CS201 Short Notes Mid term

Lec 1 to22

BY vuonlinehelp.blogspot.com

YouTube link:

Website Link:

https://vuonlinehelp.blogspot.com/

Important Note

You should never copy paste the same answer which we are providing. Please do make some changes in the solution file. If you appreciate our work, please share our website vuonlinehelp.blogspot.com with your friends.

آپ کو کبھی بھی وہی جواب کاپی پیسٹ نہیں کرنا چاہیے جو ہم فراہم کر رہے ہیں۔ براہ کرم حل فائل میں کچھ تبدیلیاں کریں۔ اگر آپ ہمارے کام کی تعریف کرتے ہیں تو براہ کرم ہماری ویب سائٹ vuonlinehelp.blogspot.com کو اپنے دوستوں کے ساتھ شیئر کریں

Lecher number # 01

What is programming?

` Computer programming is a medium for us to communicate with computers. Just like we use “Urdu” or “English” to communicate with each other, programming is a way for us to deliver our instructions to the computer.

What is C programming?

C is a programming language. C is one of the oldest and finest programming language. C was developed by the Dennis Ritchie of AT & T‟S Bell labs USA in 1971.

The BCPL language was developed in 1967 by Martin Richards as a language for writing operating systems software and compilers.

What is C++?

C++ is an update version of C. It is also called C with close. It is an object and oriented language. It was developed by at AT & T‟S Bell lobs

Uses of C and C++

C and C++ language is used to program a wide variety of system. Some of the uses of the C and C ++ are as follows:

- Major parts of windows, Linux and operating system are written in C.
- C and C++ are used to write drive programs for devices like tablets, printers etc.
- C and C++ language is used to program embedded system where properly need to run faster in limited memory (Microwave, Camera etc.).
- C and C++ used to develop fames an area where latency is very important. I.e. Computer has to react quickly on user input.

Variables, Constant, and Keywords Variables:

A variable is a constant which stores ‟ value‟

For example

In kitchen, we have containers storing Rice, Dal, Sugar, etc. Similar to that variable in C stores value of a constant.

Means that

a = 3; // a is assigned “3”

b = 4.7; // b is assigned “4.7”

c = „A“; // c is assigned “A”

Rules for naming variables in C

First character must be an alphabet or underscore (_) No commas, blanks allowed.

No special symbol other than (_) allowed. Variables names are case sensitive.

We must create meaning variables names in our programs. This enhances read easily in our programs constant.

An entity whose value does not change is called as a constant.

A variable is an entity whose value can be changed.

Lecher number # 02

Categories of software

- System Software
- Application Software

System software

System software is a type of computer program that is designed to run a computer's hardware and application programs

Sub categories of system software are:

- Operating system
- Device drivers
- Utilities

What is operating system?

Operating System controls the basic operations of the computer. It prepares the computer to provide an environment in which user communicates with the computer.

What is a Device Driver?

We will see that the operating system has installed special software to control these devices. This piece of software is called device driver software.

TWAIN stands for Technology without an Interesting Name.

What is an Utilites?

Utility software is software designed to help to analyze, configure, optimize or maintain a computer.

Application Software

Application software (app for short) is computing **software** designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users.

What is an Editor?

First of all we need a tool for writing the code of a program. For this purpose we used Editors in which we write our code.

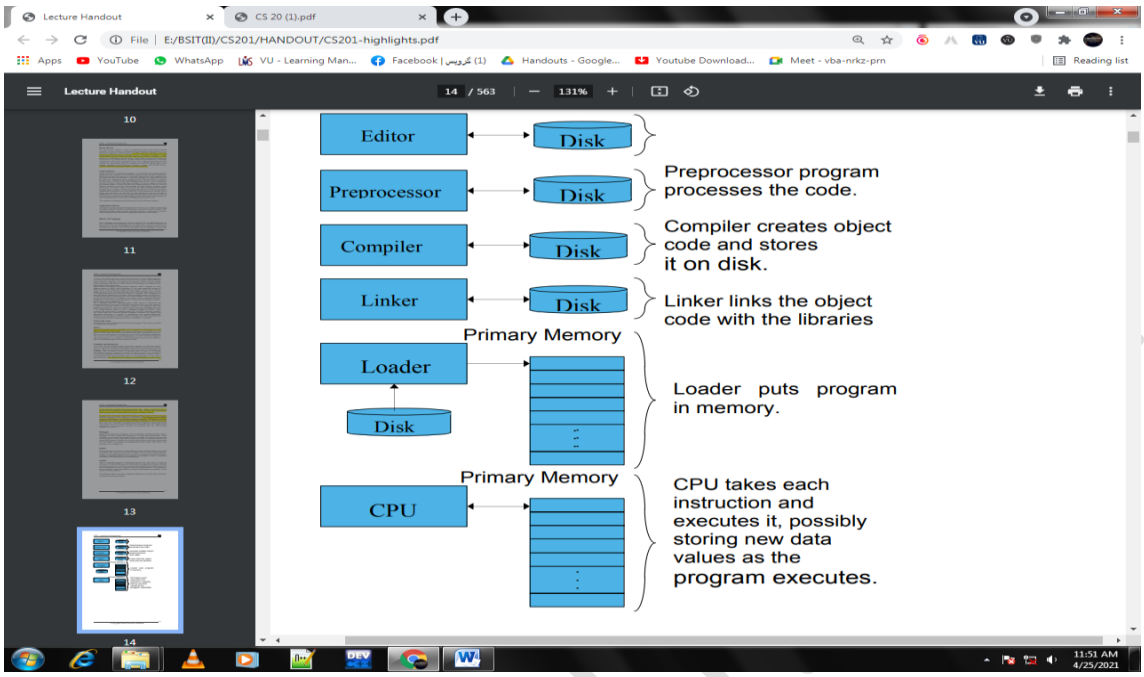
What are Compiler and Interpreter? Or Difference b/w Compiler and Interpreter?

