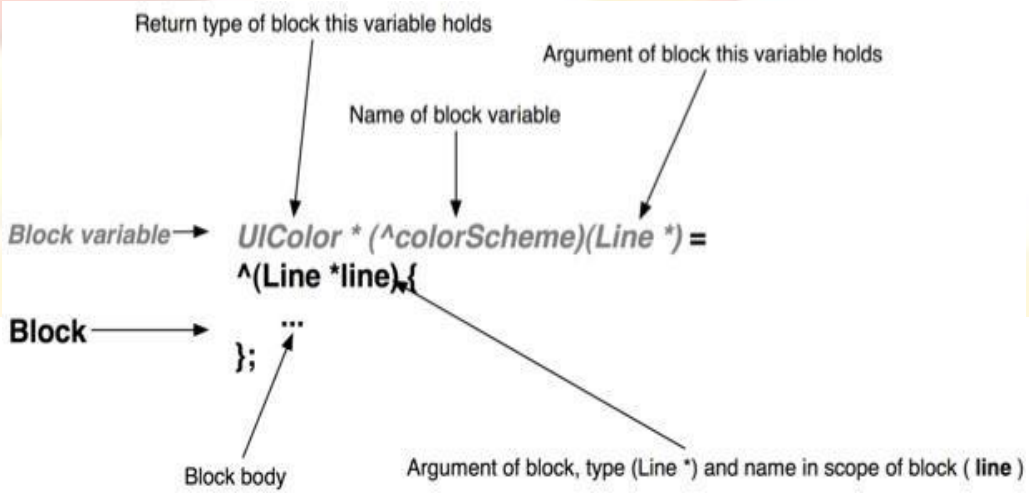


Let's use Blocks in ios 4.0 to change colors based on angle and length.

```
@interface Line: nsobject{
Cgpoint begin;
Cgpoint end;
Uicolor  *color;
}
@property  (nonatomic) cgpoint begin;
@property  (nonatomic) cgpoint end;
@property (nonatomic, retain) uicolor *color;
@end 11 @implementation Line
@synthesize color;
- (id)init
{
Self=  [super init];
If  (self){ 17 [self setcolor:[uicolor  blackcolor]];
}
Return self;
}
- (void)dealloc
{
```

```
[color release];
[super dealloc];
}
- (void)drawrect:(cgrect)rect{
Cgcontextref context= uigraphicsgetcurrentcontext();
Cgcontextsetlinewidth(context, 10.0);
Cgcontextsetlinecap(context, kcglinecapround);
// [[uicolor blackcolor] set];
For(Line *line  in  completelines){
[[line color] set];
Cgcontextmovetopoint(context,[line begin].x,[line begin].y);
Cgcontextaddlinetopoint(context,[line end].x,[line end].y);
Cgcontextstrokepath(context);
}
[[uicolor redcolor] set];
For(nsvalue *v in linesinprocess){
Line *line=[linesinprocess objectforkey:v];
Cgcontextmovetopoint(context,[line begin].x,[line begin].y);
Cgcontextaddlinetopoint(context,[line end].x,[line end].y);
Cgcontextstrokepath(context);
}
}
```





```
- (void)transformlinecolorswithblock:(uicolor* (^)(Line *))colorforline
{
For(Line *l in completelines){
Uicolor *c= colorforline(l);
[l setcolor:c];
}
[self setneedsdisplay];
}

- (void)colorize
{
// Vertical means more red, horizontal means more green,
// longer means more blue
//A block variable named colorscheme is created here:
Uicolor* (^colorscheme)(Line*)=^(Line*l){
// Compute delta between begin and end points
// for each component
Float dx=[l end].x- [l begin].x;
Float dy=[l end].y- [l begin].y;
// If dx is near zero, red= 1, otherwise, use slope
Float r = (fabs(dx)< 0.001?1.0: fabs(dy/ dx));
// If dy is near zero, green=1, otherwise, use inv. Slope
Float g = (fabs(dy)< 0.001?1.0: fabs(dx/ dy));
// blue= length over 300
Float b = hypot(dx, dy)/300.0;
Return[uicolor  colorwithred:r green:g blue:b alpha:1];
};
// Pass this colorscheme block to the method
// that will iterate over every line and assign
// the computed color to that line
[self  transformlinecolorswithblock:colorscheme];
}
```

Colors when you shake.

```
- (BOOL)canbecomefirstresponder
{
Return YES;
}
- (void)didmovetowindow
{
[self becomefirstresponder];
}
- (void)motionbegan:(uieventsubtype)motion
Withevent:(uievent *)event
{
[self colorize];
}
```

Blocks capture variables like anonymous functions in C#. Inline block objects(values copied). Blocks are an alternate to callbacks. Blocks are used for GCD. Kind of like tasks.

