SRI CHANDRASHEKARENDRA SARASWATHI

VISWA MAHAVIDYALAYA

(UNIVERSITY ESTABLISHED UNDER SECTION 3 OF UGC ACT 1956)

ENATHUR, KANCHIPURAM - 631561

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

OBJECT ORIENTED PROGRAMMING USING C++

RECORD

NAME : _____

Reg.No: _____

Class : II BE (ECE)

Subject Code : BECF183P70

BONAFIDE CERTIFICATE

Certified to be the Bonafide Record of the work done by ……(Name) ……..…….(RollNo.) ……(Semester) ……. (Branch) in the Object Oriented Programming Laboratory Using C++ during the year 2021-2022

Place:

Date:

…….	**…….**
Faculty – in - charge	Head of the department
Sri.A.Niranjan	Prof.V.Swaminathan
Assistant Professor, ECE	Head of the dept., ECE

Submitted for the Practical Examination held on ……...

Register No: …….

Internal Examiner **External examiner**

CONTENTS

1.ArithmeticOperationsusing C++

Aim: Write a C++ program to find

(a)Addition of two numbers

(b)Difference of two numbers

(c)Product of two numbers

(d)Division of two numbers to get quotient

(e)Division of two numbers to get remainder

Program(a): Addition of two numbers

```
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
cout<<"Enter the values of a and b  : "<<endl;
cin>>a>>b;
c = a + b;
cout<<"Answer : "<<c<<endl;
getch();
}
```

Output:

Enter the values of a and b :

.

3

Answer : 5

Program(b): Difference of two numbers

```
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
cout<<"Enter the values of a and b  : "<<endl;
cin>>a>>b;
c = a - b;
cout<<"Answer : "<<c<<endl;
getch();
}
```

Output:

Enter the values of a and b :

3

2

Answer : 1

Program(c) : Product of two numbers

```
 #include<iostream.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
cout<<"Enter the values of a and b  : "<<endl;
cin>>a>>b;
 c = a * b;
cout<<"Answer : "<<c<<endl;
getch();
}
```

Output :

Enter the values of a and b :

3

4

Answer : 12

Program(d) : Division of two numbers to get quotient

```
#include<iostream.h>
#include<conio.h>
void main()
{
 int a,b,c;
clrscr();
cout<<"Enter the values of a and b  : "<<endl;
cin>>a>>b;
c = a/b;
cout<<"Answer : "<<c<<endl;
getch();
}
```

Output :

```
Enter the values of a and b :
12
3
Answer : 4
```

Program(e): Division of two numbers to get remainder

```
#include<iostream.h>
#include<conio.h>
void main()
{
 int a,b,c;
clrscr();
cout<<"Enter the values of a and b  : "<<endl;
cin>>a>>b;
 c = a % b;
cout<<"Answer : "<<c<<endl;
getch();
}
```

Output :

Enter the values of a and b :

14

3

Answer : 2

Result : Thus programs on arithmetic operations using C++ have been executed and verified successfully.

2.LoopcontrollingstatementsinC++

Aim: Write C++ programs using Loop control statements to find

(a)Sum of n numbers using while loop

(b)Factorial of given number using while loop

(c)Sum of all digits in a given number using while loop

(d)Sum of n numbers using for loop

(e)Factorial of a given number using for loop

(f)Prime or not prime using for loop

(g)Generate first five elements of Fibonacci series

(h)Sum of first five odd numbers using for loop

(i)Factorial of a given number using do while loop

(j)Break statement

(k)Continue statement

Program(a) : Sum of n numbers using while loop

```
#include<iostream.h>
#include<conio.h>
void main()
{
int i,n,sum;
i = 1;
sum = 0;
```

```
clrscr();
cout<<"Enter the value of n : "<<endl;
cin>>n;
 while(i<=n)
{
sum = sum + i;
i = i+1;
 }
cout<<"Answer : "<<sum<<endl;
getch();
}
```

