

# Software Engineering-I

---

CS504

Junaid Malik

0304-1659294 | JUNAIDFAZAL08@GMAIL.COM

## CS504 FOR FINAL TERM MCQs and Questions File

### (Important Topics Are Below)

#### Question #1

It ensures that a class only has one instance and provides a global point of access to it.

**Singleton pattern (page # 145)**

Observer pattern

Real pattern

#### Question # 2

Static structural model shows the major system.....

Leaks

**Components (page # 129)**

Activities

Objectives

#### Question # 3

Complex Expressions

Make the code easy to modify

**Make the code difficult to modify (page # 167)**

Make the code easy to understand

Does not effect understandability

#### Question # 4

N-tier architecture stems from the struggle to find a ----- between the fat-client architecture and thin-client architecture.

Concurrency

Distribution point

**Middle ground (page # 134)**

Similarity

#### Question # 5

MVC stands for:

**Model View Controller (page # 143)**

Modern View Centre

Model View Centre

Modern View Controller

#### Question # 6

Class Variables should never be declared public. Public variables violate which of the following:

Information Hiding

Encapsulation

**Information Hiding & Encapsulation (page # 161)**

None

Question # 7 of 10:

----- provides a unified interface to a set of interfaces in a sub-system.

Observer Pattern

Singleton Pattern

**Façade Pattern (page # 146)**

All of the above

Question # 8

"Description of communicating objects and classes that are customized to solve a general design problem in a particular context." is called -----

**Design pattern (page # 140)**

System pattern

System Design

None

Question # 9

In the N-tier architecture, the idea is to enhance scalability and ----- by distributing both the data and the application using multiple server machines.

Usability

**Performance (page # 134)**

Interoperability

None

Question #10

In case of a file servers, client requests selected records from a ..... and the server transmits records to client over the network.

Local memory

Network

Database

**File (page #132 )**

Question No #11

Alpha testing is:

- Testing individual components independent of other components
- Testing a collection of dependent components
- Validation against user expectations
- **Acceptance testing for customized projects, in-house testing for products(page#201)**

**Question No #12**

Code should not be:

- **Commented (page # 130)**
- indented
- cryptic
- aligned

**Question No #13**

When an error is thrown the overall system (on the lookout for this error flag) responds by \_\_\_ the error.

- Ignoring
- Casting
- **Catching (page 187)**
- All of the given

**Question No # 14.**

The complexity of a program may \_\_\_ if there are exceptional paths in it.

- Decrease
- **Increase (page187)**
- Remain the same
- All of given options

**Question NO #15**

Which of the following is used for multi-level commenting?

- **// Comment**
- **/\*Comment\*/ (page 165)**
- (Comment)
- **\*/Comment/\***

**Question No 16**

Testing of collection of modules to discover interfacing problems among interacting modules is called

- Unit testing
- Module testing
- **Subsystem testing (page 201)**
- None

**Question N# 17**

Which of the following shows a commented statement in C++

- **# Ans = first + second**
- **// Ans = first +second**
- **\\ Ans = first + second**

Confirm

- /# Ans = first + second

Question No # 18

STL stands for \_\_\_\_

**Standard Template Library**

Standard Type Link

Standard Tempo Line

None

Question NO #19

There are \_\_\_\_ layers in OSI reference model.

- 5
- 6
- **7** (PAGE 137)
- 8

Question NO# 20

Floating point constants should always be written with decimal point and at least

- **one decimal (page 164)**
- two decimal
- three decimal
- none

Question No # 21

In case of using unrelated operators in a single expression, \_\_\_\_ would be best choice to prevent the logical errors.

- Comments
- Indents
- **Parenthesis** (PAGE 166)
- Short cuts

Question NO # 22

Physical view in Krutchen's 4+1 architectural view model captures \_\_\_\_

Object model of the design

Concurrency and synchronization aspects of the design

**Mapping(s) of the software onto the hardware and reflects its distributed aspect**(page 125)

Static organization of the software in its development environment

Question No # 23

Which of the following is/are among ten things, which the basic template of GOF design pattern includes.

- Problem
- Context
- Forces
- **All of the given (page 141)**

Question No # 24

Data-Centered Architectural Style is also called \_\_\_\_

- Repository model
- Client Server model
- **Sub system model** NOT SURE
- Reference model

Question No # 25

Charles Simonyi first discussed Hungarian Notation. He was of

- **Microsoft ( page 152)**
- IBM
- Dell
- Cisco

Question No# 26

"is" prefix should be used for----- variables and methods. Select correct option:

- General
- **Boolean ( page 155)**
- Constant
- None of the given

Question No 27

In order to make a code more portable, Instead of using vendor specific language extensions, use \_\_\_\_\_ as much as possible

- **STL ( page 182)**
- ANSI
- ISO
- CMMI

Question No 28

This code uses the % operator to set the rear pointer to 0 once it has reached

- Min-Size
- **Max-Size (Page 169)**
- Both
- None

Question No 29:

Vertical partitioning is also known as ....

- Balancing
- Mutating
- Parallelizing
- **Factoring**

Question No 30:

It ensures that a class only has one instance and provides a global point of access to it.

- **Singleton Pattern**
- Observer Pattern
- Real Pattern
- None of the given above

Question No 31:

Fat Client Model is one of the configurations of \_\_\_\_\_ Model.

- Data-centered
- Layered
- Reference
- **ClientServer**

Question No32:

tier architecture stems from the struggle to find a \_\_\_\_\_ between the fat-client architecture and the thin-client architecture.

- Concurrency
- Distribution point
- **Middle ground**

Similarity

Question No 32:

which of these represents the Krutchen's 4+1 architectural view model?

- **Logical view, Process view, Physical view, Development view, Use case view**
- Logical view, Dynamic view, Physical view, Development view, User case view
- Logical view, Process view, Physical view, Development view, Sequence view
- Dynamic view, Process view, Physical view, Development view, Use case view

Question No 33:

Thin Client Model is one of the configuration of \_\_\_\_\_ Model a. Data-centered

- **Client Server**
- Layered

- Reference

Question No 34:

In the N-Tire Architecture the idea is to enhance scalability and \_\_\_\_\_ by distributing both the data and the application using multiple server machines.

- Usability
- Performance
- Interoperability
- Integrity

Question No 35:

Floating point constants should always be written with decimal point and at least

- One decimal
- Two decimal
- Three decimal
- None of the given

Question No 36:

N-tier architecture stems from the struggle to find a \_\_\_\_\_ between the fat-client architecture and the thin-client architecture.

- Concurrency
- Distribution point
- Middleground
- Similarity

Question No 37:

Vertical participation divides the architecture application from a making perspective.

