



ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category-I University by MHRD-UGC]

KARAIKUDI – 630 003

DIRECTORATE OF DISTANCE EDUCATION



**B.C.A.
101 24**



LAB: PROGRAMMING IN C++

II - Semester



ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category-I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003



Directorate of Distance Education

B.C.A.

II - Semester

101 24

LAB: PROGRAMMING IN C++

Author:

Dr. Kavita Saini, Associate Professor, Galgotias University, Greater Noida

"The copyright shall be vested with Alagappa University"

All rights reserved. No part of this publication which is material protected by this copyright notice may be reproduced or transmitted or utilized or stored in any form or by any means now known or hereinafter invented, electronic, digital or mechanical, including photocopying, scanning, recording or by any information storage or retrieval system, without prior written permission from the Alagappa University, Karaikudi, Tamil Nadu.

Information contained in this book has been published by VIKAS® Publishing House Pvt. Ltd. and has been obtained by its Authors from sources believed to be reliable and are correct to the best of their knowledge. However, the Alagappa University, Publisher and its Authors shall in no event be liable for any errors, omissions or damages arising out of use of this information and specifically disclaim any implied warranties or merchantability or fitness for any particular use.



VIKAS®

Vikas® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.

E-28, Sector-8, Noida - 201301 (UP)

Phone: 0120-4078900 • Fax: 0120-4078999

Regd. Office: 7361, Ravindra Mansion, Ram Nagar, New Delhi 110 055

• Website: www.vikaspublishing.com • Email: helpline@vikaspublishing.com

Work Order No. AU/DDE/DE1-291/Preparation and Printing of Course Materials/2018 Dated 19.11.2018 Copies - 500

LAB: PROGRAMMING IN C++

Syllabi

BLOCK 1

1. Simple Programs using decisions, loops and arrays
2. Simple functions & Inline functions

BLOCK 2

3. Usage of classes and Objects
4. This pointer and Static functions
5. Constructors and Destructors

BLOCK 3

6. Function overloading
7. Operator Overloading
8. Friend functions

BLOCK 4

9. Inheritance & Multiple Inheritance
10. Pointers
11. Polymorphism
12. Virtual Functions

BLOCK 5

13. Files
 14. I/O Streams
-

INTRODUCTION

NOTES

C++ has become one of the most popular OOP languages used for developing real-world applications. C++ is a programming language that extended from the ubiquitous C language. It treats data as a crucial element—not allowing it to move freely around the system. Therefore, the main emphasis in C is on data and not on the procedure. You can design programs around the data being operated upon in C++. An object-oriented language helps in combining data and functions that operate on data into a single unit known as object. C++ is used for developing different types of applications, such as real-time systems, simulation modelling, expert systems. It also provides flexibility to a user to introduce new types of objects in his programming on the basis of the requirement of the application.

This lab manual, *Lab: Programming in C++*, contains several programs based on C++ concepts, such as classes, inheritance, constructors and destructors, to provide the concept of programming. In addition, it will help students in coding and debugging their programs. The manual provides all logical, mathematical and conceptual programs that can help to write programs very easily in C++ language. These exercises shall be taken as the base reference during lab activities for students of BCA. There are also many Try Yourself Questions provided to students for implementation in the lab.

INTRODUCTION

C++ language is invented by Bjarne Stroustrup in 1980 at Bell Laboratories, New Jersey. C++ language was initially called “C with Classes” but in 1983 this name was changed to C++. C++ is a superset of C.

The purpose of C++ is to overcome this limit and provide a better way to manage larger, more complex programs, by using object oriented programming (OOP). C++ is very popular language as it has many features as mentioned below:

- Classes and objects
- Encapsulation
- Information hiding
- Inheritance
- Polymorphism

Portable Language

It is the concept of carrying the instruction from one system to another system. In C++ language **.cpp** file contain source code, we can also edit this code and **.exe** file contain application, only we can execute this file. When we write and compile any C++ program on window operating system then that program easily run on other window based system.

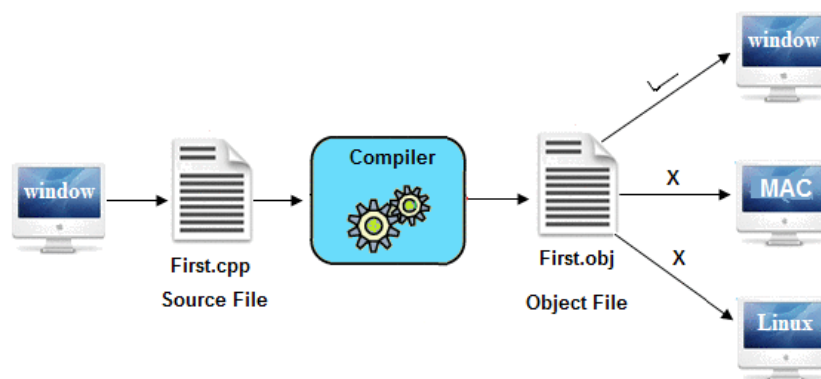


Fig. 1.1 Representing the C++ object file running on Windows.

When we can copy .exe file to any other computer which contain window operating system then it works properly, because the native code of application of operating system is same.

NOTES

RECOMMENDED SYSTEM / SOFTWARE REQUIREMENTS

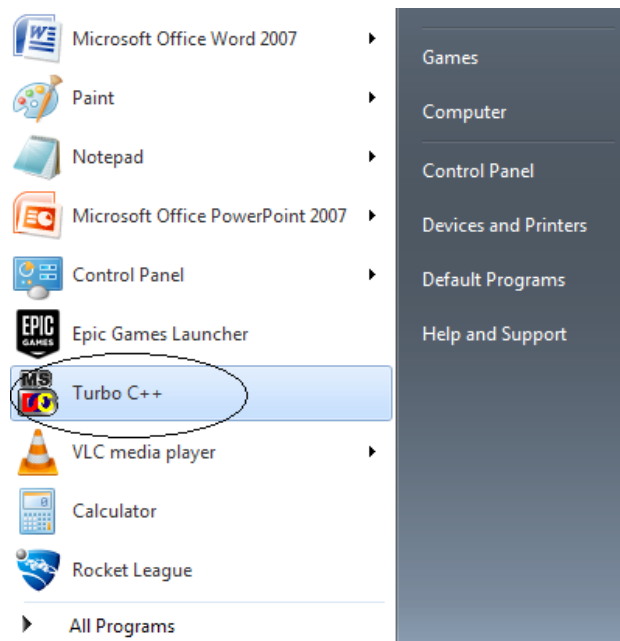
NOTES

1. Intel based desktop PC of 166MHz or faster processor with at least 64 MB RAM and 100 MB free disk space.
2. Turbo C++ compiler or GCC compilers.

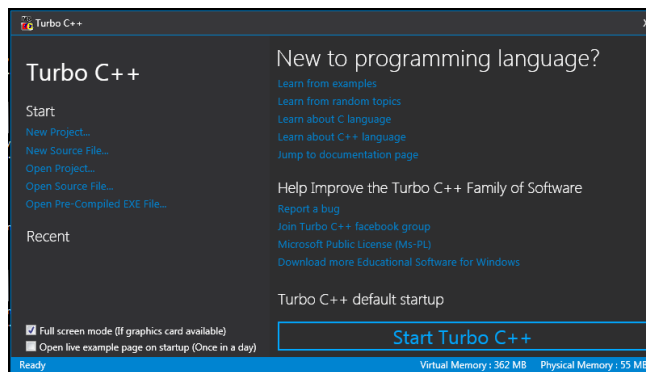
In this manual we have used Turbo C++. To write C++ code first we need to open Turbo C++. For every C++ program we need to follow following steps for writing and executing a program.

Write a program code → save your program (F2) → compile (Alt+F9) → Run(Ctrl +F9)

Step 1: Click on Turbo C++ from start menu or double click on Turbo C++ on desktop.

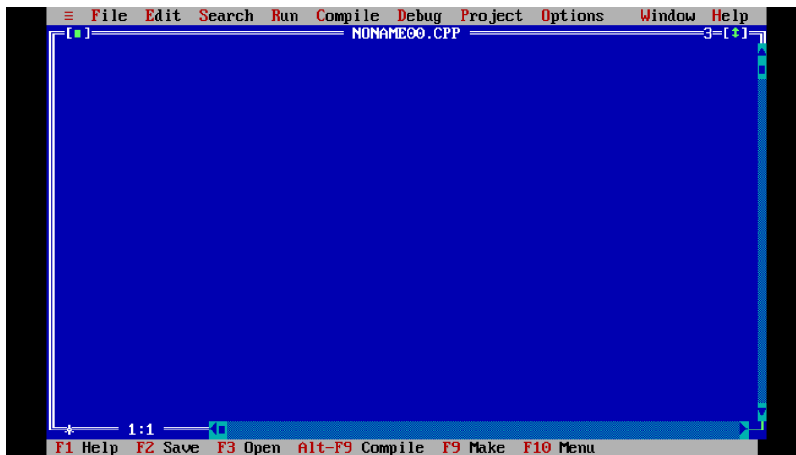


After clicking on Turbo C++ following screen will appear:



Step 2: Click on Start Turbo C++. After clicking on Start Turbo C++ button following screen will appear:

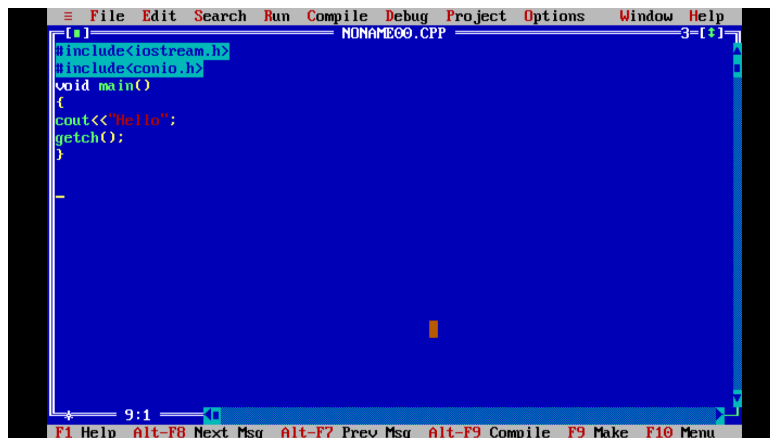
Lab: Programming in C++



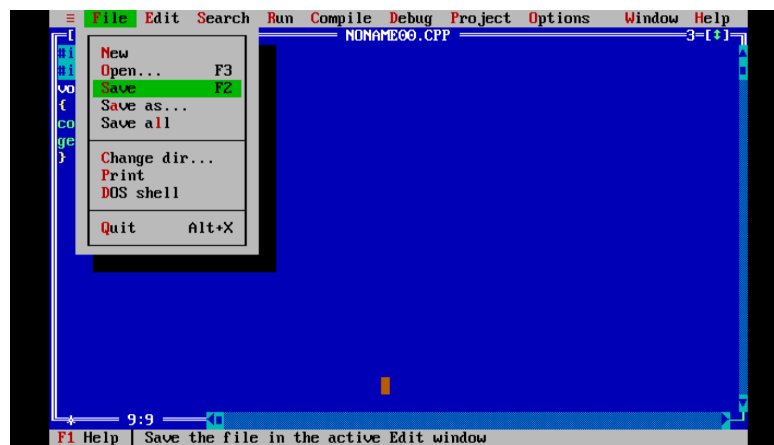
NOTES

This is the editor where we will write code of C++ programs.