Interpreter The drawback of the interpreter is that the program executes slowly as the interpreter translates the program line by line. Another drawback is that as interpreters are reading the program line by line so they cannot get the overall picture of the program hence cannot optimize the program making it efficient.

Compiler Compilers also translate the English like language (Code written in C) into a language. The Compiler read the whole program and translates it into machine language completely. The difference between interpreter and compiler is that compiler will stop translating if it finds an error.

IDES stand for (Integrated Development Environment)

These IDEs contain editor, compilers, debugger, linker and loader. The benefit of an IDE is that we have all these things at the same place.



Lecture No # 3

So here is our very first program in C.

Program shape

```
1 #include<iostream>
2 using namespace std;
3 int main ()
4 {
5
6
7
8       return 0;
9}
```

#include: This is a pre-processor directive. The sign # is known as HASH and also called SHARP.

Next, there is a curly bracket also called braces ("{ }").

There is a semicolon (;) at the end of the above statement.

What is cout?

This is known as output stream in C and C++. cout takes data from computer and sends it to the output. The sign << indicates the direction of data.

The thing between the double quotes (“”) is known as character string.

What is Variables?

During programming we need to store data.

In a program every variable has

1. Name 2. Type 3. Size 4. Value

What is Assignment Operator?

In C language equal-to-sign (=) is used as assignment operator.
Example

Where as in C language X = 2 (where X is a variable name)

Define Data Types?

A variable must have a data type associated with it, for example it can have data types like integer, decimal number characters etc.

1. Define int Data Type?

The data type int is used to store whole numbers (integers). Space of 4 bytes in memory. In programming before using any variable name we have to declare that variable with its data type. Example

int x ; x = 5; The declaration statement int i ; reserves 4 bytes of memory

Define short Data type/ long Data Type?

Short Data type

The C provides another data type for storing small whole numbers which is called short. The size of short is two bytes and it can store numbers in range of -32768 to 32767.

Long Data Type

A very large whole number that cannot be stored in an int then we use the data type long provided by C. for example Long x = 300500200;

2. Define Float Data Type?

To store real numbers, float data type is used. The float data type uses four bytes to store a real number.

Float x; X=35.36;

3. Define char Data Type?

For storing the character data C language provides char data type. Used around the character as 'a'.

Lecture #4

What is cin?

cin is the counter part of the cout. Here cin is the input stream that gets data form the user and assigns it to the variable on its right side. We know that the sign >>.

Program

```
1. #include<iostream>

2. Using namespace std;

3. int main ()

4 .{      // declaration of variables, the age will be in whole numbers int age1, age2, age3, age4, age5, age6, age7, age8, age9, age10; int Total Age, Average Age;

5.   // take ages of the students from the user

6   .  cout << "Please enter the age of student 1: ";
```

7. cin >> age1;

8. cout << “Please enter the age of student 2: ”;

9. cin >> age2;

10. cout << “Please enter the age of student 3: ”;

11. cin >> age3;

12. cout << “Please enter the age of student 4: ”;

13. cin >> age4;

14. cout << “Please enter the age of student 5: ”;

15. cin >> age5;

16. cout << “Please enter the age of student 6: ”;

17. cin >> age6;

18. cout << “Please enter the age of student 7: ”;

19. cin >> age7;

20. cout << “Please enter the age of student 8: ”;

22. cin >> age8;

23. cout << “Please enter the age of student 9: ”;

24. cin >> age9;

25. cout << “Please enter the age of student 10: ”;

26. cin >> age10;

27. // calculate the total age and average age Total Age = age1 + age2 + age3 + age4 + age5 + age6 + age7 + age8 + age9 +age10;

28. Average Age = Total Age / 10;

29. // Display the result (average age)

30. cout << “Average age of class is: “ << Average Age;

return 0; }

A sample output of the above program is given below.