- Decision
- Design
- Conclusion
- Move

Question No 38:

Issues like system performance, availability, scalability, and security are analyzed in

\_\_\_\_\_.

- Logical view
- Physical view
- Code view
- Concurrency view

Question No 39:

Class variables should never be declared public. Public variable violate which of the following:

- Information hiding
- Encapsulation
- Information hiding and encapsulation
- None of the given

Question No 40:

Variable should be initialized where they are \_\_\_\_\_ and they should be declared in the \_\_\_\_\_ scope possible.

- Defined and smallest
- Declared and medium
- Defined and medium
- Declared and smallest

Question No 41:

Patterns are devices that allows program to share knowledge about their \_\_\_\_\_.

- Code
- Design
- Analysis
- Implementation

Question No 42:

Identifier names also play a significant role in enhancing the \_\_\_\_\_ of the program.

Write ability

Read ability

Reliability

All of the given choices

Question No 43:

Go-to statements violates the idea of a. Object oriented code

Structured code

Control structure

Repetition structure

Question No 44:

“Is” prefix should be used for \_\_\_\_\_ variables and methods.

- General
- Boolean
- Constant
- None of the given

Question No 45:

development view in Krutchen’s 4+1 architectural view model captures.

- object model of design
- concurrency and synchronization aspects of the design
- Mapping(s) of the software onto the hardware and reflects its distributed aspect
- Static organization of the software in its development

Question No 46:

A self-documenting code is a code that explain its self without the need of comments  
extraneous documentation like

- flow chart
- UML diagrams
- process flow state diagram
- All of these

Question No 47:  
using proper parenthesis normally makes the code

- easy to read
- easy to understand
- less ambiguous
- all of the given options

Question No 48:  
\_\_\_\_\_ was the first pure Object Oriented language in which observe pattern was used in implementing its model view controller pattern

- Smalltalk
- PASCAL
- JAVA
- C++

Question No 49:  
the form for (;;) should be used for a. nested loop

- empty loop
- more than 1000 iteration
- less than 100 iteration
- None of these

Question No 50:  
Static structural model shows the major systems

- Leaks
- Components
- Activities
- Objectives

Question No 51:  
Global variables in C++ should always be referred to by using the

- ::operator
- operator
- Without an operator
- None of the given

Question No 52:  
\_\_\_\_\_ architecture model is widely used in mainframe application.

- **Client-server Model**
- Repository
- Filter Model
- Layered Model

Question No 53:

The responsibilities of different subsystems so that we get a software system which is easy to maintain, is called .....the architecture.

- Subtracting
- **Partitioning**
- Cloning
- Balancing

Question No 54:

C++ header files should have the extension---- Source files can have the extension --

- .c and .h
- .cc and .h
- **.h and .cpp**
- . All of the given

Question No 55:

- Names representing methods and functions should be----and written in mixed case starting----case.
- Noun----lower
- **Verb----lower**
- Noun----upper
- Verb----upper

Question No 56:

Maintain ability and portability issues are discussed in \_\_\_\_\_.

- Functional view
- Physical view
- **Code view**
- Logical view

Question No 57:

Which of the following is used for multi-level commenting? a. //Comment

- **/\*Comment\*/**
- (Comment)
- \*(Comment)\*
- None of these

Question No 58:

In case of a file servers, Client requests selected records from a \_\_\_\_\_ and the server transmits records to client over the network.

- Local memory
- Network
- Database
- **File**

Question No 59:

In case of header file construction is to avoid \_\_\_\_\_ error. The construction should appear in the top of the file (before the file header).

- **compilation**
- run time
- logical
- all of the given

Question No 60:

Charles Simonyi first discussed Hungarian notation. He was of \_\_\_\_\_

- **Microsoft**
- IBM
- Dell
- Cisco

Question No 61:

Which one is correct?

- **doubletotal=0.5;**
- double total =5:
- double total =50:
- all of the given

Question No 62:

In client server model the application is modeled as a set of \_\_\_\_\_ that are provided by

- servers. Requests
- protocol
- **services**
- Requirements

Question No 63:

\_\_\_\_\_ architectural model is widely is used in mainframe applications.

- **client-server model**
- Repository model
- filter model
- layered model

Question No 64:

client server is a \_\_\_\_ system model a. integrated

- **distributed**
- heterogeneous
- Homogeneous
- None of these

Question No 65:

the form for (;;) should be used for a. nested loop

- **empty loop**
- more than 1000 iteration
- less than 100 iteration

Question No 66:

Client Server model tries to data and processing

**Distribute**

Merge

Clone

Proceed

Question No 67:

Anti- Patterns is another concept that corresponds to common in analysis and design.

Mistake

Issues

**Problems**

All of the given

Question No 68:

vertical partitioning is also known as

- Balancing
- Mutating
- Parallelizing
- **Factoring**

Question No 69

**This client Model is one of the configuration of \_\_\_\_\_ Model. (page 133)**

- **Client Server**
- Thin Client
- Fat Client
- None of these

Question No 70

block of error-sensitive code with exception handling is called

- **Trying to execute a block**

Question No 71

Which function throw for several reasons but all that matters for this function is whether

- **Title ()**

Question No 72

Writing test cases and generating test data are processes that demand \_\_\_\_\_ building

- capabilities.
- Code
- Domain
- System
- **Scenario**

Question No 73

What types of errors are missed by black-box testing and can be uncovered by whitebox testing?

- ❖ Runtime errors
- ❖ **Logic errors**
- ❖ Performance errors
- ❖ Input errors

Question No 74

The cyclomatic complexity metric provides the designer with information regarding the number of \_\_\_\_\_.

- Statements in the program
- Cycles in the program
- Errors in the program
- **Independent logic paths in the program**

Question No 74

The best reason for using Independent software test teams is that

- **Software developers do not need to do any testing**
- Strangers will test the software mercilessly
- Testers do not get involved with the project until testing begins
- The conflicts of interest between developers and testers is reduced

Question No 75

\_\_\_\_\_ is intended to define a many to many relationship between objects so that when

- one object changes state all its dependents are notified and updated automatically.
- **Observer Pattern**
- Facade Pattern
- Singleton Pattern
- Joint Pattern

Question No 76

\_\_\_\_\_ ensures that a class only has one instance and provides a global point of access to it.

- Behavioral Pattern
- Joint Pattern
- **Singleton Pattern (PAGE 145)**
- Observer Pattern

Question No 77

Inspections cannot check \_\_\_\_\_ characteristics.