Output :

```
Enter the value of n :
4
Answer : 10
```

Program(b) : Factorial of given number using while loop

```
#include<iostream.h>
#include<conio.h>
void main()
{
int i,n,fact;
```

```
i = 1;
fact = 1;
clrscr();
cout<<"Enter the value of n : "<<endl;
cin>>n;
 while(i<=n)
{
fact = fact * i;
i = i+1;
}
cout<<"Factorial : "<<fact<<endl;
getch();
}
```

Output :

```
Enter the value of n :
5
Factorial : 120
```

Program(c) : Sum of all digits in given number using while loop

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

```
{
 int n,sum,remainder;
sum = 0;
clrscr();
cout<<"Enter any number with at least 2 digits : "<<endl;
cin>>n;
 while(n!=0)
{
remainder = n%10;
sum = sum + remainder;
n = n/10;
}
cout<<"Answer : "<<sum<<endl;
getch();
}
```

Output :

```
Enter any number with at least 2 digits :
123
Answer : 6
```

Program(d) : Sum of n numbers using for loop

```
#include<iostream.h>
#include<conio.h>
void main()
{
int i,n,sum;
sum = 0;
clrscr();
cout<<"Enter the value of n : "<<endl;
cin>>n;
for(i=1;i<=n;i++)
{
sum = sum + i;
}
cout<<"Answer : "<<sum<<endl;
getch();
}
```

Output :

```
Enter the value of n:
5
Answer : 15
```

Program(e) : Factorial of given number using for loop

```
#include<iostream.h>
#include<conio.h>
void main()
{
int i,n,fact;
fact = 1;
clrscr();
cout<<"Enter the value of n : "<<endl;
cin>>n;
for(i=1;i<=n;i++)
{
fact =  fact * i;
}
cout<<"Factorial : "<<fact<<endl;
getch();
}
```

Output :

```
Enter the value of n :
4
Answer : 24
```

Program(f) : Prime or not prime using for loop

```
#include<iostream.h>
#include<conio.h>
void main()
{
int i,n,count;
 count = 0;
clrscr();
cout<<"Enter the value of n : "<<endl;
 for(i=1;i<=n;i++)
 {
 if(n%i==0)
{
count = count +1;
}
}
if(count==2)
{
cout<<"The number is Prime"<<endl;
}
else
```

```
{
cout<<"The number is Not Prime"<<endl;
}
getch();
}
```

Output :

```
Enter the value of n :
11
The number is Prime
```

Program(g) : Generate first five elements of Fibonacci series

```
#include<iostream.h>
#include<conio.h>
void main( )
{
 int i,n,a,b,c;
a = 0;
b = 1;
clrscr( );
cout<<"Enter the value of n : "<<endl;
cin>>n;
for(i=1;i<=n;i++)
```

```
{
cout<<" "<<a<<endl;
c = a + b;
a = b;
b = c;
}
getch( );
}
```

Output :

```
Enter the value of n :
5
 0
1
1
2
3
```

Program(h) : Sum of first 5 odd numbers using for loop

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

```
  int i,k,sum;
  k = 1;
  sum = 0;
clrscr( );
  for(i=1;i<=5;i++)
  {
   sum = sum + k;
   k = k + 2;
  }
cout<<"Answer : "<<sum<<endl;
getch( );
}
```

Output :

Answer : 25

Program(i) : Factorial of given number using do while loop

```
#include<iostream.h>
#include<conio.h>
void main()
{
int i,n,fact;
i = 1;
```

```
fact = 1;
clrscr();
cout<<"Enter the value of n : "<<endl;
cin>>n;
do
{
fact = fact*i;
i = i+1;
}
 while(i<=n);
cout<<"Answer : "<<fact<<endl;
getch( );
}
```

Output :

```
Enter the value of n :
7
Answer : 5040
```

Program(j) : Using Break statement

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

```
{
  int i;
clrscr( );
  for(i=1;i<=10;i++)
  {
   if(i>5)
   {
     break;
   }
cout<<" "<<i<<endl;
  }
getch( );
}
```

Output :

1

2

3

4

5

Program(k) : Using Continue statement

```
#include<iostream.h>
#include<conio.h>
void main( )
{
  int i;
clrscr( );
  for(i=1;i<=10;i++)
  {
    if(i==5)
    {
    continue;
    }
cout<<" "<<i<<endl;
  }
getch( );
}
```

Output :

1

2

3

4

6

7

8

9

10

Result : Thus the programs using loop controlling statements have been executedand verified successfully.