Step 3: Write a program to print “Hello” on screen (Hello.cpp).

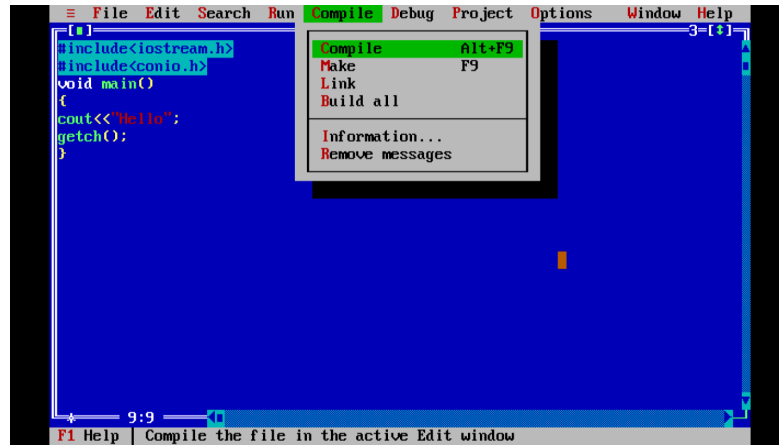


Step 4: Save program by name hello.cpp by pressing F2 key or by using menu option File → Save As.

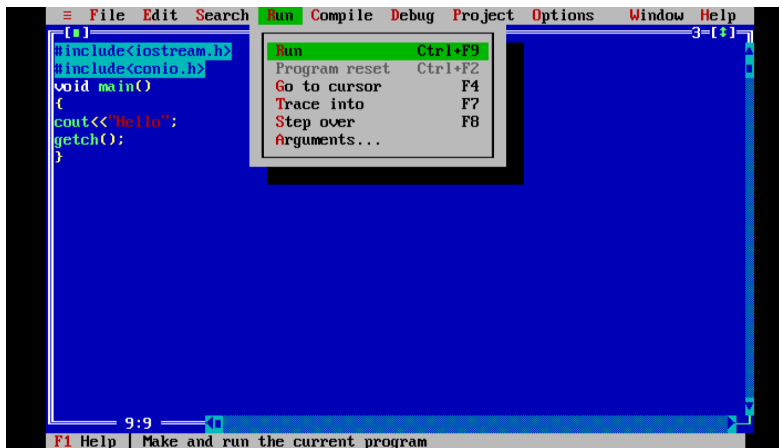


Step 5: Compile program i.e. hello.cpp by pressing Alt+F9 keys or by using menu option Compile → Compile:

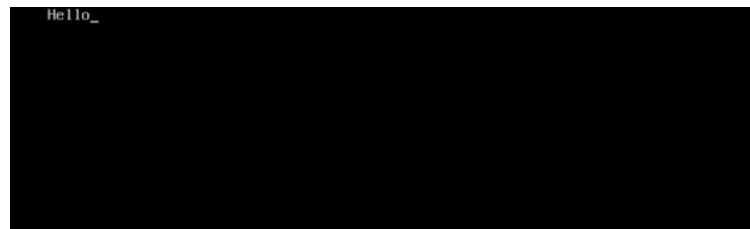
NOTES



Step 6: Run program i.e. hello.cpp by pressing Ctrl +F9 keys or by using menu option Run → Run.



Output:



1. Write a program that takes two numbers as input and print their sum and average.

```
//Program to take two numbers as input and provides  
sum and average  
#include<iostream.h>
```

```

void main()
{
int num1, num2, sum, avg;
cout<<"Enter two numbers"<<endl; //output statement
cin>>num1; //input statement
cin>>num2;
sum=num1+num2;
avg=sum/2;
cout<<"Sum of two numbers  "<<sum<<endl;
cout<<"Average of two numbers  "<<avg;
}

```

NOTES**Output:**

```

Enter two numbers
65
87
Sum of two numbers 152
Average of two numbers 76

```

2. Write a program to swap two numbers without using a third variable.

```

// Program to swap two numbers without using a third
variable
#include <iostream.h>
void main()
{
int num1, num2;
cout<<"Enter two numbers"<<endl;
cin>>num1>>num2;
num2 = num1+num2;
num1 = num2 - num1;
num2 = num2 - num1;
cout<<"values after swaping :\n";
cout<<"Value of a Num1 "<<num1<<endl;
cout<<"Value of a Num2 "<<num2<<endl;
}

```

Output:

```

Enter two numbers
3
4
values after swaping :
Value of a Num1 4
Value of a Num2 3

```

NOTES

Try yourself:

- (i) Write a program to calculate volume of cylinder.
Volume of cylinder= $PI*r*r*h$.
- (ii) Write a program to calculate curved surface area of cylinder.
Curved surface area of cylinder= $2*PI*r*h$
- (iii) Write a program to print ASCII value of digits, uppercase and lowercase alphabets.

3. Write a program check whether the given number is even or odd.

```
// Program to check whether number is even or odd

#include <iostream.h>
void main()
{
int num;
cout<<"Enter a number ";
cin>>num;
if (num%2==0)
{
cout<<"Number is even ";
}
else
{
cout<<"Number is odd ";
}
}
```

Output:

```
Enter a number 2
Number is even
```

4. Write a program to print the largest number among three numbers given by the user.

```
// program print the largest number among three numbers

#include <iostream.h>
void main()
{
```

```

int num1, num2, num3;
cout<<"Enter three numbers"<<endl;
cin>>num1>>num2>>num3;

if(num1 >= num2 && num1 >= num3)
{
    cout << "Largest number: " << num1;
}
else if(num2 >= num1 && num2 >= num3)
{
    cout << "Largest number: " << num2;
}
else
{
    cout << "Largest number: " << num3;
}
}

```

Output:

```

Enter three numbers
34
6
7
Largest number: 34

```

5. Write a program to print sum, difference, product and division of two numbers according to the user choice using Switch case.

```

#include <iostream.h>
void main()
{
    int num1, num2; char op;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;
    cout << "Enter operator : ";
    cin >> op;
    switch (op)
    {
        case '+': cout <<"\n Sum of two numbers "<<
                num1+num2;
                break;

```

NOTES

NOTES

```
case '-': cout << "\n Subtraction of two numbers " <<
          num1-num2;
          break;
case '*': cout << "\n Multiplication of two numbers
          " << num1*num2;
          break;
case '/': cout << "\n Division of two numbers " <<
          num1/num2;
          break;
default:  cout << "\n Invalid operator";
          break;
    }
}
```

Output:

```
Enter two numbers: 56
9
Enter operator : *
Multiplication of two numbers 504
```

Try yourself:

- (i) Write a program to convert a lowercase alphabet to uppercase and vice-versa.
- (ii) Write a program to check whether a year is leap year or not.
- (iii) Write a program to check whether a given character is uppercase or lowercase alphabet or a digit or a special character.

6. Write a program to print table of any number using for loop.

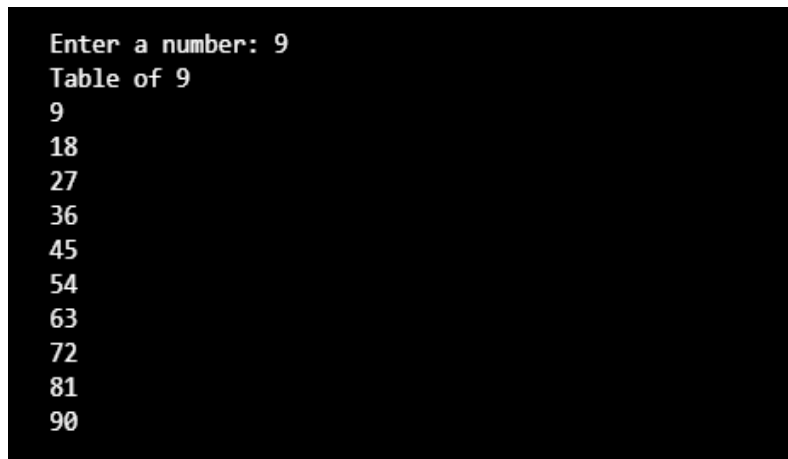
```
// program to print table of any number

#include <iostream.h>
void main()
{
    int num, i;
    cout<<"Enter a number: ";
    cin>>num;
```

```

cout<<"Table of "<<num<<endl;
for(i=1;i<=10;i++)
{
cout<<num*i<<endl;
}
}

```

Output:


```

Enter a number: 9
Table of 9
9
18
27
36
45
54
63
72
81
90

```

7. Write a program to print Fibonacci series (0, 1, 1, 2, 3, 5, 8, 13, 21...).

```

// Program to print Fibonacci Series using for loop
#include <iostream.h>
void main()
{
    int num, i, a=0, b=1, c;
    cout<<"Enter a number of terms for Series: ";
    cin>>num;
    cout<<"Fibonacci series : \n";
    for(i=0; i<num; i++)
    {
        cout<<"\n"<<a;
        c=a+b;
        a=b;
        b=c;
    }
}

```

NOTES

NOTES

```

Enter a number of terms for Series: 9
Fibonacci series :

0
1
1
2
3
5
8
13
21

```

8. Write a program to check whether a given number is Armstrong.

A number is known as Armstrong number if sum of the cubes of its digits is equal to the number itself.

