

**LAB MANUAL**

**(PROGRAMMING FUNDAMENTALS)**

**Submitted By**

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reg#: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prepared By

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CONTENTS**

Introduction-------------------------------------------------------------------------------------------2

Arithmetic Expression --------------------------------------------------------------------- 9

IF/IF-ELSE/NESTED-IF Statements ---------------------------------------------------16

Switch Statements--------------------------------------------------------------------------29

Loops ------------------------------------------------------------------------------------------------33

FOR Loop ----------------------------------------------------------------------------------34

While Loop -------------------------------------------------------------------------------- 37

Do-While Loop --------------------------------------------------------------------------- 38

Arrays -----------------------------------------------------------------------------------------------41

1-D/2-D arrays ---------------------------------------------------------------------------- 42

Bubble Sorting ---------------------------------------------------------------------------- 51

Selection Sorting ------------------------------------------------------------------------- 53

Structure ------------------------------------------------------------------------------------------- 59

Function ------------------------------------------------------------------------------------------- 73

Classes --------------------------------------------------------------------------------------------- 83

File Handling ------------------------------------------------------------------------------------- 94

**INTRODUCTION**

**T**he objective of this lab manual is to give students step-by-step examples to become familiar with programming concepts, design, and coding.

**F E AT U R E S**

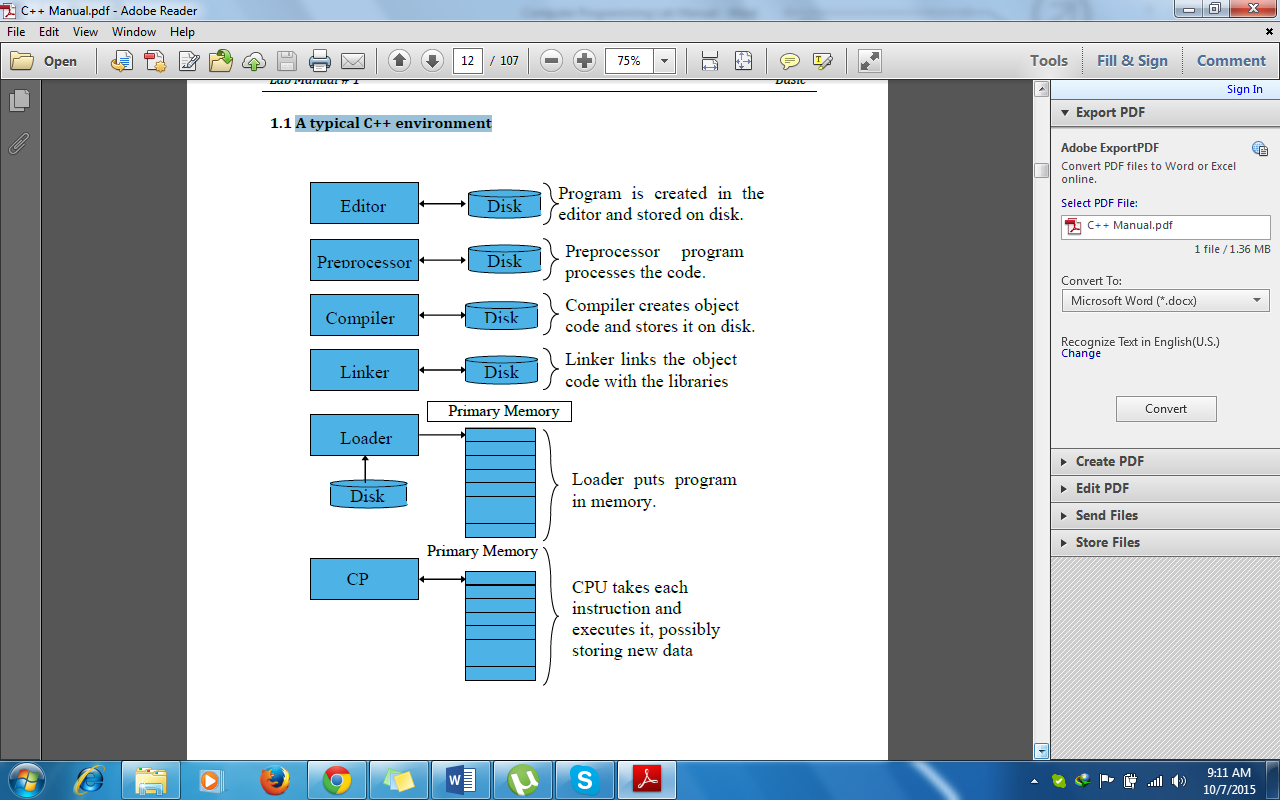
To ensure a successful experience for instructors and students alike, this lab manual includes the following features:

* **Lab Objectives**—Every lab has a brief description and list of learning objectives.
* **Materials required—**Every lab includes information on software you will need to complete the lab.
* **Completion Times**—Every lab has an estimated completion time so that you can plan your activities more accurately.
* **Activity Sections**—Labs are presented in manageable sections; where appropriate, additional Activity Background information is provided to illustrate the importance of a particular activity.
* **Step-by-Step Instructions**—Every lab provides steps to enhance technical proficiency; some labs include Critical Thinking exercises to challenge students.
* **Review Questions**—Some labs include review questions to help reinforce concepts presented in the lab.
* **SOFTWA R E REQUIREMENTS** —Computer running Windows Windows XP, Recommended compiler is Borland C++ 3.0 version, Turbo C++ 3.0 version or Microsoft Visual Studio 2008, 2010 etc.

**COMPLETING THE LAB ASSIGNMENTS**

* Some lab assignments require written answers to complete an exercise, while others are programming assignments that require you to work with a C++ compiler.
* Check with your instructor for instructions on completing the written assignments. For example, you can print pages directly from the appropriate editor, and then write directly on the page.
* To complete the programming assignments, use the compiler that your instructor recommends or requires.
* Print all the documentation assigned, including program code, program prompts, input, and output displayed on the screen, input files, and output files.
* You can submit your written answers and the printed documentation with a lab cover sheet for grading.
* If your instructor requires an electronic copy of your work, e-mail the completed assignment to your instructor or include a removable disk with your work.
* Your instructor will tell you what is needed, but be sure to submit the .cpp.
* To provide program documentation, compile and run your program, copy the prompts, input, and output (if appropriate), and paste them as a block comment at the end of your program.
* Use the Copy and Paste features of your C++ program development kit to do so. After you paste the comment in the program, either print the program file from your text editor or submit the program file to your instructor electronically.

**A typical C++ environment**



**Activity No.1** Date: \_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Write a program to display a message on computer screen “Welcome to University of Education, Township Campus, College Road, Lahore”.**

#include<iostream.h>

#include<conio.h>

void main()

{

cout<< “Welcome to University of Education, Township Campus, College road, Lahore”;

getch();

}

**Output of the Program:**

**Declare Variables**

Declare 3 Integer Type & 3 float type Variables.

|  |  |
| --- | --- |
| **Integer** | **Variable** |
|  |  |
|  |  |
|  |  |

**Activity No.2 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take the radius of a circle from user and calculate the area and circumference of circle and display the results on screen.**

#include<iostream.h>

#include<conio.h>

#define pi 3.14.15

void main()

{

int R;

float A, C;

cout<< “ enter the value of R=”;

cin>> R;

A= pi\*R\*R;

C= 2\*pi\*R;

cout<< “Area of circle =” <<A<<endl;

cout<< “Circumference of circle=”<<C;

getch();

}

**Output of the Program:**

**Activity No.3 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take two numbers from user and display the sum, difference, multiplication, division and remainder of the numbers on computer screen.**

#include<iostream.h>

#include<conio.h>

void main()

{

int x,y, Sum, Diff, Mul, Div, Rem;

cout<< “enter first integer=”;

cin>> x;

cout<< “enter second integer=”;

cin>>y;

Sum=x+y;

Diff=x-y;

Mul=x\*y;

Div=x/y;

Rem=x%y;

cout<< “Sum of integers=” <<Sum<<endl;

cout<< “Difference of integers=” <<Diff<<endl;

cout<< “Multiplication of integers=” <<Mul<<endl;

cout<< “Division of integers=” <<Div<<endl;

cout<< “Remainder of integers=” <<Rem<<endl;

getch();

}

**Output of the Program:**

**Activity No.4 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take a 4-digit number from user and display the number in reverse order on computer screen.**

**Output of the Program:**

**Activity No.5 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take the temperature in Fahrenheit from user and convert it in Centigrade temperature using formula (C=5/9(F-32)) and display the result on computer screen.**

#include<iostream.h>

#include<conio.h>

void main()

{

int F;

float C;

cout<< “enter the Fahrenheit temperature=”;

cin>> F;

cout<< “Centigrade temperature=” <<C;

getch():

}

**Output of the Program:**

**Activity No.6 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take the roll number of the student and amount in dollar from user and display the roll number of the student, amount in dollar and amount in rupees on computer screen.**

#include<iostream.h>

#include<conio.h>

void main()

{  
 int RN, Dollar, Rupees;

cout<< “enter your Roll No=”;

cin>>RN;

cout<< “enter amount in dollars=”;

cin>>Dollar;

Rupees=Dollars\*138;

cout<< “Student Roll No=”<<RN<<endl;

cout<< “ Rupees=”<<Rupees;

getch();

}

**Output of the Program:**

**Activity No.7 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take the student name, discipline, year of admission and registration from user and display on the computer screen.**

**Output of the Program:**

**Arithmetic Operators**

Define Arithmetic Operators

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

**Precedence**

Write precedence of the arithmetic operators

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**ARITHMETIC EXPRESSION**

* + An arithmetic expression is a combination of variables, constants and arithmetic operators.
  + It is used to calculate the value of an arithmetic formula.
  + It returns a single value after evaluating the expression.
  + The value of the expression is assigned to a variable on the left side of ‘=‘ operator. This variable is known as the receiving variable.
  + The operator ‘=‘ is called assignment operator. It is used to assign the calculated value of the expression to the receiving variable.
  + The receiving variable must be of numeric type.

e.g.

if m=10, x=5 the expression may be written as

m\*x+100 and is called an arithmetic expression.