3.C++programsusingSwitchcase

Aim: Write a C++ program using Switch case

(a)To perform all arithmetic operations

(b)To print grade according to the marks

Program(a): Arithmeticoperations using switch case

```
#include<iostream.h>
#include<conio.h>
void main()
{
float a,b,c;
int opt;
clrscr();
cout<<"Enter the value of a and b : "<<endl;
cin>>a>>b;
cout<<"1.Addition"<<endl;
cout<<"2.Subtraction"<<endl;
cout<<"3.Multiplication"<<endl;
cout<<"4.Division"<<endl;
cout<<"Enter the option : "<<endl;
cin>>opt;
switch(opt)
{
```

```
case 1 :
c = a + b;
cout<<"Answer : "<<c<<endl;
break;
case 2 :
c = a - b;
cout<<"Answer : "<<c<<endl;
break;
case 3 :
c = a * b;
cout<<"Answer : "<<c<<endl;
break;
case 4 :
c = a / b;
cout<<"Answer : "<<c<<endl;
break;
default :
cout<<"Invalid option"<<endl;
break;
}
getch();
}
```

Output :

Enter the value of a and b :

2

3

1.Addition

2.Subtraction

3.Multiplication

4.Division

Enter the option :

1

Answer : 5

Program(b) : Print grades according to the marks using switch case

```
#include<iostream.h>
#include<conio.h>
void main( )
{
  int score;
clrscr();
cout<<"Enter the score for 100 marks : "<<endl;
cin>>score;
  switch(score/10)
  {
```

```
   case 10 :
cout<<"S Grade"<<endl;
   break;
   case 9 :
cout<<"A Grade"<<endl;
   break;
   case 8 :
cout<<"B Grade"<<endl;
   break;
   case 7 :
cout<<"C Grade"<<endl;
   break;
   case 6 :
cout<<"D Grade"<<endl;
   break;
   case 5 :
cout<<"E Grade"<<endl;
   break;
   default :cout<<"Fail"<<endl;
   break;
  }
getch();
```

}

Output :

Enter the score for 100 marks :

91

A Grade

Result : Thus the programs using switch case have been executed and verified successfully.

4.DecisionControllingstatementsinC++

Aim: Write a C++ program using decision control statements to find

(a)Given number is even or odd using if else

(b)Greatest of two numbers using if else

(c)Greatest of three numbers using if else

(d)Smallest of three numbers using if else

Program(a) : Given number is even or odd using if-else

```
#include<iostream.h>
#include<conio.h>
void main( )
{
  int n;
clrscr();
cout<<"Enter the value of n : "<<endl;
cin>>n;
  if(n%2==0)
  {
cout<<"Given number is Even"<<endl;
  }
  else
  {
cout<<"Given number is Odd"<<endl;
```

```
  }
getch();
}
```

Output :

```
Enter the value of n :
7
Given number is Odd
```

Program(b) : Greatest of two numbers using if-else

```
#include<iostream.h>
#include<conio.h>
void main( )
{
  int a,b;
clrscr();
cout<<"Enter the values of a and b : "<<endl;
cin>>a>>b;
  if(a>b)
  {
cout<<"a is greater than b"<<endl;
  }
  else
  {
```

```
cout<<"b is greater than a"<<endl;
  }
getch();
 }
```

Output :

```
Enter the values of a and b :
5
6
b is greater than a
```

Program(c) : Greatest of three numbers using if else

```
#include<iostream.h>
#include<conio.h>
void main( )
{
  int a,b,c;
clrscr( );
cout<<"Enter the values of a ,b and c  : "<<endl;
cin>>a>>b>>c;
  if(a>b)
  {
   if(a>c)
   {
```

```
cout<<"a is greatest number"<<endl;
   }
  else
  {
cout<<"c is greatest number"<<endl;
  }
  }
  else
  {
   if(b>c)
   {
cout<<"b is greatest number"<<endl;
   }
  else
   {
cout<<"c is greatest number"<<endl;
   }
   }
getch( );
}
```