Please enter the age of student 1: 12

Please enter the age of student 2: 13

Please enter the age of student 3: 11

Please enter the age of student 4: 14

Please enter the age of student 5: 13

Please enter the age of student 6: 15

Please enter the age of student 7: 12

Please enter the age of student 8: 13
Please enter the age of student 10: 11
Average age of class is: 12

age2 = 10;

if (age1 > age2)

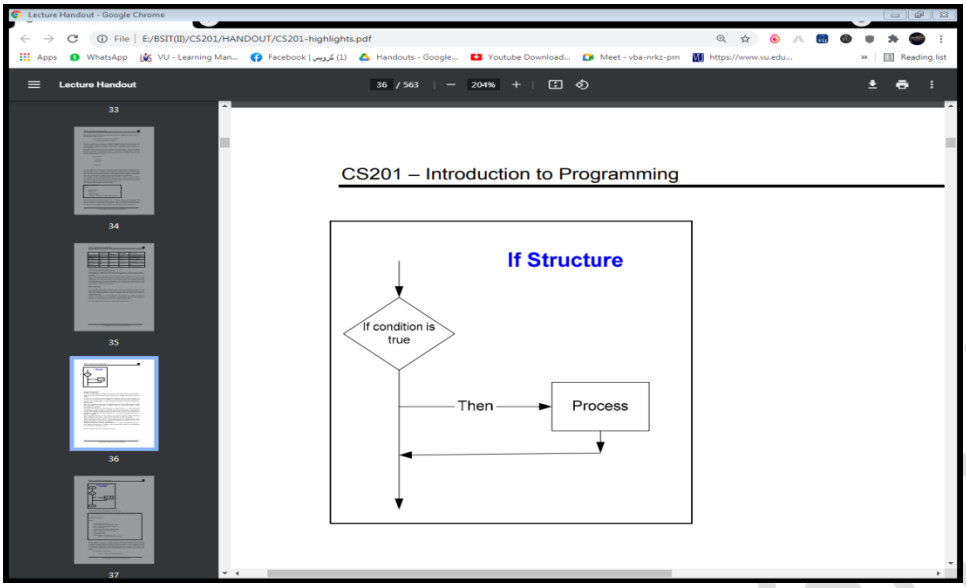
cout<<"Student 1 is older than student 2";

}

What is Flow Charting?

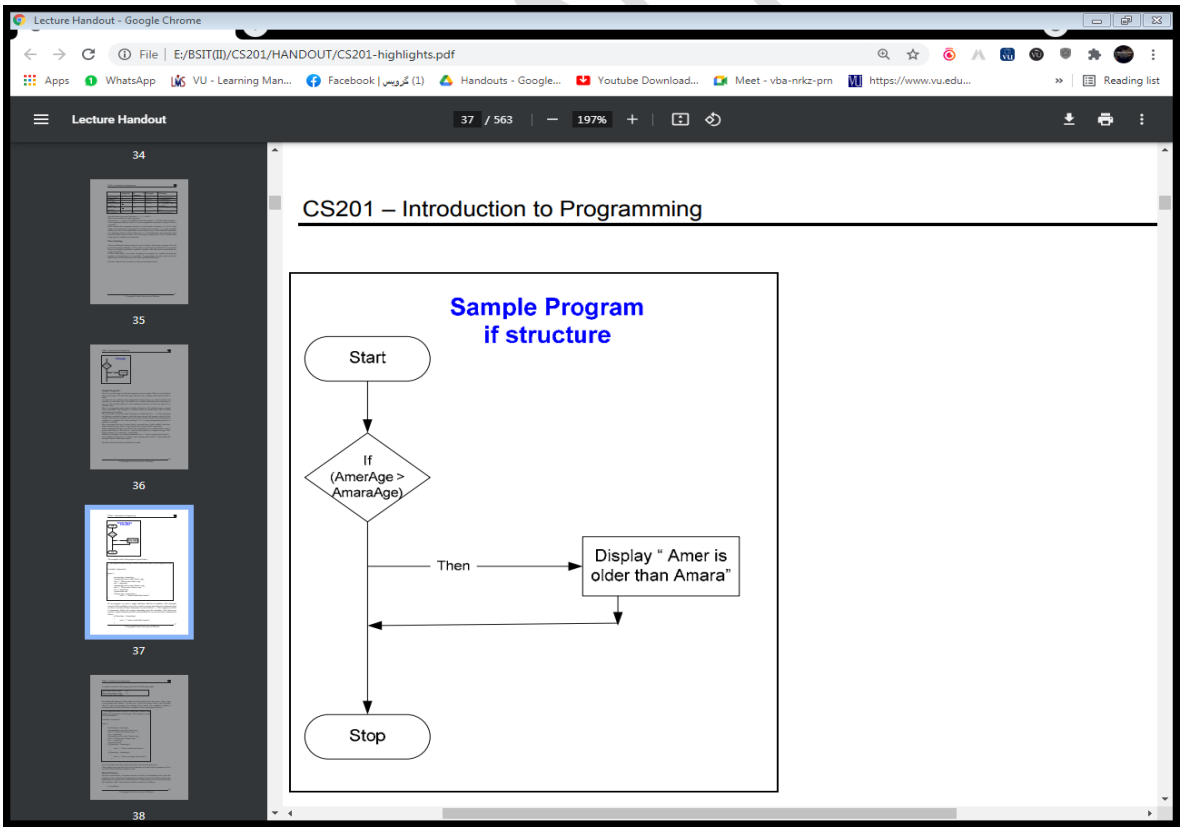
A flow chart helps in correctly designing the program by visually showing the sequence of instructions to be executed.