- **Non-Functional (Page 213)**
- Business
- User

Question No 78

Secondary private classes can be declared as \_\_\_\_\_ and reside in the file of the class

- Asynchronous Classes
- **Inner classes (PAGE158)**
- Outer classes
- Synchronous classes

Question No 78

The construction should appear at the \_\_\_\_\_ of the header file.

- **Top (PAGE 160)**
- Bottom
- Left
- Right

Question No 79

In the switch statement, cases should always end with \_\_\_\_\_.

- Return
- Semi colon
- **Break (Page 170)**
- Full stop

Question No 80

Modularity is a tool that can help us in \_\_\_\_\_ the size of individual functions.

- Stabilizing
- **Reducing (PAGE 173)**
- Increasing
- Strengthening

Question No 81

Abstraction and encapsulation are two important tools that can help in managing and mastering the \_\_\_\_\_ of a program.

- Usability
- **Complexity (Page173)**
- Understandability
- Reliability

Question No 82

One of the guidelines to avoid common mistakes is to never use \_\_\_\_\_ except for declaration.

- **, (PAGE 179)**
- ;
- =
- ||

Question No 83

Which of the following is/are NOT one of the umbrella activities?

- Requirement analysis
- Architecture design
- Test case development
- **All of the given options (Page 14)**

- Project Planning and Management
- Configuration Management
- Software Quality Assurance
- Installation and Training

These are Umbrella Activities

Question No 84

In a ....., each program module is represented by a rectangular box.

- **Use case diagram (GOOGLE)**
- Class diagram
- Document flow diagram
- Data flow diagram

Question No 85

..... is a diagramming technique used to identify the types of objects in the system and the static relationships that exist among them.

- **Class Diagram (GOOGLE)**
- Document flow diagrams
- Data flow diagrams
- Flow charts

Question No 86

Which one is NOT a type of messages which Sequence Diagrams Depict \_\_\_\_\_

- Asynchronous
- Synchronous
- Create
- **Update (Page 111)**

Question No 87

Identifying Whole-Part structures (Aggregations) means, what are my -----

- **Components (PAGE 98)**
- Structures
- Modules
- Interaction Protocols

Question No 88

The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is:

- **Architectural Design (PAGE 118)**
- Interface Design
- Component Design
- Data Design

Question No 89

A complex System evolves from a

- Smaller system
- **Simpler system (GOOG)**
- Bigger system
- Medium system

Question No 90

A context diagram \_\_\_\_\_

- Describes the context dependencies of a system
- **Is a DFD which gives an overview of the system (Conceptual)**
- Is a detailed description of a system
- Is not used in drawing a detailed DFD

Question No 91

When measure of independence of a module or component is low to the other, How would changes in one component have effect on other component?

- No
- Low
- **High (Conceptual)**
- Equal

Question No 92

What is meant by the term 'software crisis'?

- Ability to deliver software on time with good quality
- A situation in which experienced developers leave the company
- **Inability of new software to inter-operate with existing software (Conceptual)**
- A situation in which large scale software projects took more time and budget than was planned

Question No 93

Which is not included in test criteria applied in a phase of testing?

- Functional validity
- Interface integrity
- Correctness
- **Programming Logic (GOOGLE)**

Question No 94

For inspections, ----- are prepared that contain information regarding defects.

- Tables
- **Checklists (PAGE 213)**
- Lists
- Frames

Question No 95

Static analyzers are software tools for ----- processing.

- Analysis text
- **Source text (PAGE 214)**
- Design text
- Maintenance Text

Question No 96

Holistic medicine, concerns itself with the state of the body as a whole, not the ----- that is currently attacking it.

- ❖ Target
- ❖ Reason
- ❖ **Disease (Page 227)**

❖ Source

Question No 97

What factor has no precipitation in more sophisticated and complex computer-based systems?

❖ **Vast use of personal computers. (COCEPTUAL)**

- ❖ Vast increases in computer memory and storage capacity.
- ❖ Greater variety of exotic input/output options.
- ❖ Profound changes in computer architectures.

Question No 98

Software Bugs have multiple names. Which one of the below is not the name of Software Bugs?

- ❖ Bugs
- ❖ Defects
- ❖ Errors

❖ **Mistakes (Page 216)**

Question No 99

Unit testing is roughly equivalent to \_\_\_\_\_ testing for hardware in which each chip is tested thoroughly after manufacturing.

- Circuit level
- **Chip Level (Page 210)**
- Component level
- System level

Question 100

Equivalence \_\_\_\_\_ help you in designing test cases to test the system effectively and efficiently.

- Functions
- **Classes (PAGE 202)**
- Objects
- Interfaces

Question No 101

\_\_\_\_\_ are used to describe flow of data or control in an application.

- Code structures
- String matching.
- Paths
- **Flow graphs (Page 203)**

Question No 102

Writing test cases and generating test data are processes that demand \_\_\_\_\_ building capabilities.

- Domain
- System
- **Scenario (PAGE 198)**
- Code

Question No 103

The raising of the imaginary error flag is simply called raising or \_\_\_\_\_ an error.

- **Throwing (Page 187)**
- Sending

- Casting
- Handling

Question No 104

\_\_\_\_\_ requirements are often called product features.

- Functional
- **Business**
- User
- Non-functional

Question No 105

Many compilers limit the maximum number of bits in the bit field to the size of a(n)

\_\_\_\_\_.

- **Integer (PAGE186)**
- Float
- Character
- Double

Question No 106

In Java, ">>" is used for \_\_\_\_\_ shift and ">>>" for \_\_\_\_\_ shift.

- **Arithmetic, Logical (PAGE 184)**
- Mathematical, Logical
- Incremental, Arithmetic
- Logical, Arithmetic

Question No 107

When large amount of data is to be shared, repository model is used. This model has been extensively used in the \_\_\_\_\_ based application.

- **Mainframe (PAGE 130)**
- Super Computers
- Personal Computers
- Real time

Question No 108

There are four basic coding structures: sequence, if statement, case statement, and

\_\_\_\_\_.

- For loop
- **While loop (PAGE 203)**
- Switch statement
- Logical operations

Question No 109

\_\_\_\_\_ is only a reference for defining protocols and designing and implementing systems developed by different parties.

- OSI model
- **Reference architecture (PAGE 137)**
- Layered architecture
- N-tier architecture

Question No 110

The pattern movement became very quiet until \_\_\_\_\_ when patterns appeared again at OOPSLA conference.

- **1987 (PAGE 141)**
- 1988
- 1962

➤ 1995

Question No 111

Testing activities require destructive instincts in \_\_\_\_\_ for the purpose of breaking system to discover loopholes into its functionality.

- Bug Fixers
- **Tester (PAGE 198)**
- Developer
- Requirement Engineer

Question No 112

Camel Case is now the official convention for file names and identifiers in the \_\_\_\_\_ programming language.