Output :

Enter the values of a,b and c :

4

8

6

b is greatest number

Program(d) : Smallest of three numbers using if else

```
#include<iostream.h>
#include<conio.h>
void main( )
{
  int a,b,c;
clrscr();
cout<<"Enter the values of a,b and c  : "<<endl;
cin>>a>>b>>c;
  if(a<b)
  {
   if(a<c)
   {
cout<<"a is smallest number"<<endl;
   }
  else
  {
cout<<"c is smallest number"<<endl;
```

```
  }
  }
  else
  {
    if(b<c)
   {
cout<<"b is smallest number"<<endl;
   }
   else
  {
cout<<"c is smallest number"<<endl;
  }
  }
getch();
}
```

Output :

Enter the values of a,b and c :

1

9

5

a is smallest number

Result : Thus the programs using decision control statements have been executed and verified successfully.

5.Class&ObjectsinOOPS(Using Inline & Outline Functions)

Aim : Write a C++ program with classes and objects to find

(a)Area and Perimeter of rectangle using inline function

(b)Factorial of given number using inline function

(c)Simple Interest using inline function

(d)Area and Perimeter of rectangle using outline function

(e)Factorial of given number using outline function

(f)Simple Interest using outline function

Program(a) : Area and perimeter of rectangle using inline function

```
#include<iostream.h>
#include<conio.h>
class rect
{
  private :
floatl,b,a,p;
  public :
  void input( )
  {
cout<<"Enter the values of l and b : "<<endl;
cin>>l>>b;
  }
  void process()
```

```
  {
   a = l * b;
   p = 2 * (l + b);
  }
  void output()
  {
cout<<"Area : "<<a<<endl;
cout<<"Perimeter : "<<p<<endl;
  }
};
void main()
{
rect r;
clrscr();
r.input();
r.process();
r.output();
getch();
}
```

Output :

```
Enter the values of l and b :
10
```

20

Area : 200

Perimeter : 60

Program(b) : Factorial of given number using inline function

```
#include<iostream.h>
#include<conio.h>
class fact
{
 private :
 int i,n,fact;
 public :
 void input( )
 {
cout<<"Enter the value of n : "<<endl;
cin>>n;
 }
 void process( )
 {
 fact = 1;
 for(i=1;i<=n;i++)
 {
 fact = fact*i;
```

```
 }
 }
 void output( )
 {
cout<<"Answer : "<<fact<<endl;
 }
};
void main( )
{
fact f;
clrscr( );
f.input( );
f.process( );
f.output( );
getch( );
}
```

Output :

```
Enter the value of n :
4
Answer : 24
```

Program(c) : Simple Interest using inline function

```
#include<iostream.h>
#include<conio.h>
class simple
{
 private :
 float p,n,r,s;
 public :
 void input( )
 {
cout<<"Enter the values of p,n,r : "<<endl;
cin>>p>>n>>r;
 }
 void process( )
 {
 s = (p * n * r)/100;
 }
 void output( )
 {
cout<<"Simple Intrest : "<<s<<endl;
 }
};
void main( )
```

```
{
simple s;
clrscr( );
s.input( );
s.process( );
s.output( );
getch( );
}
```

Output :

```
Enter the value of p,n,r :
2000
3
2.5
Simple Interest : 150
```

Program(d) : Area and Perimeter of rectangle using outline function

```
#include<iostream.h>
#include<conio.h>
class rect
{
private :
floatl,b,a,p;
public :
```

```
void input( );
void process( );
void output( );
};
void rect::input( )
{
cout<<"Enter the values of l and b :"<<endl;
cin>>l>>b;
}
void rect::process( )
{
a = l * b;
p = 2 * (l + b);
}
void rect::output( )
{
cout<<"Area : "<<a<<endl;
cout<<"Perimeter : "<<p<<endl;
}
void main( )
{
rect r;
```

```
clrscr( );
r.input( );
r.process( );
r.output( );
getch( );
}
```