The flow chart for the if structure is shown in the figure below.



Sample Program if stretcher

There are two students Amar and Amara. We take their ages from the user, compare them and tell who is older?



#include <iostream>

```
using namespace std;
int main()
{
int AmerAge, AmaraAge;
//prompt the user to enter Amer's age
cout<<"Please enter Amer's age";
cin>>AmerAge;
//prompt the user to enter Amara's age
cout<<"Please enter Amara's age";
cin>>AmaraAge;
//preform the test
if (AmerAge > AmaraAge)
cout << "Amer is older than Amara";
}
```

Define Logical Operators?

In programming we use logical operators (&& and ||) for AND and OR respectively with relational operators. These are binary operators and take two operands.

Expression 1	Expression 2	Expression1 && Expression 2	Expression 1 || Expression 2
T	T	F	t
T	F	T	t
F	T	F	F
F	F	f	T

The truth table for the logical negation operator (!) is given below.

Expression	! Expression
True	False
False	True

Lecture No. 6

What is while loop?

In while loop, condition is evaluated first and if it returns true then the statements inside while loop execute, this happens repeatedly until the condition returns false. When condition returns false, the control comes out of loop and jumps to the next statement in the program after while loop.

'While' is also a key word of 'C

While means, 'do it until the condition is true'

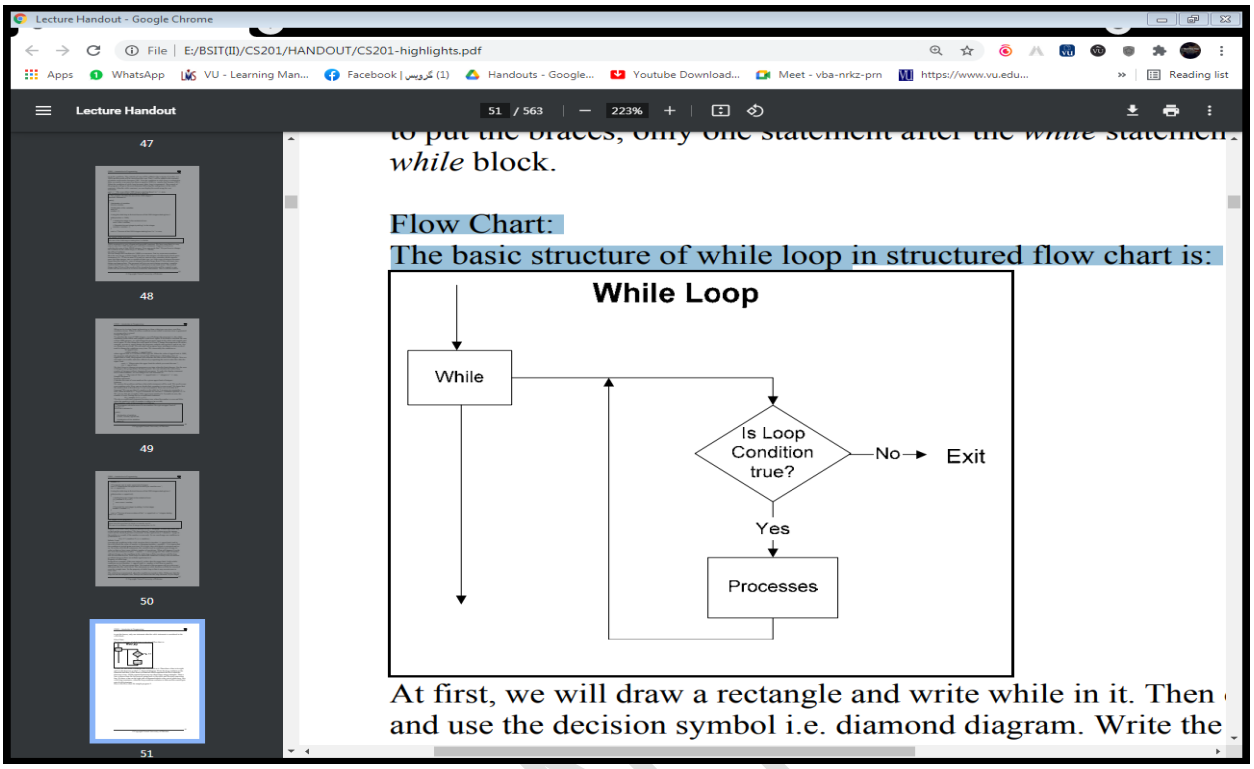
The syntax of while construct is as under

```
While (Logical Expression)
{
statement1;
 statement2;
  ……
 }
```