- C#
- C++
- **Java (PAGE152)**
- Visual Basic.Net

Question No 113

In presence of \_\_\_\_\_ bug in a program, the results are the opposite of what is expected.

- Memory Leak
- Memory Over-runs
- Syntax Error
- **Logical Error (Page 221)**

Question No 114

Split lines occur when a statement exceed the \_\_\_\_\_ column limit.

- **80 (158)**
- 90
- 95
- 85

Question No 115

Identifier names play a significant role in enhancing the \_\_\_\_\_ of a program.

- Writ ability
- **Readability (PAGE 151)**
- Reliability
- Usability

Question No 116

Comments should be indented relative to their position in the \_\_\_\_\_.

- **Code (PAGE 165)**
- Design
- Analysis
- Requirements

Question No 117

When planning for performance, one should always remember the \_\_\_\_\_ rule.

- **80/20 (PAGE 180)**
- 70/20
- 60/20
- 100/20

Question No 118

Which of the items listed below is NOT one of the software engineering layers?

- Tools
- Process
- **Manufacturing**
- Methods

Question No 119

Which of the following is a non-functional requirement of a website which sells songs?

- A catalogue of the stock needs to be available for the users to choose from
- Customer information should be retained to allow future transactions easier
- Users should be able to choose from a set of different languages for the interface
- **Time taken to download songs in the catalogue should not irritate users**

**(CONCEPTUAL)**

Question No 120

The state transition diagram \_\_\_\_\_

- Depicts relationships between data objects
- Depicts functions that transform the data flow
- **Indicates how data are transformed by the system (CONCEPTUAL)**
- Indicates system reactions to external events

Question No 121

The order in which bytes of one word are stored is \_\_\_\_\_ dependent.

**hardware** (PAGE 184)

software

language

syntax

Question No 122

Exception handling is a powerful technique that separates error-handling code from \_\_\_\_\_ code.

**Normal** (PAGE 187)

Faulty

Buggy

Complex

Question No 123

Bit fields are a convenient way to express many difficult operations. However, bit fields suffer from one problem

Lack of usability

Lack of security

Lack of performance

**Lack of portability** (PAGE 186)

Question No 124

The complexity of a program may \_\_\_\_\_ if there are exceptional paths in it.

Decrease

**Increase** (PAGE 188)

Remain same

All of given options

Question No 125

When an error is thrown the overall system (on the lookout for this error flag) responds by \_\_\_\_\_ the error.

Ignoring  
Casting

**Catching** (PAGE 187)

All of the given options

Question No 126

Comments are not syntax checked

**TRUE** (PAGE 165)

FALSE

Question No 127

Modularity is a tool that can help us in increasing the size of individual functions, making them less readable.

True

**False** (PAGE 173)

Question No 128

Three tier architecture contains layers

Presentation

Application

Database

All of the above

Question No 129

Vertical partitioning divides the architecture application from a ..... making perspective.

Decision (Page 138)

Design

Conclusion

Move

Question No 130

Modules with high cohesion and low coupling can be treated and analysed as

White boxes

black boxes (PAGE 75)

grey boxes

none of these

Question No 131

While establishing the services for an object, the following fundamental questions should be asked:

- Why does the system need this object anyway?
- **What useful questions can it answer?**
- What useful action can it perform?

- All of the given options.

Question No 132

-----is a role that each actor plays in the system under consideration.

- An act
- A participant (PAGE 96)
- A function
- None of the given

Question NO 133

Any Engineering approach must be founded on organizational commitment to -----.

- Cost
- Scheduling
- Quality Page 115
- Performance

Question No 134

Return values in synchronous messages are:

- Compulsory
- May not used when response is obvious
- Not used at all
- Represented by solid lines

Question No 135

According to Caper Jhones analysis of project activities, coding only has ----- affect part in system development.

- 13-14%
- 36-40%
- 50-60%
- 70-80%

Question No 136

If you are initializing a variable at the time of declaration, do not declare another \_\_\_\_\_ in the same statement.

- Variable 179 (correct)
- Object
- Class

- Module

Question No 137

A change becomes ----- because of close presence of data and functions

- Localized Google
- Private
- Global
- Accessible

Question No 38

Following are some statements associated with data flow diagrams. Identify the correct statement from among them.

- DFDs are used to model complex interfaces.
- DFDs are used to represent only functional processing, data stores and data movements between functions.
- DFDs depict only processes which can be decomposed.
- DFDs do not show external data sources and external data sinks.

Question No 39

A project is considered successful if:

- The system was delivered in time and within budget.
- The system meets at least some of the customer's requirements.
- The system development process has a maximum impact on the ongoing business process.
- Minimum time was spent for requirement gathering and designing.

Question No 40

A process which does not take input is called \_\_\_\_\_

- Miracle process
- Core process
- Secondary process
- Zombie process

Question No 41

Krutchén's 4+1 architectural view model proposes the development of \_\_\_\_\_ main views

- 5
- 4
- 3
- 2

Question No 142

\_\_\_\_\_ are used to describe flow of data or control in an application.

- Code structures
- String matchings
- Paths
- Flow graphs (PAGE 203)

Question No 143

A memory leak bug is one in which memory is somehow allocated from either the operating system or an ----- "pool", but never deallocated when the memory is finished being used

- Mixed memory
- External memory
- Internal Memory Page 219
- Mutually exclusive memory

## Question No 144

Testing is an intellectually demanding activity and has a lifecycle ----- to software development.

- Equal
- Parallel (PAGE 198)
- Sequential
- In contrast

## Question No 145

Software deteriorates rather than wears out because

- Software suffers from exposure to hostile environments
- Defects are more likely to arise after software has been used often
- Multiple change requests introduce errors in component interactions
- Software spare parts become harder to order

## Question No 146

Inspections can check conformance with a specification but not conformance with the --- ----- -requirements.

- Customer's real PAGE 213
- Developer
- Tester
- Manger

## Question No 147

Which is not included in test criteria applied in a phase of testing?

- Functional validity
- Interface integrity
- Correctness
- Programming Logic

## Question No 148

For equivalence partitions, we divide the problem in ----- obvious categories of equal strings

- 4
- 3
- 2 PAGE 202
- 5

## Question No 149

Which of the following is not a fundamental structured programming construct?

- Recursion
- Condition
- Repetition
- Sequence

## Question No 150

Control flow diagrams are:

- Needed to model event driven systems
- Required for all systems
- Used in place of data flow diagrams

- Useful for modeling user interfaces

Question No 151

Bugs that won't "stand still" (almost random) are the ----- to deal with.