For example: 370 is an Armstrong number because:

$$\begin{aligned}
 370 &= 3*3*3 + 7*7*7 + 0*0*0 \\
 &= 27 + 343 + 0 \\
 &= 370
 \end{aligned}$$

```

// C++Program to check Armstrong Number
#include <iostream.h>
void main()
{
    int num, sum = 0, rem,temp;
    cout<<"Enter a number: ";
    cin>>num;
    temp =num;
    while (num>0)
    {
        rem =num%10;
        sum =sum+(rem*rem*rem);
        num =num/10;
    }
    if (temp==sum)
    cout<<"Number is Armstrong "<<endl;
    else
    cout<<"Number is not Armstrong ."<<endl;
}

```

Output:

```
Enter a number: 370
Number is Armstrong
```

NOTES**9. Write a program to print table of any number using do while loop.**

```
//C++ program to print table of any number using do
while loop
#include <iostream.h>
void main()
{
    int num, i;
    cout << "Enter any number: ";
    cin >> num;
    cout<<"\n Table of" <<num<<endl;
    i=1;
    do{
        cout<<num*i<<endl;
        i++;
    }while(i<=10);
}
```

Output:

```
Enter any number: 12

Table of12
12
24
36
48
60
72
84
96
108
120
```


NOTES

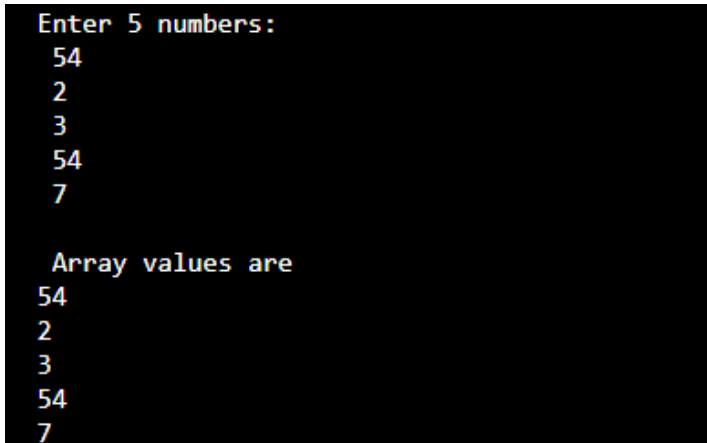
Try yourself:

- (i) Write a program to reverse a given number.
- (ii) Write a program to check whether a number is prime or not.
- (iii) Write a program to convert binary number to decimal number.

10. Write a program that takes values in an array and also display them.

```
//C++ program to scan and print values using array
#include <iostreamh>
int main()
{
    int arr[5],i;
    cout << "Enter 5 numbers:\n ";
    for(i=0;i<5;i++)
    cin >> arr[i];
    cout<<"\n Array values are "<<endl;
    for(i=0;i<5;i++)
    cout<<arr[i]<<endl;
}
```

Output:



```
Enter 5 numbers:
54
2
3
54
7

Array values are
54
2
3
54
7
```

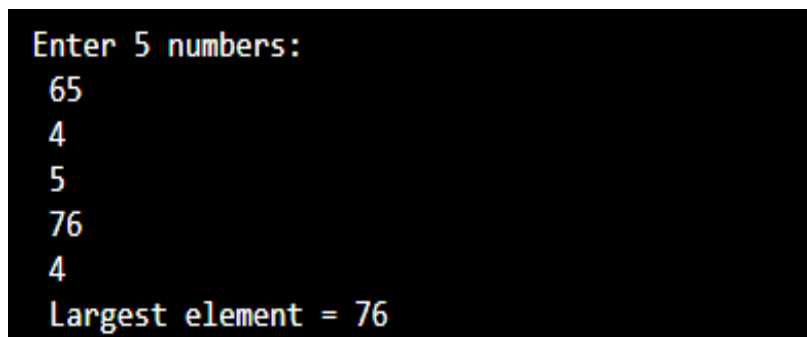
11. Write a program to print the largest value in an array.

```
//C++ program to print the largest value in an array
#include <iostream.h>
int main()
{
    int arr[5],i,max;
    cout<<"Enter 5 numbers:\n ";
```

```

for(i=0;i<5;i++)
    cin>>arr[i];
    max=arr[0];
for(i = 1;i < 5; i++)
{
    if(max < arr[i])
        max = arr[i];
}
cout<<"Largest element"<<max;
}

```

Output:


```

Enter 5 numbers:
65
4
5
76
4
Largest element = 76

```

NOTES

12. Write a program to search an element in the array using binary search.

Algorithm: Binary Search

INPUT : SORTED LIST OF SIZE N, KEY VALUE KEY

OUTPUT : POSITION OF KEY IN THE LIST = KEY

1. BEGIN

2. [INITIALIZE]

SET MAX := SIZE

SET MIN := 1

SET FOUND := FALSE

3. WHILE (FOUND IS FALSE AND MAX \geq MIN)

SET MID := (MAX + MIN)/2

4. IF KEY = LIST [MID] THEN

SET I := MID

SET FOUND := TRUE

NOTES

```
EXIT
ELSE IF KEY < LIST [MID] THEN
SET MAX := MID -1
ELSE
SET MIN := MID +1
[END OF IF]
[END OF LOOP]
5. IF FOUND = FALSE THEN
WRITE : VALUE IS NOT IN LIST
ELSE
WRITE VALUE FOUND AT MID LOCATION
6. END
```

```
//C++ program for binary Search
#include <iostream.h>
// Binary Search Function
void binary_search (int a[ ] , int size , int key)
{
int low ,high ,mid ,flag ;
flag= 0;
low = 0;
high = size -1;
while (low <= high && flag ==0)
{
mid =(low +high)/2;
if ( key == a [mid])
{
flag=1;
break;
}
else if (key < a[mid ] )
{
high = mid -1;
}
else
{
low = mid +1;
}
```

```
}
}
if ( flag ==1)
{
cout<<"value found at  location"<<mid +1;
}
else
cout<<"value not found";
}

void main()
{
int arr[10],i,k;
cout<<"Enter 10 values\n";
for(i=0;i<10;i++)
cin>>arr[i];
cout<<"Enter  value to be searched ";
cin>>k;
//call of binary_Search function
binary_search(arr,10,k);
}
```

Output:



```
Enter 10 values
10
20
30
40
50
60
70
80
90
100
Enter  value to be searched 60
value found at  location6
```

NOTES

13. Write a program to sort an array using selection sort.

```
//C++ program for selection sort
#include <iostream.h>
void selection_sort (int a[ ], int size )
{
```

NOTES

```
int temp ,i,j, min;

    for(int i = 0; i < size-1 ; i++)
    {
        min = i ; //considering element i as minimum

        for(int j = i+1; j < size ; j++ )
        {
            if(a[ j ] < a[ min ])
            {
                min = j ;
            }
        }

        temp= a[ min ];
        a[ min ]=a[ i ] ;
        a[ i]=temp;

    }

}

//main function
void main()
{

int arr[10],i;
cout<<"Enter 10 values\n";
for(i=0;i<10;i++)
cin>>arr[i];
//call of selection sort function
selection_sort(arr,10);
cout<<" \n Sorted Values \n";
for(i=0;i<10;i++)
cout<<endl<<arr[i];
}
```

Output:

Lab: Programming in C++

```
Enter 10 values
99
8
67
5
6
34
78
1
2
43

Sorted Values

1
2
5
6
8
34
43
67
78
99
```

NOTES

14. Write a program for bubble sort.

Algorithm: Bubble Sort

INPUT : LIST [] OF N ITEMS

OUTPUT : LIST [] OF N ITEMS IN SORTED ORDER

1. BEGIN
2. FOR I=1 THROUGH N DO
3. FOR J=N THROUGH I+1 DO
4. IF $A[J] < A[J-1]$ THEN
SET TEMP = A[J]
SET A[J] = A[J-1]
SET A[J-1] = TEMP
5. END

```
//C++ program for bubble sort
#include <iostream.h>
void bubble_sort (int a[ ], int size )
{
    int temp ,i,j;
    for(i=0; i<size; i++)
    {
        for(j=0; j<size-1; j++)
        {
```

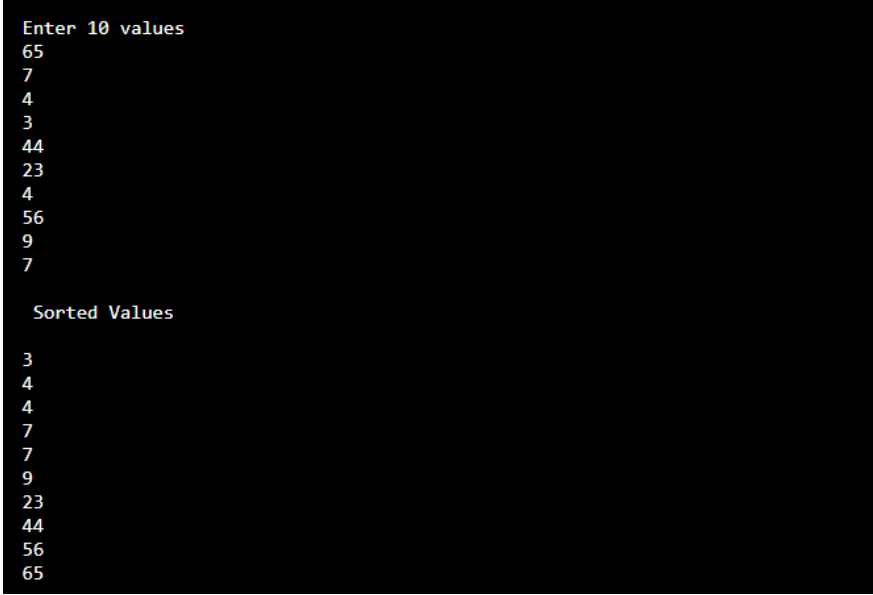
NOTES

```
        if(a[j]>a[j+1])
        {
            temp=a[j];
            a[j]=a[j+1];
            a[j+1]=temp;
        }
    }
}
//main function

void main()
{
    int arr[10],i;
    cout<<"Enter 10 values\n";
    for(i=0;i<10;i++)
        cin>>arr[i];

    //call of bubble sort function
    bubble_sort(arr,10);
    cout<<" \n Sorted Values \n";
    for(i=0;i<10;i++)
        cout<<endl<<arr[i];
}
```