----------Program----------

```
#include <iostream>
using namespace std;
int main()
{
        int num1;
```

```
        int num2;
        cout<<"plz enter 1st number";
        cin>>num1;
        cout<<"plz enter 2st number";
        cin>>num2;
        if (num1 > num2)
        {
        cout<<"the number is grater";
        }
        else if (num1 < num2)
        {
        cout<<"the number is less";
        }
        else
        {
        cout<<"the numbers are equll";
        }
            return 0;
}
```



Always use the self-explanatory variable names

Practice a lot. Practice makes a man perfect

While loop may execute zero or more time
Make sure that loop test (condition) has an acceptable exit.

Lecture No. 7

What is Do-While Statement?

The **C do while** statement creates a structured loop that executes as long as a specified condition is true at the end of each pass through the loop. The syntax for a **do while** statement is: ... If the value of the expression is "false" (i.e., compares equal to zero) the loop is exited

The syntax of do-while do

```
{
Statement;
}
```

While (condition);

Define For Loop?

A for **loop** is a repetition control structure that allows you to efficiently write a **loop** that needs to execute a specific number of times.

There are three things in a loop structure i.e.

(i) initialization,
(ii) a continuation/termination condition and
(iii) Changing the value of the condition variable, usually the increment of the variable value.

The syntax of for loop is given below.

For (initialization condition; continuation condition; incrementing condition)

{

Statement;

}

Imp

Comments should be meaningful, explaining the task

Don't forget to affect the value of loop variable in while and do-while loops

Make sure that the loop is not an infinite loop

Don't affect the value of loop variable in the body of for loop, the for loop does this by itself in the form statement.

Use pre and post increment/decrement operators cautiously in expressions.