**Order of Precedence of Operation**

* The order in which the arithmetic expression is evaluated is called the order of precedence.
* It is also known as hierarchy of operations.
* When an arithmetic expression is evaluated, the computer performs only one operation at a time.
* All multiplications and divisions are performed first from left to right.
* All additions and subtractions are then performed from left to right.
* If parentheses are used in an expression, the expressions within parentheses are first computed from left to right.
* When parentheses are used within parentheses, the expression within innermost parentheses is evaluated first.

(4-(3\*5))+2

**IF/IF-ELSE/NESTED-IF/SWITCH STATEMENTS**

**The “if” Statement**

* + The “if” statement is used to execute a set of statements after testing a condition.
  + The “if” statement evaluates a condition. If the condition is true, the statement “following the” if statement is executed, otherwise if the condition is false, the “if statement” is ignored and the control transfers to the next statement.

if(condition)

{ statement -1;

statement-2;

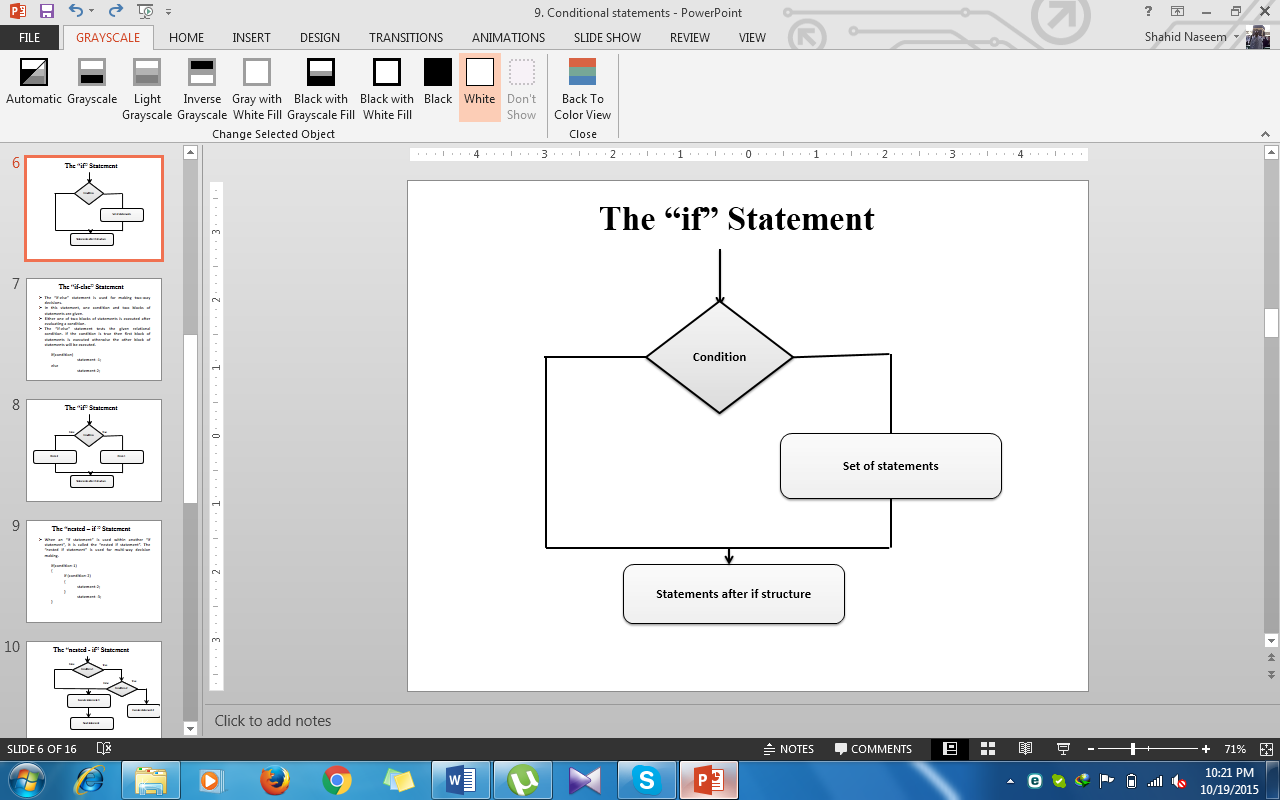
-----------------

-----------------

statement –n;

}

end if



**Activity No.8 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take two numbers from user and display the greater number using IF-Statement.**

#include<iostream.h>

#include<conio.h>

void main()

{

int A,B;

cout<< “enter first number=”:

cin>>A;

cout<<”enter second number=”

cin>>B;

if(A>B)

cout<< “First number is greater”;

if(B>A)

cout<<” Second number is greater”;

getch();

}

**Output of the program**

**Activity No.9 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take previous and current month reading from user and calculate the consumed units in a month. If the consumed units are less and equal to 300, then calculate the “electricity bill” by taking Rs. 13/- per unit rate. On the other hand, if the consumed units are more than 300, then calculate the “electricity bill” by taking Rs.18/- per unit rate as well as 5% surcharge in the bill using IF-Statement.**

**Output of the program**

**Activity No.10 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

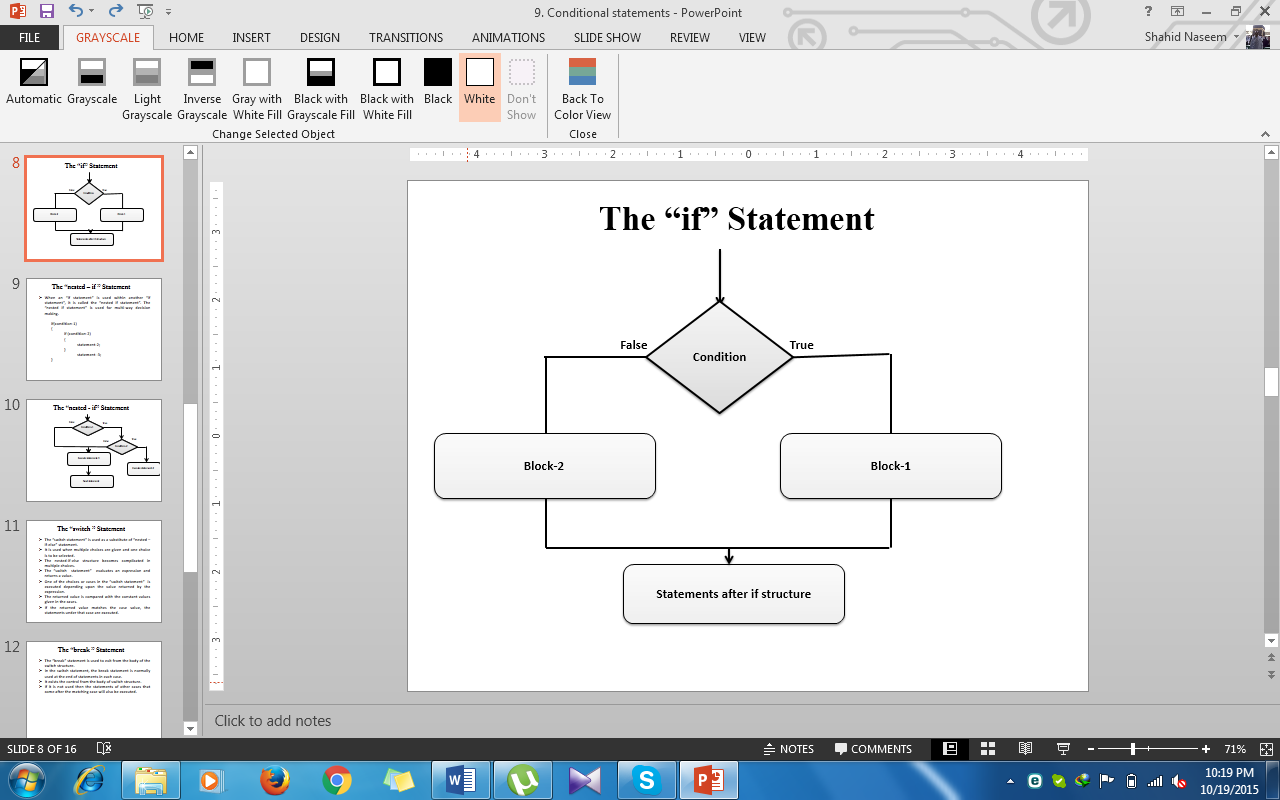
**Write a program that take the employee name and basic salary from user. If the basic salary is less than or equal to 5000, then take house rent (30% of basic salary), contract allowance (45% of basic salary), conveyance allowance (45% of basic salary), medical allowance Rs.193/. On the other hand, if the basic salary is more than 5000, then take all the parameters same except taking income tax (2% of basic salary), medical allowance Rs. 96/- and calculate net salary by adding basic salary, house rent, contract allowance, conveyance allowance, medical allowance and subtracting income tax using IF-Statement.**