Output :

```
Enter the value of l and b :
Area : 600
Perimeter : 100
```

Program(e) : Factorial of given number using outline function

```
#include<iostream.h>
#include<conio.h>
class fact
{
private :
int n,i,fact;
public :
void input( );
void process( );
void output( );
};
```

```
void fact::input( )
{
cout<<"Enter the value of n :"<<endl;
cin>>n;
}
void fact::process()
{
fact = 1;
for(i=1;i<=n;i++)
{
fact = fact * i;
}
}
void fact::output()
{
cout<<"Factorial : "<<fact<<endl;
}
void main()
{
fact f;
clrscr();
f.input();
```

```
f.process();
f.output();
getch();
}
```

Output :

```
Enter the value of n :
5
Factorial : 120
```

Program(f) : Simple interest using outline function

```
#include<iostream.h>
#include<conio.h>
class simple
{
private :
floatp,n,r,s;
public :
void input();
void process();
void output();
};
void simple::input()
{
```

```
cout<<"Enter the values of p,n and r :"<<endl;
cin>>p>>n>>r;
}
void simple::process()
{
s = (p * n * r)/100;
}
void simple::output()
{
cout<<"Simple Interest : "<<s<<endl;
}
void main()
{
simple s;
clrscr();
s.input();
s.process();
s.output();
getch();
}
```

Output :

Enter the values of p,n and r :

4000

3

4.5

Simple Interest : 540

Result : Thus the C++ programs with classes and objects using inline function and outline function have been executed and verified successfully.

6.ProgramsusingMemberfunctionswithargumentsandretu rnvalue

Aim : Write C++ program to find

(a)Area and Perimeter of rectangle using member functions with arguments

(b)Factorial of given number using member functions with arguments

(c)Simple Interest using member functions with arguments

(d)Largest among two numbers using member functions with arguments and return value

Program(a): Area and Perimeter of rectangle using member function with arguments

```
#include<iostream.h>

#include<conio.h>

class rect

{

private :

floatl,b,a,p;

public :

void input(int x,int y)

{

l = x;

b = y;

}
```

```
void show( )
{
cout<<"Length : "<<l<<endl;
cout<<"Breadth : "<<b<<endl;
}
void process( )
{
a = l * b;
p = 2 * (l + b);
}
void output( )
{
cout<<"Area : "<<a<<endl;
cout<<"Perimeter : "<<p<<endl;
}
};
void main( )
{
rect r;
clrscr( );
r.input(40,20);
r.show( );
```

```
r.process( );
r.output( );
getch( );
}
```

Output:

Length : 40

Breadth : 20

Area : 800

Perimeter : 120

Program(b): Factorial of given number using member function with arguments

```
#include<iostream.h>
#include<conio.h>
class fact
{
private :
int i,n,fact;
public :
void input(int x)
{
n = x ;
}
```

```
void show( )
{
cout<<"n : "<<n<<endl;
}
void process( )
{
fact = 1;
for(i=1;i<=n;i++)
{
fact = fact * i;
}
}
void output( )
{
cout<<"Factorial : "<<fact<<endl;
}
};
void main( )
{
fact f;
clrscr( );
f.input(4);
```

```
f.show( );
f.process( );
f.output( );
getch( );
}
```

Output:

```
n : 4
Factorial : 24
```

Program(c): Simple Interest using member functions and arguments

```
#include<iostream.h>
#include<conio.h>
class simple
{
private :
int p,n,r,s;
public :
void input(float x,floaty,float z)
{
p = x ;
n = y ;
r = z ;
}
```

```
void show( )
{
cout<<"p : "<<p<<endl;
cout<<"n : "<<n<<endl;
cout<<"r : "<<r<<endl;
}
void process( )
{
s = (p * n * r)/100;
}
void output( )
{
cout<<"Simple Interest : "<<s<<endl;
}
};
void main( )
{
simple s;
clrscr( );
s.input(1000,3,5);
s.show( );
s.process( );
```

```
s.output( );
getch( );
}
```