Output:



```
Enter 10 values
65
7
4
3
44
23
4
56
9
7

Sorted Values

3
4
4
7
7
9
23
44
56
65
```

15. Write a program for quick sort.**Algorithm:** Quick Sort

QUICK_SORT (ARRAY, FIRST, LAST)

1. SET LOW: = FIRST
SET HIGH: = LAST
SET PIVOT: =ARRAY [(LOW + HIGH) /2]
2. REPEAT THROUGH STEP 7 WHILE (LOW ≤HIGH)
3. REPEAT STEP 4 WHILE (ARRAY [LOW] <PIVOT)
4. SET LOW: = LOW+1
5. REPEAT STEP 6 WHILE (ARRAY [HIGH]>PIVOT)
6. SET HIGH: = HIGH-1
7. IF (LOW ≤HIGH)
ARRAY [LOW] <->ARRAY [HIGH]
SET LOW: = LOW+1
SET HIGH:= HIGH-1
8. IF (FIRST<HIGH) THEN
QUICK_SORT (ARRAY, FIRST, HIGH)
9. IF (LOW < LAST)
QUICK_SORT (ARRAY, LOW, LAST)
10. END

```
//C++ program for quick sort

#include <iostream.h>
void quick_sort (int a[ ], int first, int last)
{
    int low ,high ,pivot, temp, i ;
    low= first ;
    high =last ;
    pivot =a[(first +last)/2];

    do
    {
        while (a[low]<pivot)
        {
            low++;
        }
    }
}
```

NOTES

NOTES

```
while (a [high]>pivot)
{
high--;
}
if(low <=high)
{
temp= a [low];
a [low]= a[high];
a[high]= temp ;
low++;
high--;
}
}while (low <=high);

if (first <high)
{
quick_sort (a, first, high);
}
if(low< last)
{
quick_sort (a, low, last);
}
}

void main()
{

int arr[10],i,k;
cout<<"Enter 10 values\n";
for(i=0;i<10;i++)
cin>>arr[i];
//call of Quick Sort function
quick_sort(arr,0,10);
cout<<" \n Sorted Values \n";
for (i=0;i<10;i++)
cout<<endl<<arr[i];
}
```

Output:

Lab: Programming in C++

```
Enter 10 values
3
4
56
7
23
5
77
5
1
7

Sorted Values
1
3
4
5
5
7
7
23
56
77
```

NOTES

16. Write a program for merge sort.

Algorithm: Two-Way Merge Sort

TWO_WAY_MERGE_SORT (LIST, START, FINISH)

1. [COMPUTE THE SIZE OF CURRENT SUB-TABLE]
SET SIZE := FINISH - START+1
2. [TEST BASE CONDITION FOR SUB-TBLE OF SIZE ONE]
IF SIZE <=1 THEN
RETURN
3. [CALCULATE MID POINT POSITION OF CURRENT SUB-TABLE]
SET MID :=START + SIZE /2 -1
4. [RECURSIVELY SORT THE FIRST SUB-TABLE]
CALL TWO_WAY_MERGE_SORT (LIST, START, MID)
5. [RECURSIVELY SORT THE SECOND SUB-TABLE]
CALL TWO_WAY_MERGE_SORT (LIST, MID + 1 , FINISH)
6. CALL SIMPLE_MERGE (LIST, START, MID+1, FINISH)
7. RETURN

```
//C++ program for merge sort
```

```
#include <iostream.h>
```

NOTES

```
// function to merge the two half into a sorted data.
void merge_array(int a[], int low, int high, int mid)
{
    // low to mid and mid+1 to high array are already
sorted
    int i, j, k;
    int temp_arr[high-low+1];
    i = low;
    k = 0;
    j = mid + 1;

    while (i <= mid && j <= high)
// merging of two parts into temp array
    {
        if (a[i] < a[j])
        {
            temp_arr[k] = a[i];
            k++;
            i++;
        }
        else
        {
            temp_arr[k] = a[j];
            k++;
            j++;
        }
    }

    while (i <= mid)
// insertion of remaining values from i to mid into
temp array.
    {
        temp_arr[k] = a[i];
        k++;
        i++;
    }

    while (j <= high)
// insertion of remaining values from j to high
into temp array.
```

```

    {
        temp_arr[k] = a[j];
        k++;
        j++;
    }

    // assign sorted data stored in temp array to a
    array.
    for (i = low; i <= high; i++)
    {
        a[i] = temp_arr[i-low];
    }
}

// A function to split array into two parts.
void merge_sort(int a[], int low, int high)
{
    int mid;
    if (low < high)
    {
        mid=(low+high)/2;
        // split array into two parts
        merge_sort(a, low, mid);
        merge_sort(a, mid+1, high);
        // merge arraythem to get sorted values
        merge_array(a, low, high, mid);
    }
}

void main()
{
    int arr[10],i,k;
    cout<<"Enter 10 values\n";
    for(i=0;i<10;i++)
    cin>>arr[i];
    //call of merge sort function
    merge_sort(arr, 0, 9);
    cout<<" \n Sorted Values  \n";
    for(i=0;i<10;i++)
    cout<<endl<<arr[i];
}

```

NOTES

NOTES**Enter 10 values**88
6
5
7
45
3
4
6
7
1**Sorted Values**1
3
4
5
6
6
7
7
45
88**Try Yourself:**

- (i) Write a program to sort n numbers in descending order using bubble sort.
- (ii) Write a program to implement selection sort method using functions.
- (iii) Write a program to sort the n names in an alphabetical order.

17. Write a Program that takes string as input and print it.

//C++ program to take string as input and print it

```
#include <iostream.h>
#include <conio.h>
void main()
{
char str[15];
cout<<"Enter your name: ";
cin>>str;
cout<<"\nWelcome "<<str;
getch();
}
```

Output:

```
Enter your name: Rajan
```

```
Welcome Rajan
```

NOTES**18. Write a program to print the length of a given string without using string function.**

```
//C++ program to count string length
#include<iostream.h>
void main( )
{
    int i, count=0;
    char str[50];
    cout<<"Enter any string ";
    cin.getline(str, 50);
    //getline function allows user to input string
    with space
    //loop will run till it reaches to string
    terminator '\0'
    for(i = 0; str[i] != '\0'; i++)
    {
        count++;
    }
    cout << "\n Length of string is " << count;
}
```

Output:

```
Enter any string programming
```

```
Length of string is 11
```

19. Write a program to check whether a given string is palindrome or not.

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

NOTES

```
int i, len=0;
char str[50], rev_str[50];
cout<<"Enter any string ";
cin.getline(str, 50); //getline function allows
                        user to input string with
                        space

                        //count length of string
for(i = 0; str[i] != '\0'; i++)
{
    len++;
}
cout << "\n Length of string is" << len;
                        //copy str to rev_str
int j=0;
for (i = len - 1; i >= 0 ; i--,j++)
{
    rev_str[j] = str[i];
}
rev_str[j] = '\0'; //reverse string is terminated

//compare both strings
int flag=0;
for (i = 0; i < len ; i++)
{
    if (str[i]==rev_str[i])
        flag = 1;
    else
    {
        break; //exit from loop
    }
}

if (flag == 1)
    cout<<" \n string is a palindrome";
    else
    cout<<" \n string is a not palindrome";
}
```

Output:

```
Enter any string nitin
string is a palindrome
```

NOTES

Try yourself:

- (i) Write a program to insert an element in an array.
- (ii) Write a program to find sum of elements of an array.
- (iii) Write a program to find largest number from an array.

20. Write a program to print sum of two matrices.

```
//C++ program to print sum of two matrices
#include<iostream.h>
int main()
{
    int i,j, m1[10][10], m2[10][10], sum[10][10];
    cout << "Enter the elements of first matrix\n";
    for ( i = 0 ;i < 3 ; i++ )
    {
        cout<<"\n enter values for row " <<i+1<<endl;
        for ( j = 0 ; j<3 ; j++ )
        {
            cin >> m1[i][j];}
    }

    cout << "Enter the elements of second matrix\n";

    for ( i = 0 ;i < 3; i++ )
    {
        cout<<"\n enter values for row " <<i+1<<endl;
        for ( j = 0 ; j< 3 ; j++ )
        {
            cin >> m2[i][j];
        }
    }
    cout << "Sum of two matrices \n";
    for ( i = 0 ;i < 3 ; i++ )
    {
```



```
        for ( j = 0 ; j<3 ; j++ )
        {
            sum[i][j] = m1[i][j]+m2[i][j];
            cout << sum[i][j] << "\t";
        }
        cout<<endl;
    }
}
```

NOTES

Output:

```
Enter the elements of first matrix

enter values for row 1
1 2 3

enter values for row 2
2 3 4

enter values for row 3
2 3 1
Enter the elements of second matrix

enter values for row 1
5 4 3
enter values for row 2
2 3 4

enter values for row 3
4 5 6
Sum of two matrices
6 6 6
4 6 8
6 8 7
```

21. Write a program to find out the product of two matrices.

```
//C++ program for matrix multiplication
#include<iostream.h>

int main()
{
    int i, j, k, m1[10][10], m2[10][10], res[10][10];
    cout << "Enter the elements of first matrix\n";
    for ( i = 0 ;i < 3 ; i++ )
    {
        cout<<"\n enter values for row "<<i+1<<endl;
```

```

        for ( j = 0 ; j<3 ; j++ )
        {
            cin >> m1[i][j];
        }

cout << "Enter the elements of second matrix\n";

for ( i = 0 ;i < 3; i++ )
{
    cout<<"\n enter values for row "<<i+1<<endl;
    for ( j = 0 ; j< 3 ; j++ )
    {
        cin >> m2[i][j];
    }
}

for (i = 0; i < 3; ++i)
{
    for (j = 0; j < 3; ++j)
    {
        res [i][j]=0;

        for (k = 0; k < 3; ++k)
        {
            res [i][j] += m1[i][k] * m2[k][j];
        }
    }
}

cout << "Multiplication of two matrices \n";
for ( i = 0 ;i < 3 ; i++ )
{
    for ( j = 0 ; j<3 ; j++ )
    {
        cout << res[i][j] << "\t";

    }
    cout<<endl;
}
}

```

NOTES

NOTES

```

Enter the elements of first matrix

enter values for row 1
1 2 3

enter values for row 2
1 2 3

enter values for row 3
1 2 3
Enter the elements of second matrix

enter values for row 1
2 3 4

enter values for row 2
2 3 4

enter values for row 3
2 3 4
Multiplication of two matrices
12 18 24
12 18 24
12 18 24

```

Try yourself:

- (i) Write a program to print sum of diagonal values of a square Matrix.
- (ii) Write a program to find highest and lowest element of a Matrix.
- (iii) Write a program to convert first letter of each word of a string to uppercase and other to lowercase.
- (iv) Write a program to find substring in string (Pattern Matching).