**The “if-else” Statement**

* + The “if-else” statement is used for making two-way decisions.
  + In this statement, one condition and two blocks of statements are given.
  + Either one of two blocks of statements is executed after evaluating a condition.
  + The “if-else” statement tests the given relational condition. If the condition is true then first block of statements is executed otherwise the other block of statements will be executed.

if(condition)

statement -1;

else statement-2;

**Activity No.11 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to input a number from the user. Use if-else statement to find out whether the number is less than or greater than 100.**

#include<iostream.h>

#include<conio.h>

void main ()

{

int n;

cout<< “enter an integer value”;

cin>>n;

if (n>100)

cout<< “number is greater than 100”

else

cout<< “number is less than 100”;

getch();

}

**Output of the program**

**Activity No.12 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to calculate the net pay of an employee. Input the basic pay and calculate the net pay as follows using IF-Else Statement:**

* **House rent is 45% of the basic pay.**
* **Medical allowance is 2% of basic if basic is greater than Rs. 5000/-. It is 5% of basic pay if the pay is less than Rs. 5000/-.**
* **Conveyance allowance is Rs. 96/- if basic pay is less than Rs. 5000/-. It is Rs. 193/- if the basic pay is more than Rs. 5000/-.**
* **Net pay is calculated by adding basic pay, medical allowance, conveyance allowance and house rent.**

**Output of the program**

**Activity No.13 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take input marks obtained by a student in a subject. The marks of each subject is 100. Find out grade of the student by using IF-Else statement.**

**grade >= 85 Grade A+**

**grade >= 80 Grade A**

**grade >=75 Grade B+**

**grade >=70 Grade B**

**grade >=65 Grade C+**

**grade >=60 Grade C**

**grade >=55 Grade D+**

**grade >=50 Grade D**

**grade < 50 Grade F**

#include<iostream.h>

#include<conio.h>

void main()

{

int marks;

cout<< “enter obtaining marks=”;

cin>>marks;

if(marks>=85)

cout<< “ grade is A+”;

else if(marks>=80)

cout<< “grade is A”;

else if(marks>=75)

cout<< “grade is B+”;

elseif (marks>=70)

cout<< “grade is B”;

elseif (marks>=65)

cout<< “grade is C+”;

elseif (marks>=60)

cout<< “grade is C”;

elseif (marks>=55)

cout<< “grade is D+”;

elseif (marks>=50)

cout<< “grade is D”;

else

cout<< “grade is F”;

getch();

}

**Output of the program**

**Activity No.14 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take a number from user and display the number is divisible by 3 or not using IF-ELSE statement.**

#include<iostream.h>

#include<conio.h>

void main()

{

int n;

cout<< “enter a number=”;

cin>>n;

if(n%3=0)

cout<< “ number is divisible by 3”;

else

cout<< “number is not divisible by 3”;

getch();

}

**Output of the program**

**Activity No.15 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take a number from user and display the number is even or odd using IF-ELSE statement.**

**Output of the program**

**The “nested – if ” Statement**

* + When an “if statement” is used within another “if statement”, it is called the “nested if statement”. The “nested if statement” is used for multi-way decision making.

if(condition-1)

{

if (condition-2)

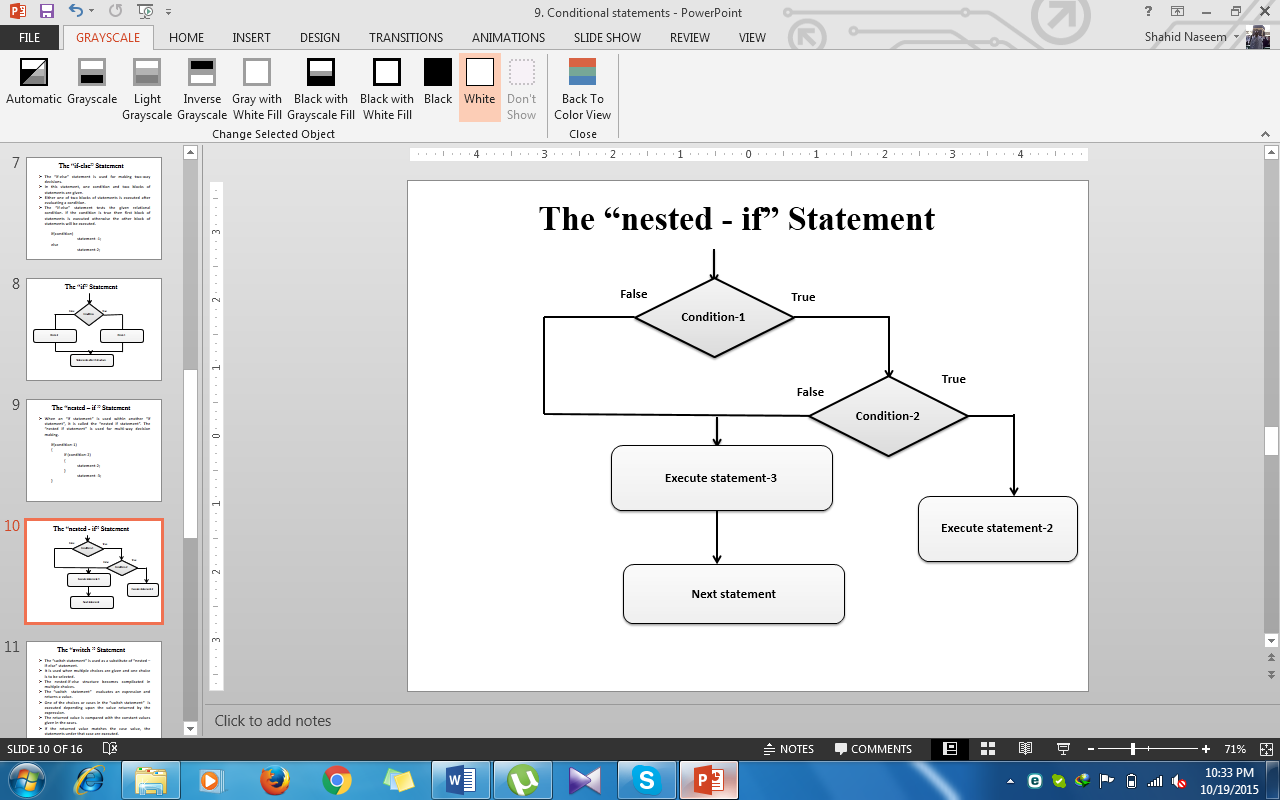
{

statement-2;

}

statement -3;

}



**Activity No.16 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to compare the three numbers and display the greater number using nested-if statement and display the result in the below box.**

**Output of the program**

**Activity No.17 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Write a program that take three numbers from user and display the numbers are equal or not using NESTED-IF statement.**

**Output of the program**

**SWITCH STATEMENT**

**The “Switch” Statement**

* + The “switch statement” is used as a substitute of “nested –if-else” statement.
  + It is used when multiple choices are given and one choice is to be selected.
  + The nested-if-else structure becomes complicated in multiple choices.
  + The “switch statement” evaluates an expression and returns a value.
  + One of the choices or cases in the “switch statement” is executed depending upon the value returned by the expression.
  + The returned value is compared with the constant values given in the cases.

If the returned value matches the case value, the statements under that case are executed.

**The “break ” Statement**

* + The “break” statement is used to exit from the body of the switch structure.
  + In the switch statement, the break statement is normally used at the end of statements in each case.
  + It exists the control from the body of switch structure.
  + If it is not used then the statements of other cases that come after the matching case will also be executed.