Output:

```
p : 1000
n : 3
r : 5
Simple Interest : 150
```

Program(d): Largest among two numbers using member functions with arguments and return value

```
#include<iostream.h>
#include<conio.h>
class large
{
private :
int a,b;
public :
void input(int x,int y)
{
a = x;
b = y;
}
void show( )
```

```
{
cout<<"a : "<<a<<endl;
cout<<"b : "<<b<<endl;
}
int process( )
{
if(a>b)
{
return a;
}
else
{
return b;
}
}
};
void main( )
{
large l;
clrscr( );
l.input(2,3);
l.show( );
```

```
int big = l.process( );
cout<<"The Largest : "<<big<<endl;
getch( );
}
```

Output:

```
a : 2
b : 3
The Largest : 3
```

Result : Thus C++ programs using member functions with arguments and return value have been executed and verified successfully.

7.OOPSprogramsusingConstructors

Aim: Write C++ program to find

(a) Simple Interest using default constructor
(b) Factorial of given number using default constructor
(c) Area and Perimeter of rectangle using parameterized constructor
(d) Program using copy constructor
(e) Simple Interest using overloaded constructor

Program(a): Simple Interest using default constructor

```
#include<iostream.h>
#include<conio.h>
class simple
{
private :
float p,n,r,s;
public :
simple()
{
p = 2000;
n = 3;
r = 2.5;
}
void show()
{
```

```
cout<<"p : "<<p<<endl;
cout<<"n : "<<n<<endl;
cout<<"r : "<<r<<endl;
}
void process()
{
s = (p * n * r)/100;
}
void output()
{
cout<<"Simple Interest : "<<s<<endl;
}
};
void main()
{
simple s;
clrscr();
s.show();
s.process();
s.output();
getch();
}
```

Output:

p : 2000

n : 3

r : 2.5

Simple Interest : 150

Program(b): Factorial of given number using default constructor

```
#include<iostream.h>
#include<conio.h>
class fact
{
private :
int n,i,fact;
public :
fact()
{
n = 4;
}
void show()
{
cout<<"n : "<<n<<endl;
}
void process()
```

```
{
fact = 1;
for(i=1;i<=n;i++)
{
fact = fact * i;
}
}
void output()
{
cout<<"Factorial : "<<fact<<endl;
}
};
void main()
{
fact f;
clrscr();
f.show();
f.process();
f.output();
getch();
}
```

Output:

n : 4

Factorial : 24

Program(c): Area and Perimeter of rectangle using parameterized construtor

```
#include<iostream.h>
#include<conio.h>
class rect
{
private :
float l,b,a,p;
public :
rect(int x,int y)
{
l = x;
b = y;
}
void show( )
{
cout<<"Length : "<<l<<endl;
cout<<"Breadth : "<<b<<endl;
}
void process( )
```

```
{
a = l * b;
p = 2 * (l + b);
}
void output( )
{
cout<<"Area : "<<a<<endl;
cout<<"Perimeter : "<<p<<endl;
}
};
void main( )
{
rect r(20,30);
clrscr( );
r.show( );
r.process( );
r.output( );
getch( );
}
```

Output:

```
Length : 20
Breadth : 30
```

Area : 600

Perimeter : 100

Program(d): Program using copy constructor

```
#include<iostream.h>
#include<conio.h>
class copy
{
private :
int a;
public :
copy(int x)
{
a = x;
}
void show( )
{
cout<<"The value of a : "<<a<<endl;
}
};
void main( )
{
copy c(10);
```

```
copy y(c);
clrscr( );
c.show( );
y.show( );
getch( );
}
```