- least difficult
- **most difficult (Page 229)**
- very easy
- easy

Question No 152

The goal of testing is to expose ----- defects in a software system before it is put to use.

- **Latent (Page 196)**
- already present
- current
- runtime

Question No 153

----- is a powerful technique that separates error-handling code from normal code.

- **Exception handling (Page 187)**
- Code handling
- variable handling
- pointer handling

Question No 154

The construction is used to avoid ----- errors.

- **compilation (Page 160)**
- runtime
- design time
- none of them

Question No 155

All comments should be written in

- **English (Page 165)**
- French
- C++
- JAVA

Question No 156

Verification and validation are the processes in which we check a product against its ----- and the ---- ----- of the users who will be using it.

- **specifications, expectations (Page 195)**
- statements, expectations
- specifications , arguments
- statements, arguments

Question No 157

Two tests are considered to be equivalent if it is believed that: if one discovers a defect, the other probably will too, and if one does not discover a defect, -----.

- the other probably may
- non of them
- **the other probably won't either (Page 202)**
- the other probably will

## Question No 158

It was lady named ----- who actually coin the term "bug" for the fiest tim

- Elisay Christopher
- **Admiral Grace Hopper (Page 216)**
- Ana Nicholson
- Jane Hopper

## Question No 159

The first "bug" was actually a moth, which flew through an open window and into one of the Mark ----- 's relays.

- **II (Page 216)**
- I
- III
- IV

## Question No 160

Software architecture elements are further divided into categories which in total are -----

- 2
- **3 (Page 122)**
- 4
- 5

Quiz Fall 2021

1. White box testing is performed to test the \_\_\_\_\_ of the program a. Logic

**b. Structure page 198**

- c. Flow
  - d. Syntax
2. Some bit field members are stored:
    - i. left to right
    - ii. right to left
    - iii. in circular array
    - b. only (1) is true
    - c. Only (II) is true
    - d. Only (III) is true

**e. Both (I) and (II) are true page 183**

3. In order to write a portable code which of the following guideline will not be helpful:

- a. Stick to the standards
- b. Program in the mainstream
- c. Size of data types

**d. Using vendor specific language extensions** **page 179**

4. \_\_\_\_\_ is qualitative benefit of Unit Testing

**a. Confidence building** **page 207**

- b. Repeatability
- c. Bounded code
- d. Cost effectiveness

5. In order to make a code more portable. Instead of using vendor specific language extensions, use \_\_\_\_\_ as much as possible

**a. STL** **page 179**

- b. ANSI
- c. ISO
- d. CMM

6. In system development life cycle, defects cannot enter into the program during

\_\_\_\_\_ phase.

- a. Requirement
- b. Design
- c. Coding

**d. Implementation** **page 209**

7. `str = 0; // str is string`

Which rewritten form of above line of code is more in line with the self-documentation philosophy than the code above.

a. str = false;

**b. str = NULL;**

**page 169**

c. str = \0':

d. str = 0.0:

8. \_\_\_\_\_ and \_\_\_\_\_ are important short circuiting logical operators.

a. AND            b. OR            c. NOTd. NOR

**a. a & b            pae 173**

b. b&c

c. c&d

d. a & c.

9. \_\_\_\_\_ testing isolates every part of the program and shows that the individual parts are correct.

a. White Box

b. Black Box

**c. Unit            google**

d. System

10. \_\_\_\_\_ is not a white box testing technique.

a. Statement Coverage

b. Branch Coverage

c. Path Coverage

**d. State Transition Coverage**

**page 202**

11. Chip level testing for hardware is equivalent to \_\_\_\_\_

a. White Box

b. Black Box

**c. Unit testing page 207**

d. System testing

12. For equivalence partitions, we divide the problem in \_\_\_\_\_ obvious categories.

**a. Two page 199**

b. Four

c. Five

d. Six

13. `x = 0; // x is floating pt`

Which rewritten form of above line of code is more in line with the self-documentation philosophy than the code above?

a. `x = false`

b. `x = NULL`

**c. `X = 0.0;` page 169**

d. `x = '0'`

14. One of the main reasons to make functions is \_\_\_\_\_ -

a. Reliability

**b. Reusability page 171**

c. Maintainability

d. Efficiency

15. 80/20 rule states that:

**a. you spend 80 percent of your time in 20 percent of the code page 177**

- b. you spend 20 percent of your time in 80 percent of the code
  - c. We should try to optimized 80 percent or at least 20 percent of the code
  - d. We should try to optimized 20 percent or at least 80 percent of the code
16. \_\_\_\_\_ identifies which statements in a method or class have been executed.
- a. Method Coverage
  - b. Statement Coverage**
  - c. Class Coverage
  - d. Branch Coverage
17. In \_\_\_\_\_ system's internal implementation details are not visible to the tester.
- a. White Box Testing
  - b. Gray Box Testing
  - c. Black Box Testing** **page 198**
  - d. Both White Box Testing & Gray Box Testing
18. Verification and validation are the processes in which we check a product against its \_\_\_\_\_ and the \_\_\_\_\_ of the users who will be using it
- a. statements, expectations
  - b. specifications, expectations** **page 192**
  - c. specifications arguments
  - d. statements, arguments
19. Objective of \_\_\_\_\_ is to show that the program does not work
- a. Requirements
  - b. Design

c. coding

**d. testing** **page 196**

20. The formula for calculating the Cyclomatic Complexity of a program is:

**a.  $E - N + 2$**  **page 208**

b.  $E + N + 2$

c.  $E - N - 2$

d.  $N - E + 2$

21. The raising of the imaginary error flag is simply called raising or \_\_\_\_ an error

a. Catching

b. Casting

**c. Throwing** **page 184**

d. Reprass

22. Be very careful when you use functions with side effects - functions that change the values of the \_\_\_\_\_

a. Objects

b. Classes

c. Structures

**d. Variables**

**Be very careful when you use functions with side effects – functions that change the values of the **parameters**. Page 176**

23. The number \_\_\_\_\_ is the most abused symbol in programs written in C or

C++.