22. Write a program to print factorial of a given number using user defined function.

```

#include <iostream.h>
int fact(int n);
int main()
{
    int n;
    cout << "Enter any number ";
    cin >> n;

```

```

        cout << "Factorial of " << n << " = " << fact(n);
        return 0;
    }

```

Lab: Programming in C++

```

int fact(int n)
{
    if(n > 1)
        return n * fact(n - 1);
    else
        return 1;
}

```

NOTES

Output:

```

Enter any number 6
Factorial of 6 = 720

```

23. Write a program to check a year is leap year or not using function.

```

// Program to check leap year or not

#include <iostream.h>
bool leapYear (int y);

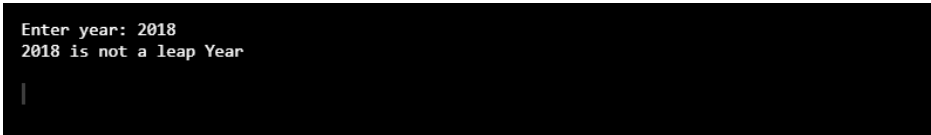
int main()
{
    int y;
    cout<<"Enter year: ";
    cin>>y;
    //Calling function
    bool flag = leapYear(y);
    if(flag == true)
        cout<<y<<" is a leap Year";
    else
        cout<<y<<" is not a leap Year";
    return 0;
}
bool leapYear(int y)
{

```

NOTES

```
bool flag = false;
if (y % 4 == 0)
{
    if (y % 100 == 0)
    {
        if (y % 400 == 0)
        {
            flag = true;
        }
    }
    else flag = true;
}
return flag;
}
```

Output:



```
Enter year: 2018
2018 is not a leap Year
|
```

24. Write a program to print array elements using user defined function.

```
#include <iostream.h>

void display(int arr[5]);
int main()
{
    int arr[5] = { 101, 201, 301, 401, 501 };
    //passing array to function
    display(arr);
}

void display(int arr[5])
{
    cout << "Display array values:"<< endl;
    for (int i = 0; i < 5; i++)
    {
        cout<<arr[i]<<"\n";
    }
}
```

Output:

Lab: Programming in C++

```
Display array values:
101
201
301
401
501
```

NOTES

25. Write a program for sequential search using user defined function.

Algorithm: Sequential Search

INPUT : LIST OF SIZE N, TARGET VALUE T

OUTPUT : POSITION OF T IN THE LIST

1. BEGIN
2. SET FOUND: = FALSE
 SET I: = 0
3. WHILE $I \leq N$ AND FOUND IS FALSE
 IF LIST [I] = T THEN
 SET FOUND: = TRUE
 EXIT
 ELSE
 SET I: =I+1
 [END OF STEP 3 LOOP]
4. IF FOUND = FALSE THEN
 WRITE: T IS NOT IN LIST
 ELSE
 WRITE: T IS FOUND AT I LOCATION
 [END OF IF]
5. END

```
//C++ program for sequential search

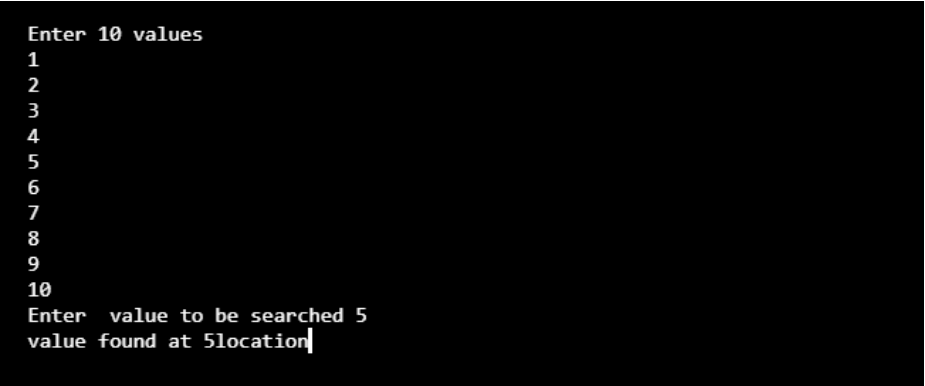
#include <iostream.h>
//definition of sequential Search function
```

NOTES

```
void sequential_search (int a[ ],int size ,int key)
{
    int flag , i ;
    flag =0;
    for ( i=0 ; i<size ; i++)
    {
        if ( a [i] == key )
        {
            flag = 1 ;
            break ;
        }
    }
    if ( flag == 1)
        cout<<"value found at "<<i+1<<" location";
    else
        cout<<"value not found";
}

void main()
{
    int arr[10],i,k;
    cout<<"Enter 10 values";
    for(i=0;i<10;i++)
    cin>>arr[i];
    cout<<"Enter  values to be searched";
    cin>>k;
    //call of sequential_search function
    sequential_search(arr,10,k);
}
```

Output:



```
Enter 10 values
1
2
3
4
5
6
7
8
9
10
Enter value to be searched 5
value found at 5location
```

26. Write a program to print factorial of a number using recursive function.

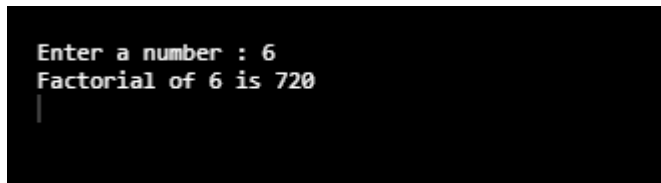
Lab: Programming in C++

```
//C++ Program to print factorial using recursive
function
#include<iostream.h>
// Factorial Function
int factorial(int n)
{
    if(n > 1)
        return n * factorial(n - 1);
//recursive call of factorial function
    else
        return 1;
}

int main()
{
    int n;
    cout << "Enter a number : ";
    cin >> n;
    cout << "Factorial of " << n << " is " << factorial(n);
    return 0;
}
```

NOTES

Output:



27. Write a program to print Fibonacci series using recursive function.

```
//C++ Program to print Fibonacci series using recursive
function
#include<iostream.h>
int Fibonacci(int n)
{
    if ((n==1) || (n==0))
    {
        return (n);
    }
}
```


NOTES

```

else
{

return (Fibonacci(n-1)+Fibonacci(n-2));
//recursive call of Fibonacci function
}
}
int main()
{
int n,i;

cout<<"Enter number of terms for Fibonacci Series:";
cin>>n;
cout<<"Fibonacci Series "<<endl;

for (i=0; i< n;i++)
{
cout<<" "<<Fibonacci(i);
}
return 0;
}

```

Output:

```

Enter number of terms for Fibonacci Series:12
Fibonacci Series
0 1 1 2 3 5 8 13 21 34 55 89

```

INLINE FUNCTION

1. We must keep inline functions small, small inline functions have better efficiency.
2. Inline functions do increase efficiency, but we should not make all the functions inline. Because if we make large functions inline, it may lead to **code bloat**, and might affect the speed too.
3. Hence, it is advised to define large functions outside the class definition using scope resolution ::operator, because if we define such functions inside class definition, then they become inline automatically.

4. Inline functions are kept in the Symbol Table by the compiler, and all the call for such functions is taken care at compile time.

Lab: Programming in C++

28. Write a program to demonstrate the concept of inline function.

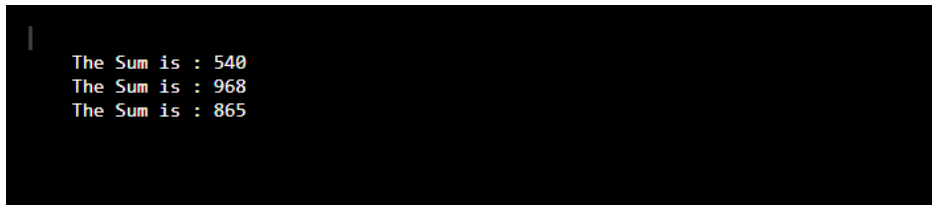
```
#include <iostream.h>

inline int sum(int x,int y)
{
    return x+y;
}

int main()
{
    cout<<"\n\tThe Sum is : " << sum(310,230);
    cout<<"\n\tThe Sum is : " << sum(145,823);
    cout<<"\n\tThe Sum is : " << sum(427,438);
}
```

NOTES

Output:



Try yourself:

- (i) Write a C++ program to find average marks of three subjects of N students in a class.
- (ii) Write a C++ program to take input of two distances in inch-feet system and stores in data members of two structure variables. Also, this program calculates the sum of two distances and displays it.
- (iii) Write a C++ program in which user is asked to enter two time periods and these two periods are stored in structure variables. The program calculates the difference between these two time periods.
- (iv) Write a C++ Program to find total salary of N employees in a department where DA=35% basic_salary and HRA=10% of basic_salary.