Output:

```
The value of a : 10
The value of a : 10
```

Program(e): Simple Interest using overloaded constructor

```
#include<iostream.h>
#include<conio.h>
class simple
{
private :
float p,n,r,s;
public :
simple( )
{
p = 2000;
n = 2.5;
r = 3.2;
```

```
}
simple(float x,floaty,float z)
{
p = x;
n = y;
r = z;
}
void show( )
{
cout<<"p : "<<p<<endl;
cout<<"n : "<<n<<endl;
cout<<"r : "<<r<<endl;
}
void process( )
{
s = (p * n * r)/100;
}
void output( )
{
cout<<"Simple Interest : "<<s<<endl;
}
};
```

```
void main( )
{
simple s,x(3000,2,5);
clrscr( );
s.show( );
s.process( );
s.output( );
x.show( );
x.process( );
x.output( );
getch( );
}
```

Output:

```
p : 2000
n : 2.5
r : 3.2
Simple Interest : 160
p : 3000
n : 2
r : 5
Simple Interest : 300
```

Result: Thus the C++ programs using constructorshave been executed and verified successfully.

8.C++ProgramsusingDestructors

Aim: Write C++ program using destructors

Program: Destructors

```
#include<iostream.h>
#include<conio.h>
class test
{
 private :
int a,b;
public :
test ( )
{
a = 10;
b = 20;
}
~test( )
{
cout<<"a : "<<a<<endl;
cout<<"b : "<<b<<endl;
getch( );
}
};
```

```
void main( )
{
clrscr( );
test t;
}
```

Output:

a : 10

b : 20

Result: Thus the C++ program using destructor has been executed and verified successfully.

9.ProgramsusingInheritanceinOOPS

Aim : Write a C++ program using

(a)Single level Inheritance(Visibility mode : private)

(b)Single level Inheritance(Visibility mode : public)

(b)Multiple Inheritance

(c)Multilevel Inheritance

(d)Hierarchical Inheritance

(f)Hybrid Inheritance

(g)Constructor in Inheritance

(h)Destructor in Inheritance

Program(a): Single level Inheritance(Visibility mode : public)

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
private :
char name[20];
int rollno;
public :
void input1( )
{
```

```
cout<<"Enter the name and rollno : "<<endl;
cin>>name>>rollno;
}
void output1( )
{
cout<<"Name : "<<name<<endl;
cout<<"Rollno : "<<rollno<<endl;
}
};
class physical:public student
{
private :
float h,w;
public :
void input2( )
{
cout<<"Enter the height and weight : "<<endl;
cin>>h>>w;
}
void output2( )
{
cout<<"Height : "<<h<<endl;
```

```
cout<<"Weight : "<<w<<endl;
}
};
void main( )
{
physical s;
clrscr( );
s.input1( );
s.input2( );
s.output1( );
s.output2( );
getch( );
}
```

Output :

```
Enter the name and rollno :
John
223
Enter the height and weight :
186
75
Name : John
Rollno : 223
```

Height : 186

Weight : 75

Program(b) : Single Level Inheritance (Visibility mode : private)

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
private :
char name[20];
int rollno;
public :
void input1( )
{
cout<<"Enter the name and rollno : "<<endl;
cin>>name>>rollno;
}
void output1( )
{
cout<<"Name : "<<name<<endl;
cout<<"Rollno : "<<rollno<<endl;
}
```

```
};
class physical:private student
{
private :
float h,w;
public :
void input2( )
{
input1( );
cout<<"Enter the height and weight : "<<endl;
cin>>h>>w;
}
void output2( )
{
output1( );
cout<<"Height : "<<h<<endl;
cout<<"Weight : "<<w<<endl;
}
};
void main( )
{
physical s;
```

```
clrscr( );
s.input2( );
s.output2( );
getch( );
}
```

Output :

```
Enter the name and rollno :
John
223
Enter the height and weight :
186
75
Name : John
Rollno : 223
Height : 186
Weight : 75
```

Program(c) : Multiple Inheritance

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
```

```
private :
char name[20];
int rollno;
public :
void input1( )
{
cout<<"Enter the name and rollno : "<<endl;
cin>>name>>rollno;
}
void output1( )
{
cout<<"Name : "<<name<<endl;
cout<<"Rollno : "<<rollno<<endl;
}
};
class marks
{
protected :
int m1,m2,m3,m4,m5,m6;
public :
void input2( )
{
```