a. 1

**b. 0** page 169

c. 2

d. 3

24. \_\_\_\_\_ is a creative activity while \_\_\_\_\_ is a destructive activity

**a. Development, Testing** page 196

b. Testing, Development

c. Validation, Verification

d. Verification, Validation

25. Which one is used to describe flow of data or control in an application

a. Class diagram

b. Use case diagram

**c. Flow Chart** page 200

d. Code

26. Testing individual components independent of other components is called:

**a. Unit testing** page 189

b. Module testing

c. Subsystem testing

d. System testing

27. The expressions with logical operators can be evaluated only from \_\_\_\_\_

a. Right to left

**b. Left to right** page 173

c. Top to bottom

d. Bottom to top

28. The complexity of a program may \_\_\_\_\_ if there are exceptional paths in it

a. Decrease

**b. Increase** **page 185**

c. Remain same

d. Cutback

29. In the switch statement, cases should always end with a \_\_\_\_\_ statement.

a. Switch

b. Go

**c. Break** **page 167**

d. Stop

30. Cyclomatic complexity measures the logical complexity\_\_\_\_\_

**a. Quantitatively** **page 205**

b. Qualitatively

c. Creditably

d. Worthily

31. The size of \_\_\_\_\_ plays a significant role in making the program

easy or difficult to understand

**a. Function** **page 170**

b. object

c. data type

d. none of the above

32. \_\_\_\_\_ depicts programming instructions that do not have branching

or any control information

**a. Sequence** page 200

- b. IF
- c. While
- d. Case

33. for (i =0, col = 0; i < 27; i++.j++)

In the above line of the code 27 is representing \_\_\_\_\_.

- a. Real Numbers

**b. Magic Numbers** page 167

- c. Constant Numbers
- d. Positive Numbers

34. The process that involves verification of product with respect to its written requirements is called \_\_\_\_\_

- a. Maintenance
- b. Debugging
- c. Development

**d. Testing** page 192

35. If an application fulfills its specifications but deviates from users expectations or their desired behavior. This means, software is verified but not -----

**a. Validated** page 192

- b. Corrected
- c. Checked
- d. Traced

36. A \_\_\_\_\_ variance from a desired product attribute.

- a. Exception
- b. Error
- c. Mistake

**d. Defect** page 192

37. If a function changes the value of some other accessible data object along with returning its value after execution is called \_\_\_\_\_

- a. Short Circuiting
- b. Modularity

**c. Side Effects** page 174

- d. Abstraction

38. if (e.Title() == "CEO" || e.Salary() > 10000)

in the above statement if (e.Title()=="CEO") is TRUE then

**a. Second part is not evaluated** page 185

- b. Second part is always evaluated
- c. Second part must also be TRUE
- d. Second part is not evaluated but e.Salary ()is called

39. When an error is thrown the overall system (on the lookout for this error flag) responds by \_\_\_\_the error

- a. Ignoring
- b. Casting

**c. Catching** page 184

- d. Pass over

40. Testing of collection of modules to discover interfacing problems among interacting modules is called

- a. Unit testing
- b. Module testing
- c. Subsystem testing page 198**
- d. System testing

41. Constant values used in the program are called a. Real Numbers

- b. Magic Numbers page 168**
- c. Constant Numbers
- d. Positive Numbers

42. Bit fields are a convenient way to express many difficult operations. However, bit fields suffer from one problem

- a. Lack of usability
- b. Lack of security
- c. Lack of performance
- d. Lack of portability page 183**

43. Comma ( , ) is very dangerous because \_\_\_\_\_

- a. Compiler does not recognize this symbol
- b. It creates linkage problem
- c. It causes side effects page 176**
- d. It does not causes side effects

44. Function testing falls under \_\_\_\_\_ testing technique

- a. White Box
- b. Black Box

**c. Unit** page 207

d. System

45. The greatest advantage of exception handling is its ability to handle:

**a. Asynchronous errors** page 184

- b. Syntax errors
- c. Memory errors
- d. Control Structure errors

46. After integrating subsystems into a system, and then testing this system as a whole is called

- a. Unit testing
- b. Component testing
- c. Subsystem testing

**d. System**

**testing** 47.

struct

packed\_struct

{

unsigned int f1:1;

} pack;

Here in "packed\_struct":

a. value of f1 = 1

**b. size of f1 = 1 bit**

**page 183**

c. value of f1 should not exceed 1

d. None of given options

48. The C/C++ language has not specified whether \_\_\_\_\_ is arithmetic or logical.

**a. Right shift >> page 181**

b. Right shift <<

c. &&

d. ||

49. A \_\_\_\_\_ is a variance from a desired product attribute

a. Exception

b. Error

c. Mistake

**d. Defect page192**

50. There are \_\_\_\_\_ basic coding structures.

a. One

b. Two

c. Three

**d. Four page 200**

51. In \_\_\_\_\_ control can take either of several branches.

a. IF

b. While

**c. Case statement page 201**

d. Sequence

52. \_\_\_\_\_ cause major portability issues

- a. Loops
- b. Bugs in code

**c. Sizes of data types**      **page 179**

d. Conditional Structures

53. A good program must contain \_\_\_\_\_ infeasible paths.

**a. 0**      **page 206**

- b. 1
- c. N
- d. Infinite

54. We do not use flow graphs to \_\_\_\_\_.

a. Show flow of data

**b. Describe decisions**      **page 200**

- c. Show Control in an application
- d. Describe different coding structures

55. The order in which bytes of one word are stored is \_\_\_\_\_ dependent

**a. Hardware**      **page 181**

- b. software
- c. language
- d. syntax

56. \_\_\_\_\_ is the starting point, base document for both testing and the development

- a. Software code
- b. Test case document
- c. Software architecture document

**d. Functional specification document** **page196**

57. if a major rework is required to translate a program written for one environment to another, it means code is not or less \_\_\_\_\_

- a. Secure
- b. Bug Free
- c. platform dependent

**d. Portable** **page 179**

58. Which of the following is the correct path for the code:

If(a%b==0)     1

C=a+b;         2

Cout<<c;        3

- a. 1-2-3-1
- b. 1-3-2
- c. 1-2-3**
- d. 2-3-1

59. Switch statement is equal to \_\_\_\_\_ statement.

- a. if
- b. if-else
- c. nested if

**d. if-else if** **page 167**

60. Exception handling provides

- a. Portability mechanism
- b. Code Usability mechanism

**c. Error Handling mechanism page 184**

d. Both Portability & Code Usability mechanism 61.