Lecture No. 8

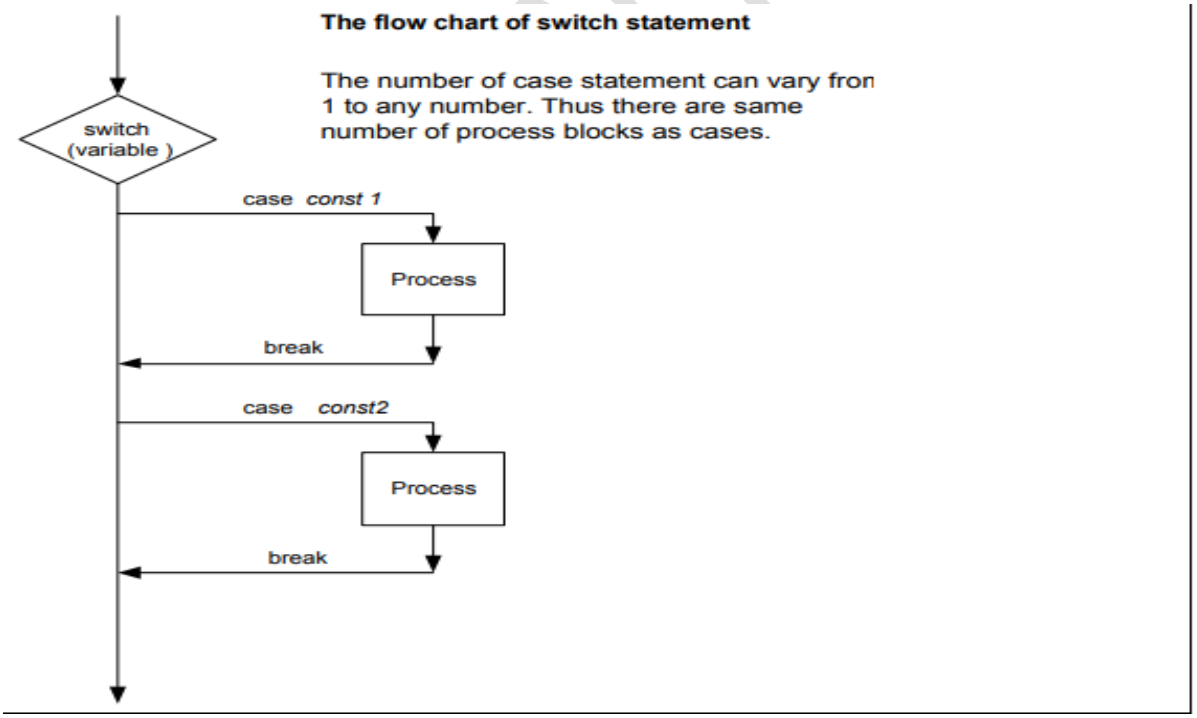
Switch Statement (break ;)

Sometimes, we have multiple conditions and take some action according to each condition.

To avoid this expensiveness, an alternate of multiple if statements can be used that is if/else statements. We can write an if statement in the body of an if statement which is known as nested if. We can write the previous code of if statements in the following nested if/else form.

The syntax of switch statement is as follows.

```
Switch (variable/expression)
{
case constant1 : statementLlist1 ;
case constant2 : statementLlist2 ;
case constantN : statementListN ;
default : statementList ;
}
```



goto Statement

The goto is an unconditional branch of execution

```
#include <iostream>
#include <conio.h>
using namespace std;
int ShowMatrix()
{
int day = 4;
switch (day) {
  case 1:
    cout << "Monday";
    break;
  case 2:
    cout << "Tuesday";
    break;
  case 3:
    cout << "Wednesday";
    break;
  case 4:
    cout << "Thursday";
    break;
  case 5:
    cout << "Friday";
    break;
```

```
  case 6:
    cout << "Saturday";
    break;
  case 7:
    cout << "Sunday";
    break;
}
}
// Outputs "Thursday" (day 4)
```

Lecture No. 9 Introduction back lecher

Introduction

We have a main ( ) in every C program. 'Main ( )' is also a function.

Structure of a Function

{

Return 0

}

return-value_type: (short question or quiz)

The keyword is return which is used to return some value from the function. There may be some functions which do not return any value. For such functions, the return_value_type is void. 'Void' is a keyword of 'C' language. The default return_value_type is of int data type i.e. if we do not mention any return_value_type with a function; it will return an int value.

Calling Mechanism:

For now, let us call the function **swap()** by passing values by reference as in the following example −

```
#include <iostream>
```

```
using namespace std;
// function declaration
void swap(int &x, int &y);
int main () {
  // local variable declaration:
  int a = 100;
  int b = 200;
  cout << "Before swap, value of a :" << a << endl;
  cout << "Before swap, value of b :" << b << endl;
 /* calling a function to swap the values using variable reference.*/
 swap(a, b);
cout << "After swap, value of a :" << a << endl;
 cout << "After swap, value of b :" << b << endl;
 return 0;
}
```

Sample Program

C is called function-oriented language. It is a very small language but there are lots of functions in it. Function can be on a single line, a page or as complex as we want.

Problem statement:

Calculate the integer power of some number (x^n).

```
#include <iostream>


using namespace std;
```

```
int main()
{
   int base, exp, i, result = 1;

 cout << "Enter base and exponent\n";
   cin >> base >> exp;

   // Calculate base^exponent by repetitively multiplying base
   for(i = 0; i < exp; i++){
      result = result * base;
   }

   cout << base << "^" << exp << " = " << result;

   return 0;
}
```

Lecture No. 10

Header files

Header files contain definitions of Functions and Variables, which is imported or used into any C++ program by using the pre-processor #include statement. ... h" which contains C++ function declaration and macro definition. Each header file contains information (or declarations) for a particular group of functions.

Scope of identifiers

Scope of an **Identifier**. Any **identifier** used in a **C++** program (such as the name of a variable or object, the name of a type or class, or the name of a named constant) has a **scope**, i.e., a region of the program in which that name can be used.