**Activity No.18 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that a character from user and display whether it is vowel or not on the screen.**

**Output of the program**

**Activity No.19 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take two numbers and an operator from user and display the sum, difference, multiplication, division and remainder using switch statement.**

**Output of the program**

**FOR/WHILE/DO-WHILE LOOPS**

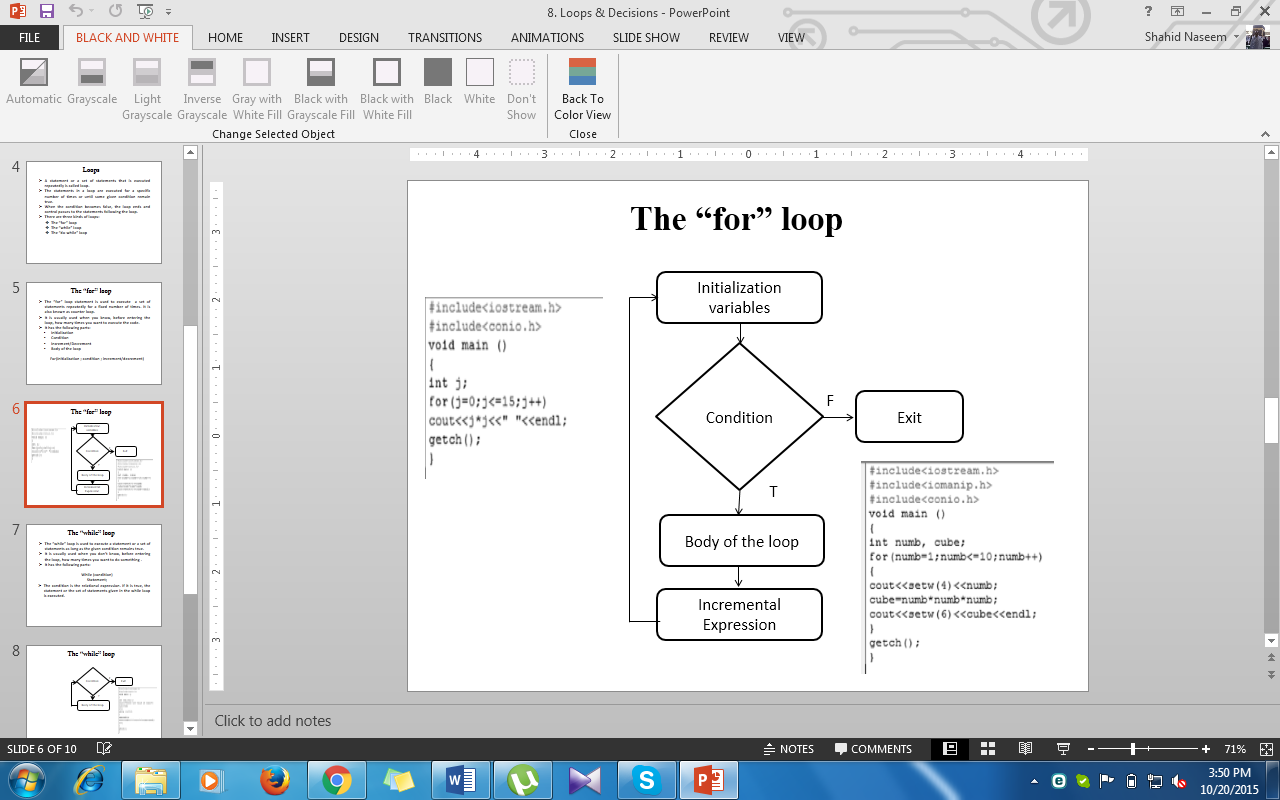
**Loops**

* + A statement or a set of statements that is executed repeatedly is called loop.
  + The statements in a loop are executed for a specific number of times or until some given condition remain true.
  + When the condition becomes false, the loop ends and control passes to the statements following the loop.
  + There are three kinds of loops:
    - The “for” loop
    - The “while” loop
    - The “do-while” loop

**The “for” loop**

* + The “for” loop statement is used to execute a set of statements repeatedly for a fixed number of times. It is also known as counter loop.
  + It is usually used when you know, before entering the loop, how many times you want to execute the code.
  + It has the following parts:
  + Initialization
    - Condition
    - Increment/Decrement
    - Body of the loop

For(initialization ; condition ; increment/decrement)



**Activity No.20 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to print a table of a given number using “for” loop.**

**Output of the Program**

**Activity No.21 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the odd numbers from 1 to 10 natural numbers and also display the sum of these numbers using FOR loop.**

**Output of the Program**

**Activity No.22 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the odd numbers from 1 to 10 whole numbers and also display the sum of these numbers using FOR loop.**

**Output of the Program**

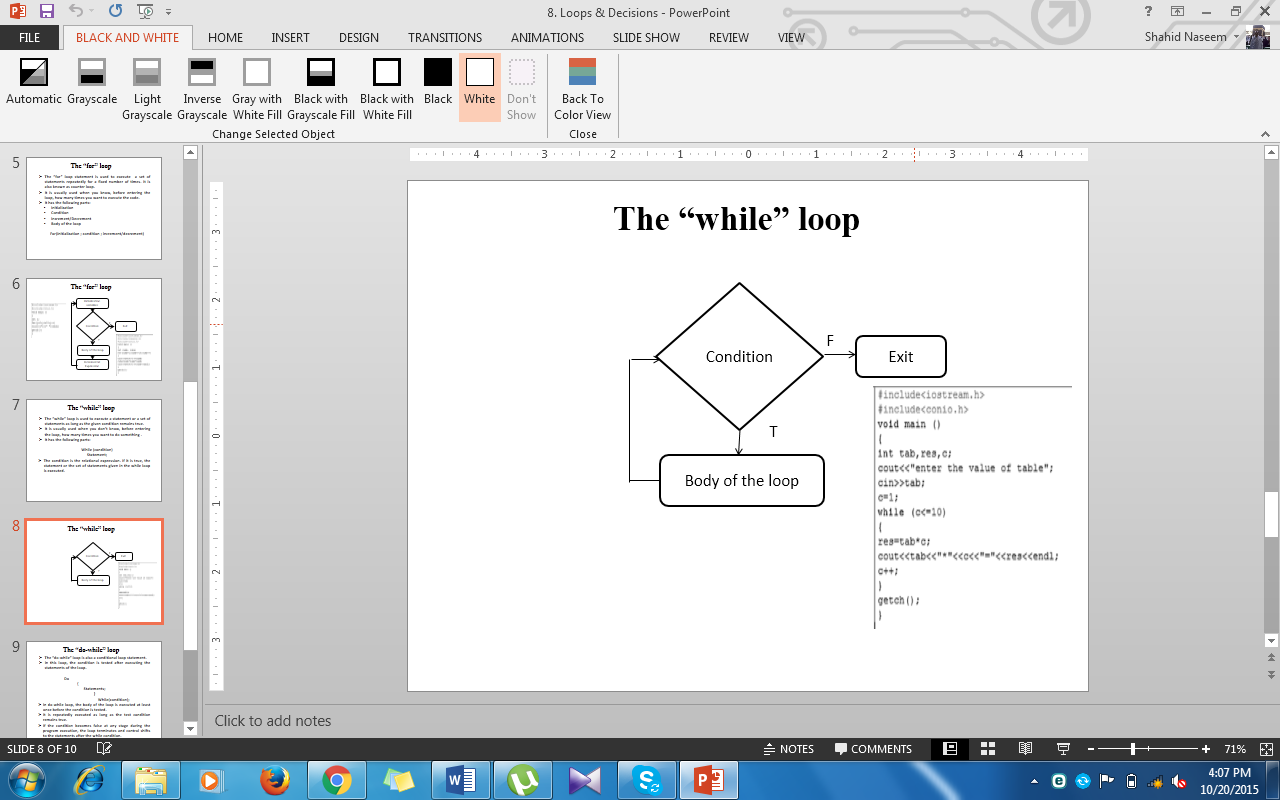
**The “while” loop**

* + The “while” loop is used to execute a statement or a set of statements as long as the given condition remains true.
  + It is usually used when you don’t know, before entering the loop, how many times you want to do something.
  + It has the following parts:

While (condition)

Statement;

* The condition is the relational expression. If it is true, the statement or the set of statements given in the while loop is executed.



**The “do-while” loop**

* + The “do-while” loop is also a conditional loop statement.
  + In this loop, the condition is tested after executing the statements of the loop.

Do

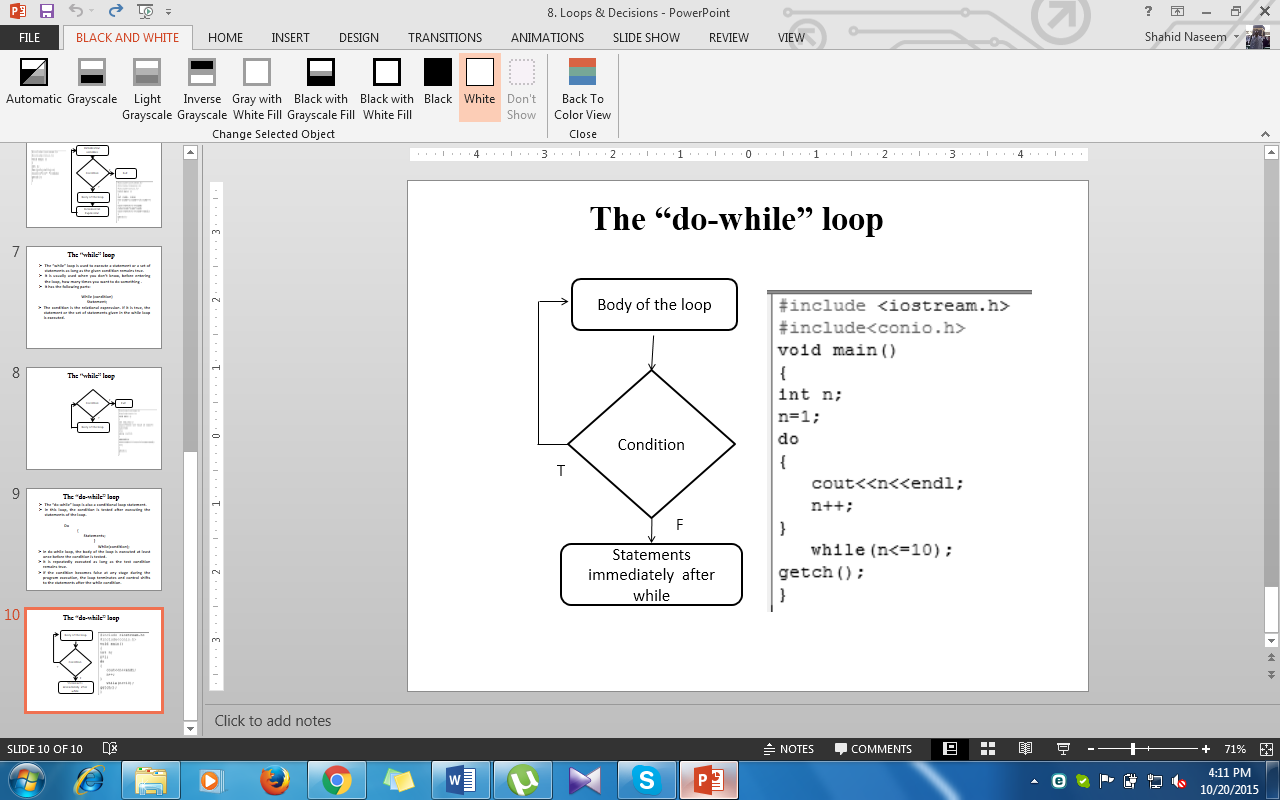
{

Statements;