29. Write a program to demonstrate the concept of class and object.**NOTES**

```
//C++ sample program for class and object
#include<iostream.h>
//class
class student
{
    private:           //scope of variables is private
                     //member variables

    int rno;
    char name[10];

    public:           //scope of functions is public
                     // member functions
void input()
{
    cout<<"\n Enter student roll number :";
    cin>>rno;
    cout<<"\n Enter student name :";
    cin>>name;
}

void display()
{
    cout<<"\n Roll Number :"<<rno;
    cout<<"\n Name :"<<name;
}
}; //class closed

int main()
{
    student obj;           //object of student class
    obj.input();           //call of input function
    obj.display();         //call of display function
}
```

Output:

```
Enter student roll number :101
Enter student name :dhruv

Roll Number :101
Name :dhruv
```

30. Write a program that shows the use of *this* pointer.

```

#include <iostream.h>
class Demo
{
private:
    int num;
    char ch;
public:
    void setvalue(int num, char ch)
    {
        this->num =num;
        this->ch=ch;
    }
    void putvalue()
    {
        cout<<num<<endl;
        cout<<ch;
    }
};
int main()
{
    Demo obj;
    obj.setvalue(450, 'A');
    obj.putvalue();

}

```

Output:

```

450
A

```

NOTES**31. Write a program using static variable and static function.**

```

//C++ sample program for static variable and static
function
#include <iostream.h>
class test
{
private:

```

NOTES

```
static int count; //Static data
int n;
public:
//Constructor
test()
{
    count=count+1;
    n=count;
}

//static function
static void function1()
{
    cout << "\nResult is: " << count<<endl;
}
//Normal function
void counter()
{
    cout << "\n Counter is: " << n<<endl;
}

//Destructor
~test ()
{
    count =count-1;
}
};
int test::count=0;
int main()
{
    test obj1;
    //Static function is accessed using class name and
    scope resolution operator (::)
    test::function1();
    test obj2,obj3,obj4;
    test::function1();
    //normal function is accessed using object name
    and the dot member access operator(.)
    obj1.counter();
    obj2.counter();
```

```
obj3.counter();  
obj4.counter();  
}
```

Lab: Programming in C++

Output:

```
Result is: 1  
Result is: 4  
Counter is: 1  
Counter is: 2  
Counter is: 3  
Counter is: 4
```

NOTES

32. Write a program using static class and variable.

//C++ program to count the object value using the keyword static variable

```
#include<iostream.h>  
class static_class  
{  
    int n;  
    static int count; //static variable  
  
public:  
    //constructor  
    static_class ()  
    {  
        n = ++count;  
    }  
  
    void obj_number()  
    {  
        cout << "\n\t Object number is :" << n;  
    }  
    static void obj_count()  
    {  
        cout << "\n Number of Objects :" << count;  
    }  
}
```

};

int static_class::count;

NOTES

```
int main()
{
    static_class obj1, obj2;
    obj1.obj_count();
    obj1.obj_number();
    obj2.obj_count();
    obj2.obj_number();
    return 0;
}
```

Output:

```
Number of Objects :2
Object number is :1
Number of Objects :2
Object number is :2
```

33. Write a C++ program using constructor in a class.

```
//C++ sample program for constructor
#include<iostream.h>

//class

class student
{
    private:          //scope of variables is private
    //member variables
    int rno;
    char name[10];

    public:          //scope of functions is public
    student()
    {
        cout<<"Constructor \n";
    }
}
```

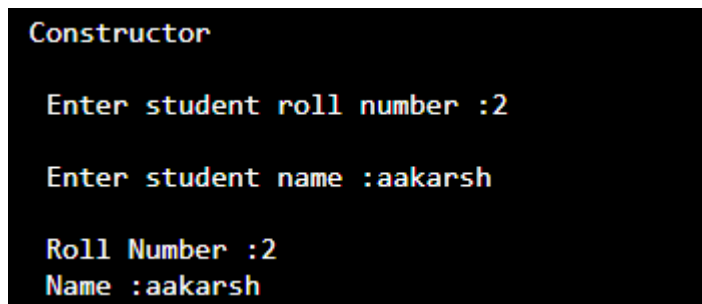
```
        rno=0;
    }

// member functions
void input()
{
    cout<<"\n Enter student roll number :";
    cin>>rno;
    cout<<"\n Enter student name :";
    cin>>name;
}

void display()
{
    cout<<"\n Roll Number :"<<rno;
    cout<<"\n Name :"<<name;
}

};

int main()
{
    student obj;
    obj.input ();
    obj.display ();
}
```

Output:

```
Constructor
Enter student roll number :2
Enter student name :aakarsh
Roll Number :2
Name :aakarsh
```

34. Write a program to demonstrate the use of constructor and destructor.

```
//C++ sample program for constructor and destructor
#include<iostream.h>
```

NOTES

NOTES

```
//class

class student
{
    private:
        //member variables
        int rno;
        char name[10];
    public

        // constructor
        student()
        {
            cout<<"Constructor \n";
            rno=0;
        }

        // member functions
        void input()
        {
            cout<<"\n Enter student roll number :";
            cin>>rno;
            cout<<"\n Enter student name :";
            cin>>name;
        }

        void display()
        {
            cout<<"\n Roll Number :"<<rno;
            cout<<"\n Name :"<<name;
        }

        //destructor
        ~student()
        {
            cout<<"\n Destructor \n";
        }
};
```

```

int main()
{
    student obj;
    obj.input();
    obj.display();
}

```

Lab: Programming in C++

NOTES

Output:

```

Constructor

Enter student roll number :101

Enter student name :Dhruv

Roll Number :101
Name :Dhruv
Destructor

```

35. Write a program to add two matrices. Create two objects of the class and each of which refers one 2D matrix. Use constructor to allocate memory dynamically and use copy constructor to allocate memory when one array object is used to initialize another.

```

#include <iostream.h>

class matrix
{
    int **a;
public:
    // Dynamic Constructor
    matrix()
    {
        int i,j;
        a=new int*[3];
        for(i=0; i<3; i++)
            a[i]=new int[3];
        cout<<"Enter elements for a 3x3 matrix:\n";
        for(i=0; i<3; i++)
            for(j=0; j<3; j++)
                cin>>a[i][j];
    }
    // Copy Constructor
    matrix(matrix & x)

```

NOTES

```
{
    int i,j;
    a=new int*[3];
    for(i=0; i<3; i++)
        a[i]=new int[3];
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
            a[i][j]=x.a[i][j];
}

// Destructor
~matrix()
{
    int i;
    for(i=0; i<3; i++)
        delete a[i];
    delete a;
}

void showdata()
{
    int i,j;
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            cout<<a[i][j]<<" ";
        }
        cout<<endl;
    }
}

friend void add(matrix,matrix);
};

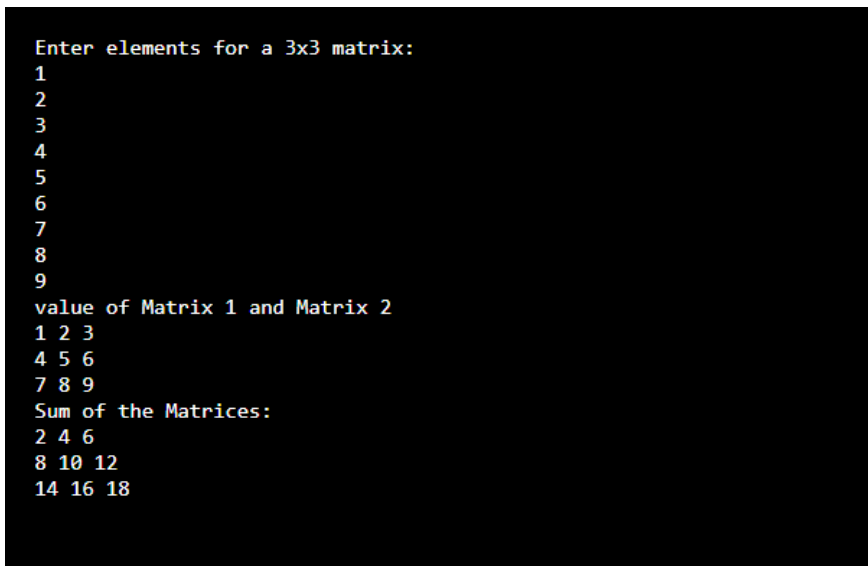
void add(matrix m1,matrix m2)
{
    int i,j;
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
```

```

        cout<<m1.a[i][j]+m2.a[i][j]<<" ";
    }
    cout<<endl;
}

//main function
int main()
{
    matrix obj1;
    matrix obj2(obj1);
    cout<<"value of Matrix 1 and Matrix 2\n";
    obj1.showdata ();
    cout<<"SUM of the Matrices:\n";
    add(obj1,obj2);
}

```

Output:


```

Enter elements for a 3x3 matrix:
1
2
3
4
5
6
7
8
9
value of Matrix 1 and Matrix 2
1 2 3
4 5 6
7 8 9
Sum of the Matrices:
2 4 6
8 10 12
14 16 18

```

36. Write a program to demonstrate the use of dynamic constructor.

```

#include <iostream.h>

class dyncons
{
private:
int *p;
public:

```

NOTES

NOTES

```
dyncons ()
{
p=new int;
*p=100;
}
dyncons (int v)
{
p= new int;
*p=v;
}
int dis()
{
return (*p);
}};

int main()
{
dyncons obj1,obj2(50);
cout<<"the value of object obj1 p is ";
cout<<obj1.dis();
cout<<"\n the value of object of obj2 p is:"<<obj2.
dis();

}
```

Output:

```
the value of object obj1 p is 100
the value of object of obj2 p is:50
```

37. Write a program using static variable.

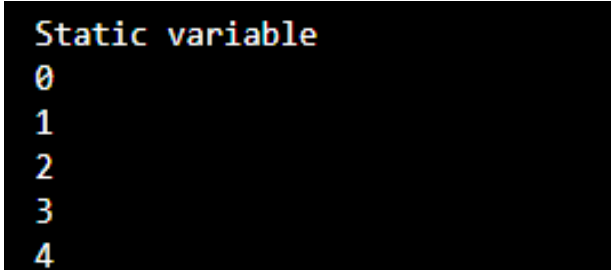
```
//C++ program for static variable
#include <iostream.h>
void test()
{
    // static variable
    static int count = 0;
    cout << count <<endl;
```

```

        count++;
    }

    int main()
    {
        cout << "Static variable " <<endl;
        for (int i=0; i<5; i++)
            test();
    }

```

Output:


```

Static variable
0
1
2
3
4

```

Try yourself:

- (i) Write a program to swap two numbers using class.
- (ii) Write a program to print numbers from 1 to N using class.
- (iii) Write a program to calculate area of a circle, a rectangle or a triangle depending on input using overloaded calculate function.