Grand central dispatch. Dispatch queues pools of threads managed.

```
- (void) docalculation{
/* Do your calculation here */
}
- (void) calculationthreadentry{
```

```
@autoreleasepool{
Nsuinteger counter= 0;
While ([[nsthread currentthread] iscancelled] == NO){
[self docalculation];
Counter++;
If (counter >= 1000){
Break;
}
}
}
}
- (BOOL)   application:(uiapplication *)application
Didfinishlaunchingwithoptions:(nsdictionary *)launchoptions{
/* Start the thread*/
[nsthread detachnewthreadselector:@selector(calculationthreadentry)
Totarget:self
Withobject:nil];
Self.window = [[uiwindow alloc] initwithframe:
[[uiscreen mainscreen] bounds]];
Self.window.backgroundcolor = [uicolor whitecolor];
[self.window  makekeyandvisible];
Return YES;
}
```

Calling back on UI thread

```
Dispatch_queue_t mainqueue = dispatch_get_main_queue();
Dispatch_async(mainqueue, ^(void) {
[[[uialertview alloc] initwithtitle:@"GCD"
Message:@"GCD is amazing!"
Delegate:nil
Cancelbuttontitle:@"OK"
Otherbuttontitles:nil, nil] show];
});
```

```
Void (^printfrom1to1000)(void) = ^{
Nsuinteger counter= 0;
For(counter = 1;
Counter <= 1000;
Counter++){
Nslog(@"Counter = %lu- Thread = %@",
(unsigned long)counter,
[nsthread  currentthread]);
}
};
Dispatch_queue_t concurrentqueue =
Dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
Dispatch_sync(concurrentqueue, printfrom1to1000);
Dispatch_sync(concurrentqueue, printfrom1to1000);
```

Chapter 45

Lecture 45

Let's download an image asynchronously.

```
Dispatch_queue_t concurrentqueue =
Dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
Dispatch_async(concurrentqueue, ^{
__block uiimage *image= nil;
Dispatch_sync(concurrentqueue, ^{
/* Download the image here*/
});
Dispatch_sync(dispatch_get_main_queue(), ^{
/* Show the image to the user here on the main queue*/
});
});
- (void) viewdidappear:(BOOL)paramanimated{
Dispatch_queue_t concurrentqueue =
Dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
Dispatch_async(concurrentqueue, ^{
__block uiimage *image= nil;
Dispatch_sync(concurrentqueue, ^{
/* Download the image here*/
/* ipad's image from Apple's website. Wrap it into two
Lines as the URL is too long to fit into one line*/
Nsstring *urlasstring = @"http://images.apple.com/mobileme/features"\
"/images/ipad_findyouripad_20100518.jpg";
NSURL *url = [NSURL  urlwithstring:urlasstring];
Nsurlrequest *urlrequest = [nsurlrequest requestwithurl:url];
Nserror *downloaderror = nil;
Nsdata*imagedata = [nsurlconnection
Sendsynchronousrequest:urlrequest
Returningresponse:nil
Error:\&downloaderror];
If (downloaderror == nil\&\&
Imagedata!= nil){
Image = [uiimage  imagewithdata:imagedata];
/* We have the image. We can use it now*/
}
Else if (downloaderror != nil){
Nslog(@"Error happened = \%@", downloaderror);
} else{
Nslog(@"No data could get downloaded from the URL.");
}
});
Dispatch_sync(dispatch_get_main_queue(), ^{
/* Show the image to the user here on the main queue*/
If (image!= nil){
/* Create the image view here*/
Uiimageview *imageview = [[uiimageview alloc]
Initwithframe:self.view.bounds];
/* Set the image*/
[imageview  setimage:image];
/* Make sure the image is not scaled  incorrectly*/
```

```
[imageview  setcontentmode:uiviewcontentmodescaleaspectfit];

/* Add the image to this view controller's view */
[self.view addsubview:imageview];
} else{
Nslog(@"Image isn't downloaded. Nothing to display.");
}
});
});
}
```