Unit Testing will be done by

- a. Testers
- b. End Users
- c. Customer

**d. Developers page 207**

62. Identify the TRUE statement

- a. Portability effects security

**b. Size of data types vary from one machine to other page 179**

- c. Size of control structures vary from one machine to other
- d. To write portable code, use vendor specific language extensions as much as possible

63. Which type of testing is not concerned with how the inputs are transformed into outputs?

- a. White Box
- b. Gray Box

**c. Black Box page 198**

- d. Both White Box & Gray Box

64. Consider the following statement: `int a,b=10;`

Which of the following is correct: a.  
variable "a" is initialized to 10

**b. Variable "b" is initialized to 10**

- c. Both variables "a" and "b" are initialized to 10
- d. variables can not be initialized this way

65. Software \_\_\_\_\_ is the process of examining the software product against its requirements.

**a. Testing page 192**

- b. Debugging
- c. Exception handling
- d. Fixing

66. The idea behind exception handling is to raise some error flag every time

- a. The code compiles
- b. The code links
- c. Memory is allocated

**d. Something goes wrong page 184**

67. Bit fields allow the packing of data in a structure, using Bit fields we can:

**a. Read 9 bit integers page 182**

- b. Avoid memory leakages
- c. Avoid memory overflow
- d. Avoid syntax errors

# AL-JUNAID INSTITUTE

68. \_\_\_\_\_ is a tool that can help us in reducing the size of individual functions.

a. Inheritance

**b. Modularity page 170**

c. Association

d. Abstraction

69. \_\_\_\_\_ and \_\_\_\_\_ are two important tools that helps in managing the program complexity

a. Composition, Inheritance

**b. Abstraction, Encapsulation page 170**

c. Aggregation, Inheritance

d. Modularity, Composition

70. A path through a program which is never traversed for any input data is called

a. Dependent path

b. Independent path

**c. Infeasible path page 206**

d. Feasible path

71. When a small set of functions (which use each other) is so overwhelmingly the bottleneck, there are two alternatives:

**a. use a better algorithm OR re-write the code page 177**

# AL-JUNAID INSTITUTE

- b. debug the code OR place assertions in code
- c. remove the functions OR add more functions
- d. changed programming language OR compiler at least

72. Alpha testing is:

- a. Testing individual components independent of other components
- b. Testing a collection of dependent components
- c. Validation against user expectations

**d. Acceptance testing for customized projects, in-house testing for**

**products**

**page 198**

73. Exception handling is a powerful technique that separates error-handling code

from \_\_\_\_\_ code.

**a. Normal** **page 184**

- b. Faulty
- c. Buggy
- d. Complex

## Questions

**1. Which Four Technique should be applied in parallel to attain maximum defect removal efficiency?**

# AL-JUNAID INSTITUTE

## Defect origination

In inspections the emphasis is on early detection and fixing of defects from the program. Following are the points in a development life cycle where defects enter into the program.

- Requirements
- Design
- Coding
- User documentation

### 2. What do you understand by term “SELF-Documenting Code”?

In web development, **self-documenting** refers to a website that exposes the entire process of its creation through public **documentation**, and whose public **documentation** is part of the development process.

3. Describe three Covering scheme related to white box testing.

Answer:

- Statement Coverage
- Branch Coverage
- Path Coverage

### 4. What are Inspection pre-Condition?

**Pre-condition** is a statement or set of statements that outline a **condition** that should be true when an action is called. The **precondition** statement indicates what must be true before the function is called.

### 5. Data model?

Data modeling is also an essential activity performed during the design phase. This includes the identification of data entities and their attributes, relationships among these entities, and the appropriate data structures for managing this data.

### 6. Layers?

Answer:

- **The subsystem layer.** Contains a representation of each of the subsystems that enable the software to achieve its customer’s defined requirements and to implement the technical infrastructure that supports customer requirements.
- **The class and object layer.** Contains the class hierarchies that enable the system to be created using generalization and increasingly more targeted specializations. The layer also contains design representations for each object.
- **The message layer.** Contains the details that enable each object to communicate with its collaborators. This layer establishes the external and internal interfaces for the system.

# AL-JUNAID INSTITUTE

- **The responsibility layer.** Contains the data structures and algorithmic design for all attributes and operations for each object.

## 7. Sequences diagram focuses on which two things?

The focus of sequence diagrams is on **objects (and classes)** and **message exchanges among them** to carry out the scenarios functionality. The objects are organized in a horizontal line and the events in a vertical time line.

## 8. How Encapsulation helpful in software engineering?

Hiding the internal details is called encapsulation.

- Encapsulation prevents clients from seeing its inside view.
- Encapsulation hides the details of the implementation of an object.
- Reduced effort in maintenance and enhancement, resulting from encapsulation.
- Intelligent encapsulation localizes design decisions that are likely to change.
- The ability to change the representation of an object without disturbing any of its clients is the essential benefit of encapsulation. Important tool
- that can help in managing and mastering the complexity of a program.

## 9. Types of Structures?

There are two types of structures: Generalization-Specialization (Gen-Spec) and whole-part.

## 10. Identify relationship between?

- |                                 |             |
|---------------------------------|-------------|
| ➤ Father and children           | Inheritance |
| ➤ Ali and house                 | Association |
| ➤ Plants and garden             | Association |
| ➤ Human being and parts of body | Composition |
| ➤ Folk music and music category | Aggregation |

## 11. Define actor and use case?

In a use case model, boundaries of the system are defined by functionality that is handled by the system. Each use case specifies a complete functionality from its initiation by an actor until it has performed the requested functionality. An actor is an entity that has an interest in interacting with the system. An actor can be a human or some other device or system.

## 12. Object data modeling?

Data modeling is also an essential activity performed during the design phase. This

# AL-JUNAID INSTITUTE

includes the identification of data entities and their attributes, relationships among these entities, and the appropriate data structures for managing this data.

## 13. Why is architecture important?

- **Mutual communication:** Software architecture represents a common high-level abstraction of the system that most, if not all, of the systems stakeholders can use as a basis for creating mutual understanding, forming consensus, and communicating with each other.
- **Early design decisions:** Software architecture represents the embodiment of the earliest set of design decisions about a system, and these early bindings carry weight far out of proportion to their individual gravity with respect to the system's remaining development, its service in deployment, and its maintenance life. It is also the earliest point at which the system to be built can be analyzed.
- **Reusable abstraction of a system:** Software architecture embodies a relatively small, intellectually graspable model for how the system is structured and how its components work together; this model is transferable across systems; in particular, it can be applied to other systems exhibiting similar requirements, and can promote large scale reuse.

## 14. Architectural design process names?

- **System structuring** - System structuring is concerned with decomposing the system into interacting sub-systems.
- **Control modeling** - Control modeling establishes a model of the control relationships between the different parts of the system.
- **Modular decomposition** - During this activity, the identified sub-systems are decomposed into modules.

## 15. Architectural Attributes

- **Performance** – Performance can be enhanced by localising operations to minimise subsystem communication. That is, try to have self-contained modules as much as possible so that inter-module communication is minimized.
- **Security** – Security can be improved by using a layered architecture with critical assets put in inner layers.
- **Safety** – Safety-critical components should be isolated
- **Availability** – Availability can be ensured by building redundancy in the system and having redundant components in the architecture.
- **Maintainability** – Maintainability is directly related with simplicity. Therefore, maintainability can be increased by using fine-grain, self-contained components.