}

While(condition);

* + In do-while loop, the body of the loop is executed at least once before the condition is tested.
  + It is repeatedly executed as long as the test condition remains true.
  + If the condition becomes false at any stage during the program execution, the loop terminates and control shifts to the statements after the while condition.



**Activity No.23 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to print the multiplication table of a number entered from the user using While Loop.**

**Output of the Program**

**Activity No.24 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the factorial of any number and display the result on screen using While Loop.**

**Output of the Program**

**Activity No.25 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the even numbers from 1 to 10 natural numbers, also display the sum of these numbers using While Loop.**

**Output of the Program**

**Activity No.26 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the sum of the series given below using While Loop.**

**1+1/2+1/3+1/4+------------------------------------------------------------+1/45**

**Output of the Program**

**1-D/2-D Arrays/Strings**

**ARRAYS**

* + An array is a sequence of objects of same data type.
  + The objects in an array are called “elements of array”.
  + An array is represented in the computer memory by a consecutive group of storage locations referenced by a single variable called “array name”.
  + Each element of an array is referenced by its position in the array and is represented by an index value or subscript.
  + In an array, the elements are indexed like 0,1,------------n-1.
  + Where ‘0’ represents the value of the first element and

‘n-1’ is the position of the last element.

* + An element of an array is accessed by its subscript value.
  + The subscript of an array is written inside a pair of square brackets [] with the name of the array.

**Types of Arrays**

* + There are two types of arrays:

i. One – dimensional arrays

ii. Two – dimensional arrays.

One-dimensional Array: One – dimensional array is also known as a list or a linear array. It consists of only one column or one row.

* + - Defining the name of array, its type and the total number of elements of an array is called declaring of the array.

Type array\_name [n];

Temp[23]

**Temp**



Temp[0]

Temp[2]

Temp[1]

Temp[n]

**Activity No.27 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take 7 temperature float values in an array and display these values on computer screen.**

**Output of the Program**

**Activity No.28 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take 7 temperature float values in first array, copy these values in a second array and then display the values in second array on computer screen.**

**Output of the Program**

**Activity No.29 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that take the characters of the name of student in first array, copy these characters in second array and display the name of the student from second array.**

**Output of the Program**

**Activity No.30 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that takes two one-dimensional arrays from user and display the arrays as well as sum of these arrays on the screen.**

**Output of the Program**

**Bubble Sorting**

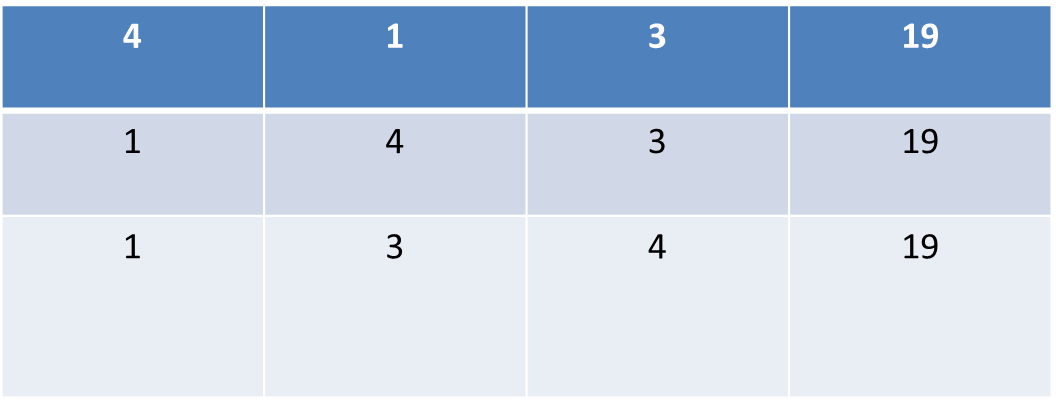
* + The bubble sort method is used to arrange values of an array in ascending or in descending order.
  + To arrange an array in ascending order, two neighboring elements are compared.
  + If one element is greater than the other, the two are exchanged.
  + Through the exchange of elements, the greater value bubble up to the top.
  + To arrange an array in descending order, the smaller value floats up to the top.
  + Bubble sort method is a slow process. It is used for sorting only small amount of data.

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 19 | 1 | 3 |

**Iteration -1:**

|  |  |  |  |
| --- | --- | --- | --- |
| **4** | **19** | **1** | **3** |
| 4 | 19 | 1 | 3 |
| 4 | 1 | 9 | 3 |
| 4 | 1 | 3 | 19 |

* + In first iteration the value of element ARR[0] is compared with element ARR[1]. Since 4 is less than 19, no need to interchange the values.
  + The values of ARR[1] and ARR[2] are compared. Since 19 is greater than 1, the values are interchanged.
  + The values of ARR[2] and ARR[3] are compared. Since 19 is greater than 3, the values are interchanged.

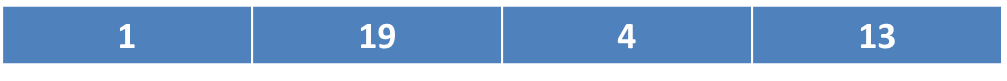
**Iteration-2:**

* + In second iteration, the value of element ARR[10] is compared with element ARR[1]. Since 4 is greater than 1, the values are interchanged.
  + The value of second element ARR[1] is compared with third element ARR[2]. Since 4 is greater than 3, the values are interchanged.
  + At the end of the second iteration, the second largest value moves to the second position in the array.
  + The selection sort method is used for arranging arrays in ascending or descending order.

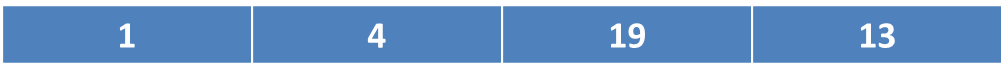
**Selection Sorting**

|  |  |  |  |
| --- | --- | --- | --- |
| **4** | **19** | **1** | **13** |

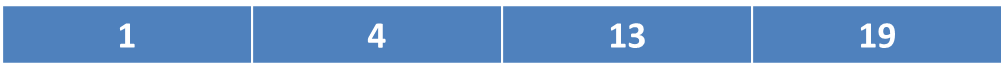
* + **Iteration -1:** Find out the smallest value from the list starting from first element to the last element of the list. The smallest value is 1. Note the address of element that has the smallest value. Interchange the value of first element with the third element. Swap values of ARR[0] and ARR[2].



* + **Iteration -2:** Find out the smallest value from the list starting from the second element to the last element. The smallest value is 4 at location 3. Interchange the value of the second element with the third element i.e. swap values of ARR[1] and ARR[2].



* **Iteration -3:** Find out the smallest value from the list starting from the third element to the last element. The smallest value is 13 at location 4. Interchange the value of the third element with the fourth element i.e. swap values of ARR[2] and ARR[3].