38. Write a program to illustrate the concept of function overloading on sum function.

```

#include <iostream.h>

class Test
{
    public:

    int sum(int a,int b)
    {
        return a + b;
    }
    int sum (int a, int b, int c)
    {

```

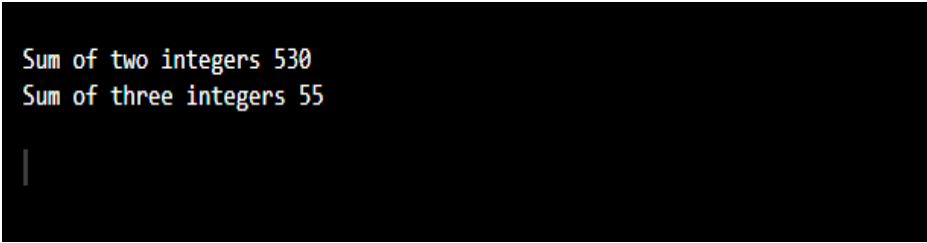
NOTES

```
        return a + b + c;  
    }
```

NOTES

```
};  
int main()  
{  
    Test obj;  
    cout<<"Sum of two integers "<<obj.sum(310,  
    220)<<endl;  
    cout<<"Sum of three integers "<<obj.sum(12, 20, 23);  
  
}
```

Output:

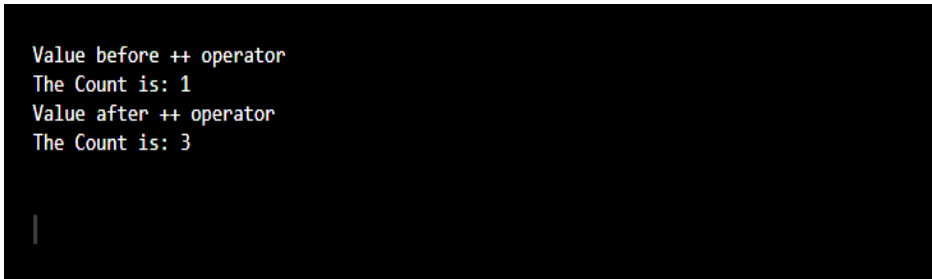


```
Sum of two integers 530  
Sum of three integers 55
```

39. Write a program to overload ++ operator.

```
#include <iostream.h>  
class Test  
{  
    private:  
        int num;  
    public:  
        Test ()  
        {  
            num=1;  
        }  
        void operator ++()  
        {  
            num = num+2;  
        }  
        void display()  
        {  
            cout<<"The Count is: "<<num;  
        }  
}
```

```
};
int main()
{
    Test obj;
    cout<<"Value before ++ operator \n";
    obj.display ();
    ++obj; // calling of operator void operator ++()
    cout<<"\nValue after ++ operator \n";
    obj.display ();
}
```

NOTES**Output:**


```
Value before ++ operator
The Count is: 1
Value after ++ operator
The Count is: 3
```

40. Write a program to demonstrate the overloading of + operators.

```
#include <iostream.h>
class overloading
{
    int value;
public:
    void setValue(int temp)
    {
        value = temp;
    }
    overloading operator+(overloading ob)
    {
        overloading t;
        t.value=value+ob.value;
        return(t);
    }
    void display()
    {
```

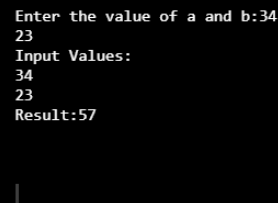


```
cout<<value<<endl;  
}  
};
```

NOTES

```
int main()  
{  
overloading obj1,obj2,result;  
int a,b;  
cout<<"Enter the value of a and b:";  
cin>>a>>b;  
obj1.setValue(a);  
obj2.setValue(b);  
result = obj1+obj2;  
cout<<"Input Values:\n";  
obj1.display();  
obj2.display();  
cout<<"Result:";  
result.display();  
  
}
```

Output:



```
Enter the value of a and b:34  
23  
Input Values:  
34  
23  
Result:57
```

41. Write a program to demonstrate the overloading of binary arithmetic operators (+, -, * and /).

```
#include <iostream.h>  
  
class arithmetic  
{  
float n;  
public:  
void get()  
{  
cout<<"\n enter number:\n";  
cin>>n;
```

```
}
arithmetic operator +( arithmetic &a)
{
arithmetic t;
t.n=n+a.n;
return t;
}
arithmetic operator -( arithmetic &a)
{
arithmetic t;
t.n=n-a.n;
return t;
}
arithmetic operator *( arithmetic &a)
{
arithmetic t;
t.n=n*a.n;
return t;
}
arithmetic operator /( arithmetic &a)
{
arithmetic t;
t.n=n/a.n;
return t;
}
void display()
{
cout<<n;
}
};

int main()
{
arithmetic a1,a2,a3;

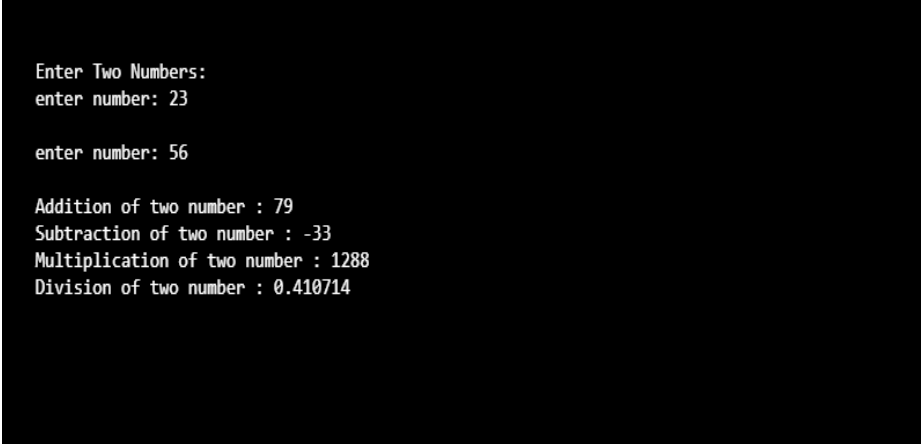
a1.get();
a2.get();
a3 = a1+a2;
cout<<"\n Addition of two number:";
a3.display();
```

NOTES

NOTES

```
a3 = a1-a2;
cout<<"\n Subtraction of two number:";
a3.display();
a3 = a1*a2;
cout<<"\n Multiplication of two number:";
a3.display();
a3 = a1/a2;
cout<<"\n Division of two number:";
a3.display();
}
```

Output:



```
Enter Two Numbers:
enter number: 23

enter number: 56

Addition of two number : 79
Subtraction of two number : -33
Multiplication of two number : 1288
Division of two number : 0.410714
```

42. Write a program to get and print student data using single inheritance.

```
// program to get and print student data using
inheritance
#include <iostream.h>

//class
class student
{
    private:          //scope of variables is private
                    //member variables

    int rno;
    char name[10];

    public:          //scope of functions is public
                    // member functions
```

```

void input()
{
    cout<<"\n Enter student roll number :";
    cin>>rno;
    cout<<"\n Enter student name :";
    cin>>name;
}

void display()
{
    cout<<"\n Roll Number :"<<rno;
    cout<<"\n Name :"<<name;
}
}; //class closed

class fee:public student
    //class fee(derived) class is
    inheriting student (base) class
{
    float fee; //default scope in private

    public:
    void input_data()
    {
        input();
        //call of input function of
        student class
    }

    cout<<"\n Enter Fee :";
    cin>>fee;
}

    void display_data()
    {
        //call of display function of
        student class

        display();
    }

    cout<<"\n Fee :"<<fee;
}
};

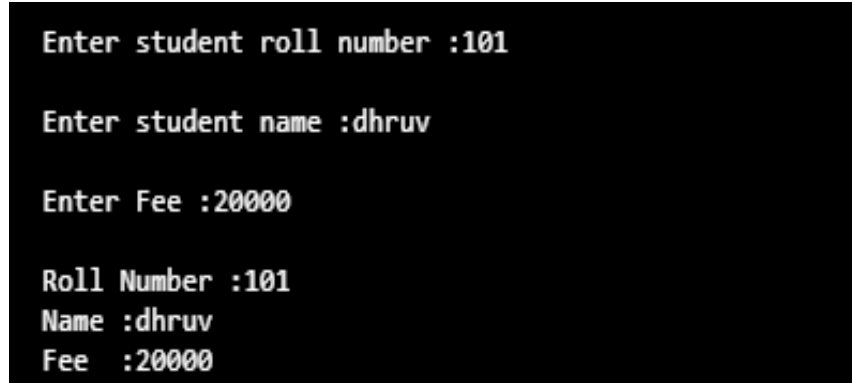
```

NOTES

NOTES

```
int main()
{
    fee obj;           //object of fee class
    obj.input_data();
    obj.display_data();
}
```

Output:



```
Enter student roll number :101

Enter student name :dhruv

Enter Fee :20000

Roll Number :101
Name :dhruv
Fee :20000
```

43. Write a program to demonstrate the concept of multiple inheritance.

```
#include <iostream.h>

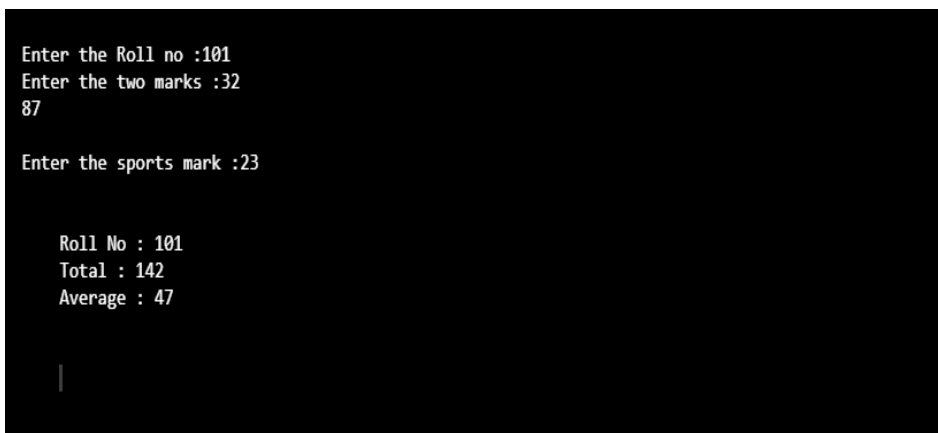
class student
{
protected:
int rno,m1,m2;
public:
void get()
{
cout<<"Enter the Roll no :";
cin>>rno;
cout<<"Enter the two marks :";
cin>>m1>>m2;
}
};

class sports
{
protected:
int sm; // sm = Sports mark
public:
void getsm()
```

```
{
cout<<"\nEnter the sports mark :";
cin>>sm;
}
};
class statement:public student,public sports
{
int tot,avg;
public:
void display()
{
tot=(m1+m2+sm);
avg=tot/3;
cout<<"\n\n\tRoll No : "<<rno<<"\n\tTotal : "<<tot;
cout<<"\n\tAverage : "<<avg;
}
};
int main()
{
statement obj;
obj.get();
obj.getsm();
obj.display();
}
```