Suppose we write the function:

```
#include <iostream>

using namespace std;

const double RATE = 10.50;
int z;
double t;

void one(int x, char y);
void two(int a, int b, char x);
void three(int one, double y, int z);

int main()
{
        int num, first;
        double x, y, z;
        char name, last;
        return 0;
}

Void one (int x, char y)
{
}
```

```
int w;


Void two (int a, int b, char x)
{
}


Void three (int one, double y, int z)
{
        char ch;
        int a;
        {
                int x;
                char a;
        }
}
```

What is function?

It will be risky to tell the address of some variables to the called function. Also, see the code above for some special arrangements for call by reference in C language.

What is Recursive Function?

A **recursive function** is a **function** that calls itself during its execution. The process may repeat several times, outputting the result and the end of each iteration. ... The result could be used as a roundabout way to subtract the number from 10.

```
Long fact (long n)
{
```

```
If (n <= 1)
    return 1;
else
 return n* fact(n-1);
}
```

Lecture No. 11

What is Arrays?

Arrays are a special data-type. If we have a collection of data of same type as in the case.IN C language every array has a data type i.e. name and size. Data type can be any valid data type.

Declaration:

The declaration of arrays is as follows:

data_type array_name [size] ;

For example:

int ages[10];

What is subscript?

In C language, the index of array starts from zero and is one less than array's size. Index of array is also called subscript.

Programing

```
int i, age [10];
for ( i = 0; i < 10 ; i++ )
{
```

```
Age[i]=0;
}
```

With the help of this simple loop, we have initialized all the elements of array age to zero.

Sample program

For example, take an integer array '**n**'.

Solution:

```
 #include <iostream>
int main()
{

        using namespace std;

        int marks[3];
        float average;
        cout << "Enter marks of first student" << endl;
        cin >> marks[0];
        cout << "Enter marks of second student" << endl;
        cin >> marks[1];
        cout << "Enter marks of third student" << endl;
        cin >> marks[2];

        average = ( marks[0] + marks[1] + marks[2] )/ 3.0;
        cout << "Average marks : " << average << endl;
```

```
        return 0;
}
```

Linear Search

linear search (**Searching** algorithm) which is used to find whether a given number is present in an array and if it is present then at what location it occurs. It is also known as **sequential search**.

```
#include<iostream>
using namespace std;


int main() {
cout<<"Enter The Size Of Array:   ";
int size;
cin>>size;
int array[size], key,i;
// Taking Input In Array
 for(int j=0;j<size;j++){
 cout<<"Enter "<<j<<" Element: ";
 cin>>array[j];
 }
//Your Entered Array Is
 for(int a=0;a<size;a++){
    cout<<"array[ "<<a<<" ]  =  ";
    cout<<array[a]<<endl;
 }
```

```
 cout<<"Enter Key To Search  in Array";
 cin>>key;
   for(i=0;i<size;i++){
      if(key==array[i]){
  cout<<"Key Found At Index Number :  "<<i<<endl;
  break;
    }
 }
if(i != size){
cout<<"KEY FOUND at index :  "<<i;
}
else{
cout<<"KEY NOT FOUND in Array  ";
}
   return 0;
}
```

The Keyword ‘const’:

The keyword const can be used with any data type and is written before the data type as:

int age [arraySize];

Now in the loop condition, we can write like this:

For (i = 0; i < arraySize; i++)

Lecture No. 12

Character Arrays

An **array** is an indexed collection of data elements of the same type.
1) **Indexed** means that the array elements are numbered (starting at 0).
2) The restriction of the **same type** is an important one, because arrays are stored in consecutive memory cells. Every cell must be the same type (and therefore, the same size).

That marker is a special character, Called null character. The ASCII code of null character is all zeros.

In C language, we represent the null character as "\0". C uses this character to terminate a string.

While the twelfth is the null character inserted automatically at the end of the string.

Can someone please explain to me why the output from the following code is saying that arrays are **not equal**?

Lecture No. 13

What is array manipulation?

.**Arrays** are sequences of numerical data, with each element having the same underlying data type – integers, real (floating point) numbers, or complex numbers.
These **arrays** can be multi-dimensional, have their dimensionality change dynamically, a can be sliced (subsets of elements taken).

Lecture No. 14 to Lecture No. 22

An *array* is a variable that can store multiple values of the same type.