**Activity No.31 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display an array of an order 2x3.**

**Output of the Program**

**Activity No.32 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the transpose of an array of order 2x3.**

**Output of the Program**

**Activity No.33 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program that takes two arrays of order 2x2 and display the sum, difference and multiplication of these arrays.**

#include<iostream.h>

#include<conio.h>

void main()

{

int a[2][2],b[2][2],sum[2][2],mul[2][2],sub[2][2],i,j,k,s,d1,d2;

cout<<"first matrix "<<endl;

for(i=0;i<=1;i++)

{

for(j=0;j<=1;j++)

{

cout<<"enter values in first matrix";

cin>>a[i][j];

}

}

cout<<"second matrix"<<endl;

for(i=0;i<=1;i++)

{

for(j=0;j<=1;j++)

{

cout<<"enter values in second matrix";

cin>>b[i][j];

}

}

cout<<"sum="<<endl;

for(i=0;i<=1;i++)

{

for(j=0;j<=1;j++)

{

sum[i][j]=a[i][j]+b[i][j];

cout<<sum[i][j]<<" ";

}

cout<<endl;

}

cout<<"subtraction="<<endl;

for(i=0;i<=1;i++)

{

for(j=0;j<=1;j++)

{

sub[i][j]=a[i][j]-b[i][j];

cout<<sub[i][j]<<" ";

}

cout<<endl;

}

cout<<"multiplication is"<<endl;

for(i=0;i<=1;i++)

{

for(j=0;j<=1;j++)

{

for(k=0;k<=1;k++)

s=s+a[i][k]\*b[k][j];

cout<<s<<" ";

s=0;

}

cout<<endl;

}

cout<<"determinant of a"<<endl;

d1=a[0][0]\*a[1][1]-a[0][1]\*a[1][0];

cout<<d1<<" "<<endl;

cout<<"determinant of b"<<endl;

d2=b[0][0]\*b[1][1]-b[0][1]\*b[1][0];

cout<<d2<<" ";

getch();

}

**Output of the Program**

**Activity No.34 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the values of x,y,z using cramer’s rule.**

#include <iostream.h>

#include<conio.h>

void main()

{

int A[3][3],B[3],D,D1,D2,D3, x, y,z;

cout<<"cramer's rule to solve linear equation x-2y+3z=4"<<endl;

cout<<"enter equation 1:"<<endl;

cin>>A[0][0]>>A[0][1]>>A[0][2];

cout<<"enter equation 2:"<<endl;

cin>>A[1][0]>>A[1][1]>>A[1][2];

cout<<"enter equation 3:"<<endl;

cin>>A[2][0]>>A[2][1]>>A[2][2];

cout<<"enter fourth column:"<<endl;

cin>>B[0]>>B[1]>>B[2];

D= A[0][0]\*((A[1][1]\*A[2][2])-(A[1][2]\*A[2][1]))

- A[0][1]\*((A[1][0]-A[2][2])-(A[1][2]-A[2][0]))

+ A[0][2]\*((A[1][0]\*A[2][1])-(A[1][1]\*A[2][0]));

D1= B[0]\*((A[1][1]\*A[2][2])-(A[1][2]\*A[2][1]))

- A[0][1]\*((B[1]-A[2][2])-(A[1][2]-B[3]))

+ A[0][2]\*((B[1]\*A[2][1])-(A[1][1]\*B[2]));

D2= A[0][0]\*((B[1]\*A[2][2])-(A[1][2]\*B[2]))

- B[0]\*((A[1][0]-A[2][2])-(A[1][2]-A[2][0]))

+ A[0][2]\*((A[1][0]\*B[2])-(B[1]\*A[2][0]));

D3= A[0][0]\*((A[1][1]\*B[2])-(B[1]\*A[2][1]))

- B[0]\*((A[1][0]-B[2])-(B[1]-A[2][0]))

+ B[0]\*((A[1][0]\*A[2][1])-(A[1][1]\*A[2][0]));

/\* Code that determines if the system has a unique solution \*/

if((D)!=0)

{

cout<<"\nSystem has a unique solution"<<endl;

x= D1/D;

y= D2/D;

z=D3/D;

cout<<"x="<<x<<endl;

cout<<"y="<<y<<endl;

cout<<"z="<<z<<endl;

}

else

{

cout<<"\nSystem does not have a unique solution because determinant is 0";

}

getch();

}

**Activity No.35 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the determinant of 2x2 martix.**

#include<iostream.h>

#include<conio.h>

void main()

{

int d1[2][2];

int i,j,ans;

cout<<"first matrix"<<endl;

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

cout<<"enter number";

cin>>d1[i][j];

}

}

cout<<"determinent of matrix"<<endl;

ans=(d1[0][0]\*d1[1][1]-d1[1][0]\*d1[0][1]);

cout<<ans<<endl;

ans=0;

getch();

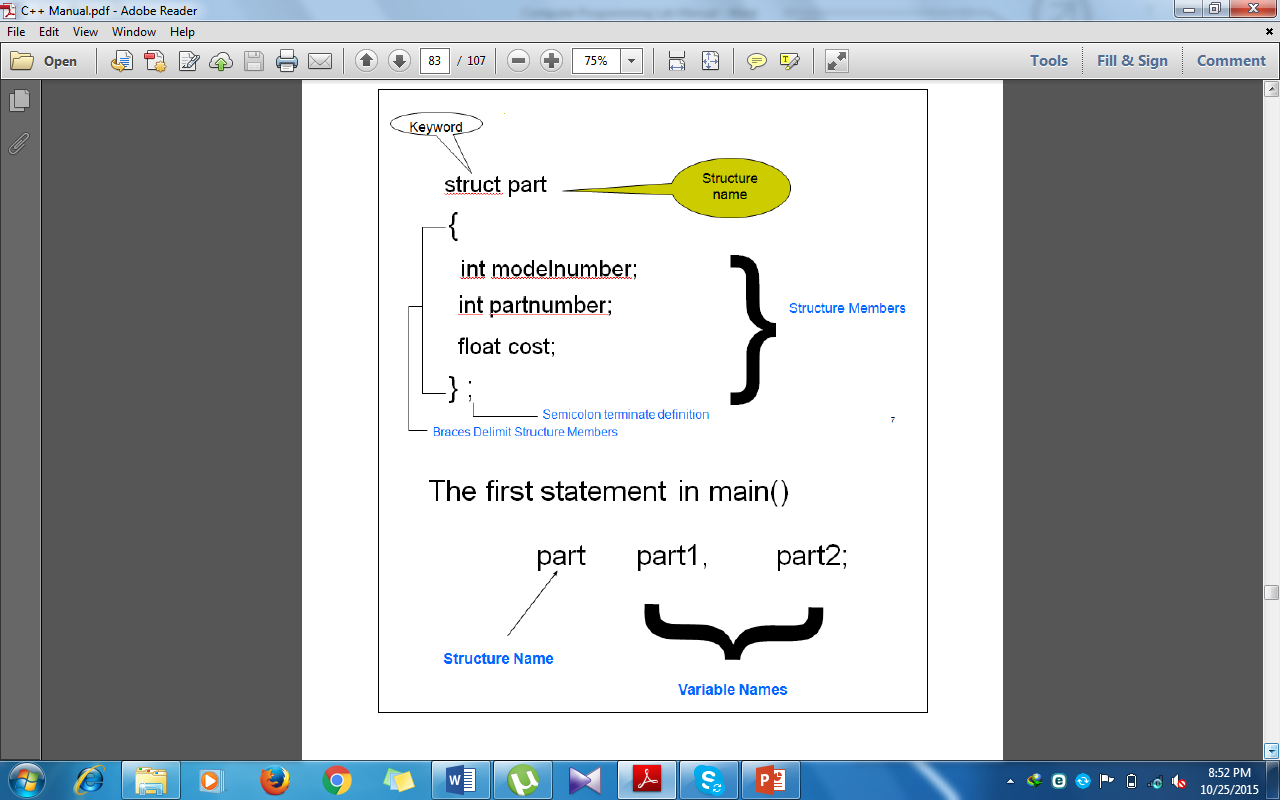
}

**Structures**

**Structures**

* + A structure is a collection of related elements or data items.
  + The elements of the structure are called “members of the structure”.
  + A structure is unlike an array.
  + All elements in an array are of the same type, but in structure, the elements can be of different types.
  + The structures are commonly used in file processing.
  + A file is a collection of related records and a record is a collection of related records and a record is a collection of related fields of same or different data types also known as “record”.

The structure definition serves only as a blueprint for the creation of variables of type part. It does not itself create any structure variables; that is, it does not set aside any space in memory or even name any variables. This is unlike the definition of a simple variable, which does set aside memory.



**Activity No.36 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the structure of model number, part number and cost of two cars.**

**Output of the Program**

**Activity No.37 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the structure of quiz, assignment, mid-term, final- term and total marks to three students.**

**Output of the Program**

**Activity No.38 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the structure of students name, age, course taken and GPA.**

**Output of the Program**

**Activity No.39 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the structure of employee having employee ID, employee name, employee designation, date of birth and date of joining.**

**Output of the Program**

**Functions**

**Functions**

* + Functions allow to structure programs in segments of code to perform individual tasks.
  + In function, a large program is divided into subprograms and each subprogram is solved separately and at the end, all the solutions of these subprograms are joined to find the solution of the large program.
  + These subprograms are called sub-routines.
  + In C++, a function is a group of statements that is given a name, and which can be called from some point of the program.

**Syntax of Function**

type name ( parameter1, parameter2, ...)