## 16. Describe any 2 key points of software design that can be easily amended?

- Requirements design
- GUI design

## 17. Two examples of aggregation from daily life?

- ❖ Car and engine or wheels etc.

# AL-JUNAID INSTITUTE

- ❖ PC and keyboard

## 18. Software design while cost is limited, how to design efficient?

Before developing any software, proper feasibility report is generated while keeping in mind budget and Time limit. Accordingly, project estimations are presented to client. In this scenario, if desired application cannot be prepared in available budget, then obviously, one thing has to be compromised i.e., either budget or 100 percent of required project. So with low budget, some features may be skipped. We have two options:

- ❖ Compromise on efficiency of the software
- ❖ Compromise on the GUI

While we have to design efficient software so have to compromise on GUI. That's a balancing act.

## 19. Template types.

There are three types of Template.

- ❖ Function Template
- ❖ Class Template
- ❖ Variable Template

## 20. Write the step of Test Phasing.

- ❖ **Requirement Analysis.** Your valuable software testers have to view, study, and analyze the available specifications and requirements. ...
- ❖ **Test Planning.** ...
- ❖ **Test Case Designing and Development.** ...
- ❖ **Test Environment Setup.** ...
- ❖ **Test Execution.** ...
- ❖ **Test Closure.**

## 21. Define Cyclomatic complexity and formula.

**Cyclomatic complexity** is a software metric used to indicate the **complexity** of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code. **Cyclomatic complexity** may also be applied to individual functions, modules, methods or classes within a program.

The complexity **M** is then defined

$$M = E - N + 2P,$$

## 22.

# AL-JUNAID INSTITUTE

In c#, properties can contain one or two code blocks called **accessors** and those are called a **get** accessor and **set** accessor.

23. Interface errors:

- **Interface misuse:** a calling component calls another component and makes an error in its use of its interface e.g. parameters in the wrong order.
- **Interface misunderstanding:** a calling component embeds assumptions about the behavior of the called component which are incorrect.
- **Timing errors:** the called and the calling component operate at different speeds and out-of-date information is accessed.

## Types of UNIT test CASE

- The first of these should reflect **normal operation** of a program and should show that the component works as expected.
  - The other kind of test case should be based on testing experience of where common problems arise. It should use **abnormal inputs** to check that these are properly processed and do not crash the component.

## 25. Parts Of Automated test

It has three parts

- A **setup** part, where you initialize the system with the test case, namely the inputs and expected outputs.
- A **call** part, where you call the object or method to be tested.
- An **assertion** part where you compare the result of the call with the expected result. If the assertion evaluates to true, the test has been successful if false, then it has failed.

## 26. Why Code portability is so important? Give out 3 ways / Guide lines to improve the code portability with examples

Many applications need to be ported on to many different platforms. As we have seen, it is pretty hard to write error free, efficient, and maintainable software. So, if a major rework is required to port a program written for one environment to another, it will be probably not come at a low cost. So, we ought to find ways and means by which we can port applications to other platforms with minimum effort. The key to this lies in how we write our program. If we are careful during writing code, we can make it portable. On the other hand, if we write code without portability in mind, we may end-up with a code that is extremely hard to port to other environment.

Following is brief guideline that can help you in writing portable code.

- **Stick to the standard 1.** Use ANSI/ISO standard C++ 2. Instead of using vendor specific language extensions, use STL as much as possible
- **Program in the mainstream** Although C++ standard does not require function prototypes, one should always write them. `double sqrt();` // old style acceptable by ANSI C `double sqrt(double);` // ANSI – the right approach
- **Size of data types** Sizes of data types cause major portability issues as they vary from one machine to the other so one should be careful with them.

# AL-JUNAID INSTITUTE

```
int i, j, k; ... j = 20000; k =
30000; i = j + k; // works if int is
4 bytes // what will happen if int
is 2 bytes?
```

## 27. What is the greatest advantage of exception handling?

One of the most powerful features of exception handling is that an error can be thrown over function boundaries. This allows programmers to put the error handling code in one place, such as the main-function of your program. Exception handling is a powerful technique that separates error-handling code from normal code. It also provides a consistent error handling mechanism. The greatest advantage of exception handling is its ability to handle asynchronous errors.

## 28. Different between flow chart and data flow diagram?

The main difference between DFD and Flowchart is that DFD is a graphical diagram that represents the data flow of a system while flowchart is a graphical diagram that represents the sequence of steps to solve a problem.

## 29. Stages Of TESTING

- **Development testing:** the system is tested during development to discover bugs and defects.
- **Release testing:** a separate testing team test a complete version of the system before it is released to users.
- **User testing:** users or potential users of a system test the system in their own environment.

## IMPORTANT TOPIC

Leacture 23	Architectural Views , Each style describes a system , Citatory that encompasses
Leacture 24	Architectural Style, Repository model characteristics, Clients Server Model, Representative Client /Server Model

# AL-JUNAID INSTITUTE

<b>Lecture 25</b>	<b>partitioning the Architecture, Analyzing Architecture design</b>
<b>Lecture 26</b>	<b>Design patterns defined, Classification of deerns,</b>
<b>lecture 27</b>	<b>singleton pattern facade pattern</b>
<b>Lecture 28</b>	<b>Maintainable code function size</b>
<b>Lecture 29</b>	<b>file handling tips for java and c++, loop structures, Miscellaneous</b>
<b>Lecture 30</b>	<b>comments define Shortcuts And cryptic code</b>
<b>Lecture 31</b>	<b>switch statement</b>
<b>Lecture 32</b>	<b>clarity modularity</b>
<b>Lecture 34</b>	<b>Portability, bit fields</b>
<b>Lecture 35</b>	<b>Exception safety multiple side effects</b>
<b>Lecture 36</b>	<b>software defect Analysis of the code</b>
<b>Lecture 37</b>	<b>Developer and tester different</b>

# AL-JUNAID INSTITUTE

<b>Lecture 38</b>	<b>Equivalence class and Equivalence partitioning string matching paths</b>
<b>Lecture 39</b>	<b>cyclamate complexity</b>
<b>Lecture 40</b>	<b>unit testing principle quantitative benefits qualitative, defect origination</b>
<b>Lecture 41</b>	<b>inspection versus testing inspection per condition static analyzer, checklist for static analysis</b>
<b>Lecture 42</b>	<b>define debugging, art and science of debugging</b>
<b>Lecture 43</b>	<b>bug class, loop error, types of errors, pointer errors, Boolean bugs,</b>

- IF YOU found any Mistake than correct it by yourself

## THANK YOU