Declaration:

The declaration of arrays is as follows: data_type array_name [size] ;

For example:

String Arr[5]

String is a datatype

Arr is array name

```
string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
```

```
double grade[27];
```

double is a data type

grade is an array name

and 27 is the size of that array

Access Elements in C++ Array

Each element in an array is associated with a number. The number is known as an array index. We can access elements of an array by using those indices.

```
// syntax to access array elements
array[index];
```

Example

```
string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
cout << cars[0];
// Outputs Volvo
```

Change an Array Element

To change the value of a specific element, refer to the index number:

```
string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
cars[0] = "Opel";
cout << cars[0];
// Now outputs Opel instead of Volvo
```

Take Inputs from User and Store Them in an Array

```
#include <iostream>
using namespace std;

int main() {
```

```
    int numbers[5];

    cout << "Enter 5 numbers: " << endl;

    //  store input from user to array
    for (int i = 0; i < 5; ++i) {
        cin >> numbers[i];
    }

    cout << "The numbers are: ";

    //  print array elements
    for (int n = 0; n < 5; ++n) {
        cout << numbers[n] << "  ";
    }

    return 0;
}
```

2D Array

Two Dimensional array in **C++** is an **array** that consists of more than one rows and more than one column. In **2-D array** each element is refer by two indexes. Elements stored in these **Arrays** in the form of matrice

```
int x[3][4];
```

int is a data type

x isan array name

3 is the number of rows

4 is the number of coloms (Number of elements in one row)

	Col 1	Col 2	Col 3	Col 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Sample Program

```
// C++ Program to display all elements
// of an initialised two dimensional array

#include <iostream>
using namespace std;

int main() {
    int test[3][2] = {{2, -5},
                      {4, 0},
                      {9, 1}};

    // use of nested for loop
    // access rows of the array
    for (int i = 0; i < 3; ++i) {

        // access columns of the array
        for (int j = 0; j < 2; ++j) {
            cout << "test[" << i << "][" << j << "] = " << test[i][j] << endl;
        }
    }

    return 0;
}
```

Arrays Comparison

Two arrays can be equal only when first of all their sizes are equal. Afterwards, we compare the values of the two arrays with one to one correspondence. If all the values in the first array are equal to the corresponding values of the second array, then both the arrays will be equal

```
// This program takes two arrays of 5 integers from user
//displays them and after comparing them displays the result
# include
 main ( )
 {
 int num1 [5], num2 [5], i, equals = 0 ;
 // input of 5 integers of first array
cout << "Please enter five integers for the first array" << endl ;
                for ( i = 0 ; i < 5 ; i ++) cin >> num1 [ i ] ;
                 // input of 5 integers of 2nd array
 cout << "Please enter five integers for the second array" << endl ;
                for ( i = 0 ; i < 5 ; i ++) cin >> num2 [ i ] ;
 //display the elements of two arrays
 cout << "\n The values in the first array are : " ;
                for ( i = 0 ; i < 5 ; i ++)
                cout << "\t" << num1 [ i ] ;
                cout << "\n The values in the second array are : " ;
```

```
        for ( i = 0 ; i < 5 ; i ++) cout << "\t" << num2 [ i ];
    // compare the two arrays
    for ( i = 0 ; i < 5 ; i ++ )
    {
        if ( num1 [ i ] == num2 [ i ] ) { cout << "\n The arrays are equal " ;
        {
         equals = 1 ;
        //set the flag to True
         break ;
        }
     equals = 0;
     //set flag to False
    }
     if (equals) cout << "\n Both arrays are equal" ; }
```

Functions and Arrays

, we can pass arrays as an argument to a function. And, also we can return arrays from a function.

The syntax for passing an array to a function is:

```
returnType functionName(dataType arrayName[arraySize]) {
   // code
}
```

```
int total(int marks[5]) {
   // code
}
```

Int is a return type

Total is a function name

Sec int is a data type of array

Marks ia an array name

[5] is the size of array.

Passing One-dimensional Array to a Function

```
// C++ Program to display marks of 5 students

#include <iostream>
using namespace std;

// declare function to display marks
// take a 1d array as parameter
void display(int m[5]) {
   cout << "Displaying marks: " << endl;

   // display array elements
   for (int i = 0; i < 5; ++i) {
      cout << "Student " << i + 1 << ": " << m[i] << endl;
   }
}

int main() {

   // declare and initialize an array
   int marks[5] = {88, 76, 90, 61, 69};

   // call display function
   // pass array as argument
   display(marks);
```

```
    return 0;
}
```

Same Procedure is for 2D and 3D array

Const in C++:

The **const** keyword specifies that a variable's value is constant and tells the compiler to prevent the programmer from modifying it. ... A pointer to a variable declared as **const** can be assigned only to a pointer that is also declared as **const** .

```
const int myNum = 15;  // myNum will always be 15
myNum = 10;  // error: assignment of read-only variable 'myNum'
```