{

Statements;

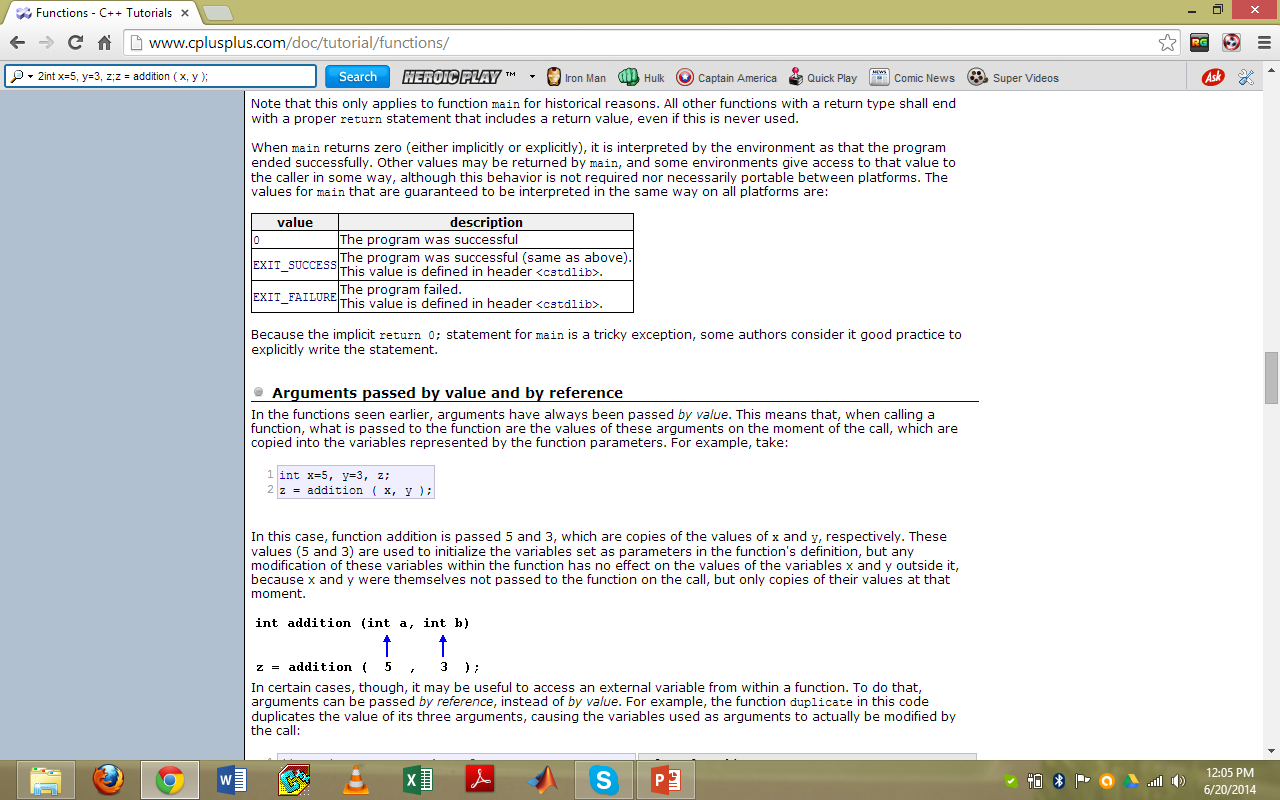
}

**Local Variables and Global Variables**

* + **Local Variables:** The variables that are declared within the main function but can’t access by everyone is called local variables.
  + **Global Variables:** The variables that are declared outside the main function are called global variables. These functions can be accessed by anyone and can be used in the program where it is needed.

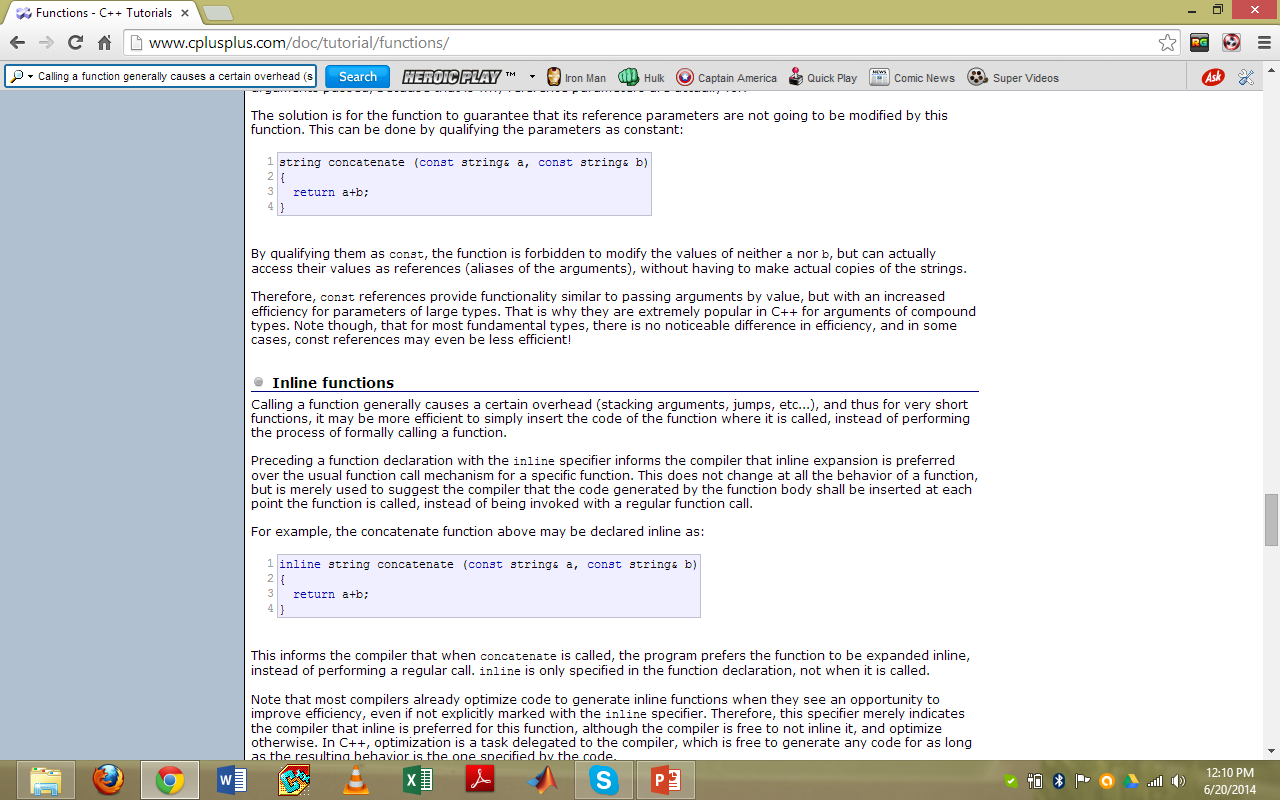
**Arguments passed by values**

* + In the functions, arguments have always been passed *by value*. This means that, when calling a function, what is passed to the function are the values of these arguments on the moment of the call, which are copied into the variables represented by the function parameters.

****

**Inline Functions**

* + Calling a function generally causes a certain overhead (stacking arguments, jumps, etc...), and thus for very short functions, it may be more efficient to simply insert the code of the function where it is called, instead of performing the process of formally calling a function.
  + Inline function informs the compiler that inline expansion is preferred over the usual function call mechanism for a specific function. This does not change at all the behavior of a function, but is merely used to suggest the compiler that the code generated by the function body shall be inserted at each point the function is called, instead of being invoked with a regular function call.



**Activity No.40 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the area of circle using function.**

#include<iostream.h>

#include<conio.h>

void main()

{

void area(float);

area(7.5);

}

void area(float r)

{

float area;

area=3.1415\*r\*r;

cout<<"Area of the circle is="<<area<<endl;

getch();

}

**Output of the Program**

**Activity No41 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the area of rectangle using function.**

#include<iostream.h>

#include<conio.h>

void main()

{

void area(int,int);

area(10,20);

}

void area(int m,int n)

{

int area;

area=m\*n;

cout<<"Area of the rectangle is="<<area<<endl;

getch();

}

**Output of the Program**

**Activity No.42 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the area of triangle using function.**

#include<iostream.h>

#include<conio.h>

void main()

{

void area(float,float);

area(6.2,5.4);

}

void area(float m,float n)

{

float area;

area=0.5\*m\*n;

cout<<"Area of the triangle is="<<area<<endl;

getch();

}

**Output of the Program**

**Activity No.43 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the number is even or odd using function.**

#include<iostream.h>

#include<conio.h>

void main()

{

int comp(int);

int n;

cout<<"enter a number";

cin>>n;

comp(n);

}

int comp(int x)

{

if(x%2==0)

{

cout<<"number is even"<<endl;

}

else

{

cout<<"number is odd"<<endl;

}

getch();

}

**Activity No.44 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to apply operators using functions with switch statement.**

#include<iostream.h>

#include<conio.h>

void main()

{

void cal (int, char, int);

int n1,n2;

char op;

cout<<"enter first number"<<endl;

cin>>n1;

cout<<"enter second number"<<endl;

cin>>n2;

cout<<"enter operator"<<endl;

cin>>op;

cal(n1,op,n2);

}

void cal(int n1,char op,int n2)

{

switch(op)

{

case '+':

cout<<(n1+n2);

break;

case '-':

cout<<(n1-n2);

break;

case '\*':

cout<<(n1\*n2);

break;

case '/':

cout<<(n1/n2);

break;

case '%':

cout<<(n1%n2);

break;

default:

cout<<"invalid operator";

}

getch();

}

**Activity No.45 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to find the table of any number using function.**

#include<iostream.h>

#include<conio.h>

void main()

{

void tab(int);

int n;

cout<<"enter any number"<<endl;

cin>>n;

tab(n);

}

void tab(int n)

{

for(int c=1;c<=10;c++)

cout<<n<<"\*"<<c<<"="<<n\*c<<endl;

getch();

}

**Classes**

**Classes**

* + Classes are an expanded concept of data structures: like data structures, they can contain data members, but they can also contain functions as members.
  + An object is an instantiation of a class. In terms of variables, a class would be the type, and an object would be the variable.
  + Classes are defined using either keyword class or keyword struct, with the following syntax:

Class area

{

Private:

Int a,b;

Public:

};

**Members of a Class**

* + **Private Members**: Private or protected members of a class are accessible only from within other members of the same class.
  + **Public Members:** Finally, public members are accessible from anywhere where the object is visible.