NOTES

Output:



```
Enter the Roll no :101
Enter the two marks :32
87

Enter the sports mark :23

Roll No : 101
Total : 142
Average : 47
```

44. Write a program to demonstrate the concept of multilevel inheritance.**NOTES**

```
#include <iostream.h>
//base class
class top
{
public :
int a;
void getdata()
{
cout<<"\n\nEnter Any Number : ";
cin>>a;
}
void putdata()
{
cout<<"\nValue is:\t"<<a;
}
};

// class middle is derived_1
class middle :public top
{
public:
int b;
void square()
{
getdata();
b=a*a;
cout<<"\n\nSquare Is :"<<b;
}
};

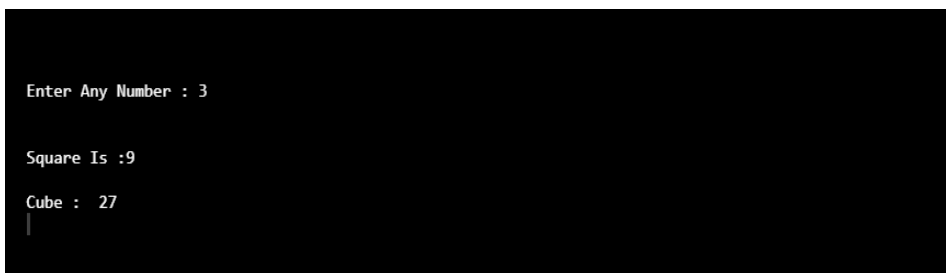
// class bottom is derived_2
class bottom :public middle
{
public:
int c;
void cube()
{
square();
}
```

```

c=b*a;
cout<<"\n\nCube : \t"<<c;
}
};

int main()
{
bottom b1;
b1.cube();
}

```

Output:


```

Enter Any Number : 3

Square Is :9

Cube : 27
|

```

Try Yourself:

- (i) Write a program to demonstrate the multilevel inheritance.
- (ii) Write a program to demonstrate the multiple inheritance.
- (iii) Write a program to demonstrate the virtual derivation of a class.

45. Write a program to demonstrate function overriding.

```

#include<iostream>
using namespace std;

//base class
class base
{
public:
    virtual void display()
    {
        cout << "\nThis is display method of base class";
    }
    void show()
    {

```

NOTES


```
        cout << "\nThis is show method of base class";  
    }  
};
```

NOTES

```
    //derived class  
class derived : public base  
{  
public:  
    // Overriding method - new working of  
    // base class's display method  
    void display()  
    {  
        cout << "\nThis is display method of derived  
                class";  
    }  
};  
  
// main function  
int main()  
{  
    derived dr;  
    base &bs = dr;  
    bs.display();  
    dr.show();  
}
```

Output:

```
This is display method of derived class  
This is show method of base class
```

46. Write a program using virtual function.

Lab: Programming in C++

```
#include <iostream.h>

class base
{
public:
    virtual void show()
    {
        cout << "\n Base class show:";
    }

    void display()
    {
        cout << "\n Base class display:";
    }
};

class drive : public base
{
public:

    void display()
    {
        cout << "\n Drive class display:";
    }

    void show()
    {
        cout << "\n Drive class show:";
    }
};

int main()
{
    base obj1;
    base *p;
    cout << "\n\t P object points to base:\n";
    p = &obj1;
    p->display();
    p->show();
}
```

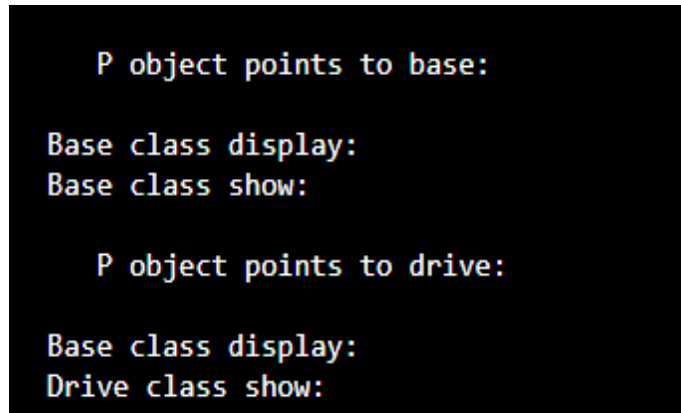
NOTES

NOTES

```

        cout << "\n\n\t P object points to drive:\n";
        drive obj2;
        p = &obj2;
        p->display();
        p->show();
    }

```

Output:


```

        P object points to base:

        Base class display:
        Base class show:

        P object points to drive:

        Base class display:
        Drive class show:

```

Pure Virtual Function

A virtual function will become pure virtual function when you append “=0” at the end of declaration of virtual function. Pure virtual function doesn’t have body or implementation. We must implement all pure virtual functions in derived class. Pure virtual function is also known as abstract function.

47. Write a program using pure virtual function/ abstract function.

```

#include <iostream.h>

class BaseClass      //Abstract class
{
public:
virtual void Display1()=0; //Pure virtual function
                        or abstract function
virtual void Display2()=0; //Pure virtual function or
                        abstract function

void Display3()
{
cout<<"\n\tThis is Display3() method of Base Class";
}
}

```

```

};

class DerivedClass : public BaseClass
{
public:
void Display1()
{
cout<<"\n\tThis is Display1() method of Derived
Class";
}

void Display2()
{
cout<<"\n\tThis is Display2() method of Derived
Class";
}
};

int main()
{
DerivedClass D;

D.Display1(); // This will invoke Display1()
method of Derived Class
D.Display2(); // This will invoke Display2()
method of Derived Class
D.Display3(); // This will invoke Display3()
method of Base Class

}

```

Output:

```

This is Display1() method of Derived Class
This is Display2() method of Derived Class
This is Display3() method of Base Class

```

NOTES

NOTES

Try yourself:

- (i) Write a program that overloads the + operator and relational operators (suitable) to perform the following operations:
 - (a) Concatenation of two strings. (b) Comparison of two strings.
- (ii) Write a programs functions to find the GCD of two given integers using pointer.

48. Write a C++ program to create file (data.txt).

```

/ /basic file operations
#include <iostream.h>
#include <fstream.h>
#include <conio.h>

void main ()
{
    ofstream file1;
    file1.open ("data.txt");
    file1 << "This is my first file.\n";
    file1.close();
    getch();
}

```

49. Write a program to create and write on a text file.

```

// C++ program of writing on a text file

#include<iostream.h>
#include<conio.h>
#include<fstream.h>

void main()
{
    ofstream file_out;
    char file_name[20];
    char str [80];
    clrscr ();

    cout<<"Enter file name to be created ";
    cin>> file_name;

    //create a new file in output mode

```

```
file_out.open (file_name, ios::out);
```

Lab: Programming in C++

```
cout<<"Enter data to be stored ";  
cin>> str;
```

```
file_out << str;  
cout<<"Information stored in file";
```

```
//close file  
file_out.close ();  
getch ();
```

```
}
```

50. Write a program to retrieve data from a text file.

```
// C++ program of retrieve data from a text file
```

```
#include<iostream.h>
```

```
#include<fstream.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    ifstream file_in;  
    char file_name [20];  
    char str[80];
```

```
    clrscr();
```

```
    cout<<"Enter file name: ";  
    cin>> file_name;
```

```
    cout<<"Enter file name to open";  
    file_in.open(file_name, ios::in);
```

```
    file_in.get(str, 80);  
    cout<<str;
```

```
    file_in.close();  
    getch();
```

```
}
```

NOTES

51. Write a program to read and write on a binary file.**NOTES**

```
#include<iostream.h>
#include<fstream.h>
#include<cstdio.h>

class Student
{
    int rno;
    char name[50];
public:
    void setData()
    {
        cout << "\n Enter roll number";
        cin >>rno;
        cout << "Enter name ";
        cin.getline (name,50);
    }

    void showData()
    {
        cout << "\n Admission no. : " << rno;
        cout << "\n Student Name : " << name;
    }
};

// function to write in a binary file.
void write_data ()
{
    ofstream file_out;
    file_out.open ("student.dat", ios::binary |
        ios::app);
    Student obj;
    obj.setData ();

    file_out.write ((char*)&obj, sizeof (obj));

    file_out.close ();
}

//function to display records of file
```

```
void display()
{
    ifstream file_in;
    file_in.open("student.dat", ios::binary);

    Student obj;

    while(file_in.read ((char*)&obj, sizeof (obj)))
    {
        obj.showData ();
    }

    file_in.close ();
}

};

int main()
{
    for(int i = 1; i <= 4; i++)
        write_record (); //Display all records
    cout << "\n List of records";
    display ();

        //Search record
    cout << "\n Search result";
    search (100);

        //Delete record
    delete_record (100);
    cout << "\n Record Deleted";

        //Modify record
    cout << "\n Modify Record 101 ";
    modify_record (101);

    return 0;
}
```

NOTES

NOTES

Try yourself:

(i) What task does the following program perform?

```
#include<iostream.h>
#include<fstream.h>

int main()
{
    ofstream ofile;
    ofile.open ("text.txt");
    ofile << "geeksforgeeks" << endl;
    cout << "Data written to file" << endl;
    ofile.close();
}
```

(ii) Write a program which copies one file to another.

(iii) Write a program to that counts the characters, lines and words in the text file.

B.C.A.
101 24
LAB: PROGRAMMING IN C++
II - Semester



ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category-I University by MHRD-UGC]

KARAIKUDI – 630 003

DIRECTORATE OF DISTANCE EDUCATION



ISBN 978-93-5338-365-7



9 789353 383657