Generate 10k random numbers if needed. Read 10K random numbrs, sort n display. Uitableview.

```
- (nsstring*) filelocation{
/* Get the document folder(s)*/
Nsarray *folders =
Nssearchpathfordirectoriesindomains(nsdocumentdirectory
Nsuserdomainmask,
YES);
/* Did we find anything?  */
If ([folders count] == 0){
Return nil;
}
/* Get the first folder  */ 12  nsstring *documentsfolder = [folders  objectatindex:0];

/* Append the file name to the end of the documents path */

Return[documentsfolder
```

```
Stringbyappendingpathcomponent:@"list.txt"];

}

- (BOOL) hasfilealreadybeencreated{

BOOL result = NO;

Nsfilemanager*filemanager = [[nsfilemanager  alloc]  init];

If  ([filemanager  fileexistsatpath:[self  filelocation]]){

Result= YES;

}
Return result;
}

Dispatch_queue_t concurrentqueue =
Dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
/* If we have not already saved an array of 10,000
Random numbers to the disk before, generate these numbers now
And then save them to the disk in an array */
Dispatch_async(concurrentqueue, ^{
Nsuinteger  numberofvaluesrequired = 10000;
If([self hasfilealreadybeencreated] == NO){
Dispatch_sync(concurrentqueue, ^{
Nsmutablearray *arrayofrandomnumbers =
[[nsmutablearray  alloc] initwithcapacity:numberofvaluesrequired]
Nsuinteger counter= 0;
For(counter = 0;
Counter < numberofvaluesrequired;
Counter++){
Unsigned int randomnumber =
Arc4random() % ((unsigned int)RAND_MAX + 1);
[arrayofrandomnumbers  addobject:
[nsnumber  numberwithunsignedint:randomnumber]];
}

/* Now let's write the array to disk */
[arrayofrandomnumbers writetofile:[self filelocation]
Atomically:YES];
});
}
__block  nsmutablearray *randomnumbers = nil;
/* Read the numbers from disk and sort them in an
Ascending fashion */57 dispatch_sync(concurrentqueue, ^{
/* If the file has now been created, we have to read it*/
If([self hasfilealreadybeencreated]){
Randomnumbers= [[nsmutablearray alloc]
Initwithcontentsoffile:[self filelocation]];
/* Now sort the numbers */
[randomnumbers  sortusingcomparator:
^nscomparisonresult(id obj1, id obj2) {
Nsnumber *number1 = (nsnumber*)obj1;
Nsnumber *number2 = (nsnumber*)obj2;
```

```
Return[number1 compare:number2];
}];
}
});
Dispatch_async(dispatch_get_main_queue(), ^{
If([randomnumbers count]> 0){
/* Refresh the UI here using the numbers in the
Randomnumbers array */
}
});
});
```

Dispatch after dispatches after a delay. Timers. Dependencies. Group of tasks.

Course Overview:

Where did we start from. Where to go from here.

- Started with handling multiple input sources in C++, discovered message loop refactored, understood events, event processing, source, target, event object downloaded VS and C# language and its features, examples of OO programming in C#.
- Discussed delegates, events, exception handling, attributes, collections. Worked with XML docs. Wpf history and xaml. Property elements and markup extensions. Mixing xaml and procedural code.
- Discussed logical and visual trees. Dependency properties. Change notifications. Prop val. Inheritance. Attached properties. Sizing, positioning, and transforming elements.
- Discussed transforms, panels stackpanel, wrappanel, canvas, dockpanel, grid. Content overflow, clipping, scaling, scrolling.
- Discussed Events, input events, attached events, touch events (manipulation... High level). Commands, persisting and restoring.
- Covered resources. Binary and logical. Static vs dynamic logical resources. Data binding, binding object, binding markup extension. Binding to collection, implicit datacontext. Datatemplates and value converters. Customizing collection view. Sorting, filtering, grouping, navigating. Data providers. Xml and object data providers.
- Concurrency and threads. Captured variables. Synchronization context and tasks. Continuations. Task completion source. Sync vs async.
- Course grained vs fine grained sync. Async wait keywords in C# 5.0. Parallelism. Cancellation and progress reporting. Task combinator and task parallel library. Parallel.Invoke, For, foreach. Concurrent collections.
- JS history and jquery library. HTML CSS JS. Client side vs server side. Dom. Jquery selectors and filters. Changing attributes and elements. Events and animations. Ajax. Xmlhttprequest, get put load. JSON. Mobile development. Objective C history. Call syntax and OO concepts. Properties. Retain count and memory management. Xib, nib, iboutlet, ibaction, interface builder.
- Button events and a QA app. Protocols and delegates. Location and map kits. Touch events and blocks. GCD and multithreading.