**Activity No.46 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the three numbers using a class.**

#include<iostream.h>

#include<conio.h>

class xyz

{

private:

int a,b,c;

public:

void get()

{

cout<<"enter three number"<<endl;

cin>>a>>b>>c;

}

void put()

{

cout<<"a="<<a<<endl;

cout<<"b="<<b<<endl;

cout<<"c="<<c;

}

};

void main()

{

xyz data;

data.get();

data.put();

getch();

}

**Activity No.47 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the name, age, year of admission, and GPA of a student using a class.**

#include<iostream.h>

#include<conio.h>

class student

{

private:

char name[10];

int age,year,gpa;

public:

void get()

{

cout<<" student name"<<endl;

cin>>name;

cout<<"enter age of student"<<endl;

cin>>age;

cout<<"enter registration year"<<endl;

cin>>year;

cout<<"enter gpa"<<endl;

cin>>gpa;

}

void put()

{

cout<<"Student Name="<<name<<endl;

cout<<"Student Age="<<age<<endl;

cout<<"Registration Year="<<year<<endl;

cout<<"GPA="<<gpa;

}

};

void main()

{

student s1;

s1.get();

s1.put();

getch();

}

**Activity No.48 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display name, basic pay, house rent, medical allowance and net pay of an employee using a class.**

#include<iostream.h>

#include<conio.h>

class emp

{

private:

char name[10];

int bpay;

float h\_rent,ma,n\_pay;

public:

void get()

{

cout<<"enter employee name"<<endl;

cin>>name;

cout<<"enter basic pay"<<endl;

cin>>bpay;

}

void allow()

{

h\_rent=bpay\*60/100.0;

ma=bpay\*20/100.0;

n\_pay=bpay+h\_rent+ma;

}

void put()

{

cout<<"Employee Name="<<name<<endl;

cout<<"Basic Pay="<<bpay<<endl;

cout<<"House Rent="<<h\_rent<<endl;

cout<<"Medical Allowance="<<ma<<endl;

cout<<"Net Pay="<<n\_pay<<endl;

}

};

void main()

{

emp E1;

E1.get();

E1.allow();

E1.put();

getch();

}

**Activity No.49 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the sum of two numbers using destructor in a class.**

#include<iostream.h>

#include<conio.h>

class prg

{

public:

prg()

{

cout<<"this is constructor function"<<endl;

}

~prg()

{

cout<<"this is destructor function"<<endl;

}

};

void main()

{

prg x;

int a,b;

a=10,b=20;

cout<<"sum="<<(a+b)<<endl;

getch();

}

**Activity No.50 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the sum of two numbers using constructor in a class.**

#include<iostream.h>

#include<conio.h>

class sum

{

private:

int n,m,s;

public:

sum(int x, int y)

{

n=x;

m=y;

s=n+m;

}

psum()

{

cout<<"sum="<<s<<endl;

}

};

void main()

{

sum a(16,10),b(2,3);

a.psum();

b.psum();

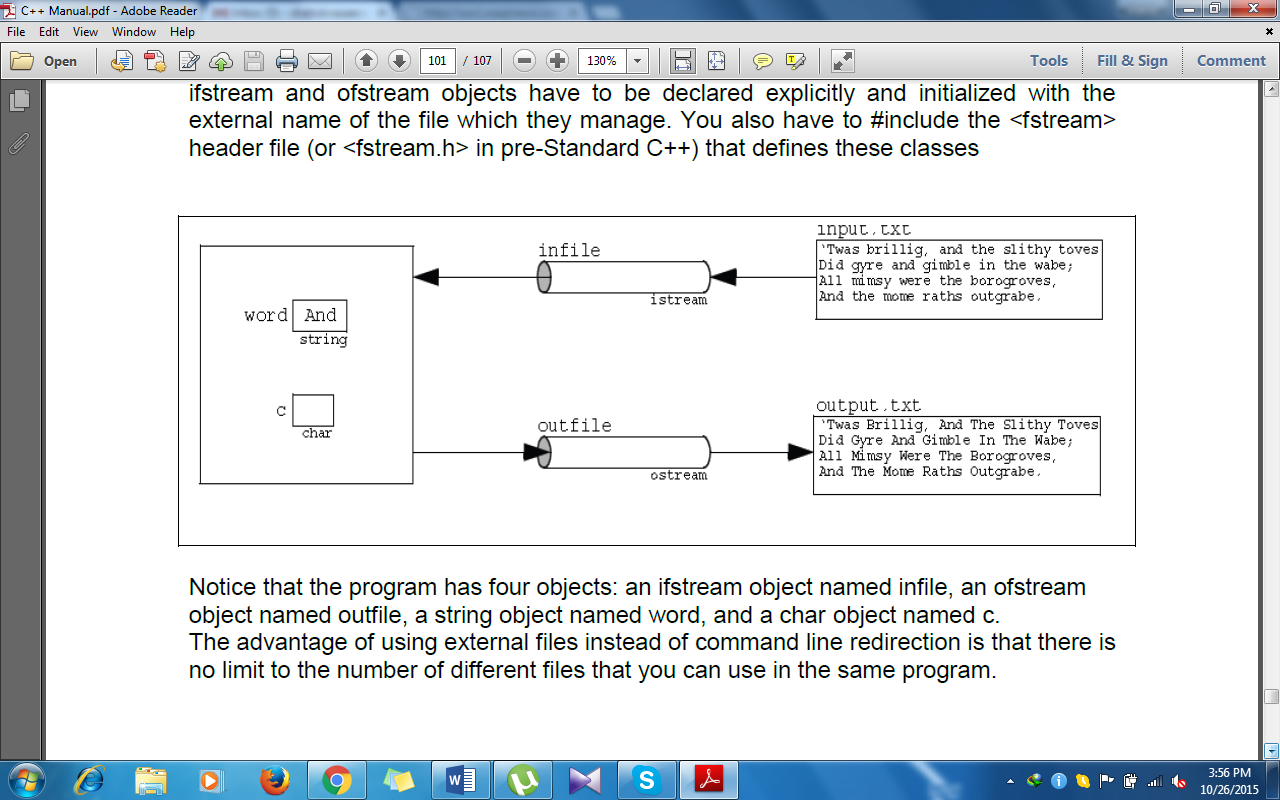
getch();

}

**File Handling**

**Introduction to Files**

File processing in C++ is very similar to ordinary interactive input and output because the same kinds of stream objects are used. Input from a file is managed by an ifstream object the same way that input from the keyboard is managed by the istream object cin. Similarly, output to a file is managed by an ofstream object the same way that output to the monitor or printer is managed by the ostream object cout. The only difference is that ifstream and ofstream objects have to be declared explicitly and initialized with the external name of the file which they manage. You also have to #include the <fstream> header file (or <fstream.h> in pre-Standard C++) that defines these classes.



The program has four objects: an ifstream object named infile, an ofstream object named outfile, a string object named word, and a char object named c. The advantage of using external files instead of command line redirection is that there is no limit to the number of different files that you can use in the same program.

**Files in C++**

#include <iostream.h>

#include <conio.h>

#include <fstream>

void main ()

{

ofstream myfile;

myfile.open ("first.txt");

myfile << "This is my first file \n";

myfile.close();

getch();

}

**Open a file**

Open (filename, mode);

 Mode is optional parameter with a combination of the following flags

ios::in Open for input operations.

ios::out Open for output operations.

ios::binary Open in binary mode. Set the initial position at the end of the file.

ios::ate If this flag is not set to any value, the initial position is the beginning of the file.

ios::app All output operations are performed at the end of the file, appending the content to the current content of the file. This flag can only be used in streams open for output-only operations.

ios::trunc If the file opened for output operations already existed before, its previous content is deleted and replaced by the new one.

All these flags can be combined using the bitwise operator OR (|). For example, if we want to open the file example.bin in binary mode to add data we could do it by the following call to member function open(): ofstream myfile;myfile.open ("example.bin", ios::out | ios::app | ios::binary).

**Activity No.51 Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-----------------------------------------------------------------------------------------------------------**

**Write a program to display the sum, subtraction, multiplication and division of three numbers using fstream.**

#include<iostream>

#include<fstream>

#include<conio>

void main()

{

ifstream input\_stream;

ofstream output\_stream;

input\_stream.open("input.txt");

output\_stream.open("ucest.txt");

int first=2,second=4,third=6;

input\_stream>>first>>second>>third;

cout<<"sum="<<(first+second+third)<<endl;

output\_stream<<"sum="<<(first+second+third)<<endl;

cout<<"sub="<<(first-second-third)<<endl;

output\_stream<<"sub="<<(first-second-third)<<endl;

cout<<"mul="<<(first\*second\*third)<<endl;

output\_stream<<"mul="<<(first\*second\*third)<<endl;

cout<<"div="<<(first/second/third)<<endl;

output\_stream<<"div="<<(first/second/third)<<endl;

input\_stream.close();

output\_stream.close();

getch();

}