```
cout<<"Enter the marks : "<<endl;
cin>>m1>>m2>>m3>>m4>>m5>>m6;
}
void output2( )
{
cout<<"Maths : "<<m1<<endl;
cout<<"EDC   : "<<m2<<endl;
cout<<"DSD   : "<<m3<<endl;
cout<<"SS  : "<<m4<<endl;
cout<<"NT    : "<<m5<<endl;
cout<<"OOPS  : "<<m6<<endl;
}
};
class result:publicmarks,public student
{
private :
int total;
float average;
public :
void output3( )
{
total = m1+m2+m3+m4+m5+m6;
```

```
average = total/6;
cout<<"Total = "<<total<<endl;
cout<<"Average = "<<average<<endl;
}
};
void main( )
{
result r;
clrscr( );
r.input1( );
r.input2( );
r.output1( );
r.output2( );
r.output3( );
getch( );
}
```

Output :

```
Enter the name and rollno :
John
223
Enter the marks :
95
```

92

94

96

97

99

Name : John

Rollno : 223

Maths : 95

EDC : 92

DSD : 94

SS : 96

NT : 97

OOPS : 99

Total = 573

Average = 95

Program(d): Multilevel Inheritance

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
private :
```

```
char name[20];
int rollno;
public :
void input1( )
{
cout<<"Enter the name and rollno : "<<endl;
cin>>name>>rollno;
}
void output1( )
{
cout<<"Name : "<<name<<endl;
cout<<"Rollno : "<<rollno<<endl;
}
};
class marks:public student
{
protected :
int m1,m2,m3,m4,m5,m6;
public :
void input2( )
{
cout<<"Enter the marks : "<<endl;
```

```
cin>>m1>>m2>>m3>>m4>>m5>>m6;
}
void output2( )
{
cout<<"MATHS : "<<m1<<endl;
cout<<"EDC   : "<<m2<<endl;
cout<<"DSD   : "<<m3<<endl;
cout<<"SS    : "<<m4<<endl;
cout<<"NT    : "<<m5<<endl;
cout<<"OOPS  : "<<m6<<endl;
}
};
class result:public marks
{
private :
int total;
float average;
public :
void output3( )
{
total = m1+m2+m3+m4+m5+m6;
average = total/6;
```

```
cout<<"Total : "<<total<<endl;
cout<<"Average : "<<average<<endl;
}
};
void main( )
{
result r;
clrscr( );
r.input1( );
r.input2( );
r.output1( );
r.output2( );
r.output3( );
getch( );
}
```

Output :

```
Enter the name and rollno :
John
223
Enter the marks :
95
92
```

94

96

97

99

Name : John

Rollno : 223

Maths : 95

EDC   : 92

DSD   : 94

SS   : 96

NT   : 97

OOPS : 99

Total = 573

Average = 95

Program(e) : Hierarchical Inheritance

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class account
{
private :
char name[20];
```

```
long int accountno;
public :
void input1( )
{
cout<<"Enter the name and account no : "<<endl;
cin>>name>>accountno;
}
void output1( )
{
cout<<"Account holder Name : "<<name<<endl;
cout<<"Account Number : "<<accountno<<endl;
}
};
class savings:public account
{
private :
int bal;
public :
void input2( )
{
cout<<"Enter the balance : "<<endl;
cin>>bal;
```

```
}
void output2( )
{
if(bal<500)
{
cout<<"Minimum Balance should be 500"<<endl;
}
else
{
cout<<"Your account is a Savings account"<<endl;
}
}
};
class current:public account
{
private :
int bal;
public :
void input2( )
{
cout<<"Enter the balance : "<<endl;
cin>>bal;
```

```
}
void output2( )
{
if(bal<1000)
{
cout<<"Minimum balnace should be 1000"<<endl;
}
else
{
cout<<"Your account is a Current account"<<endl;
}
}
};
void main( )
{
int op;
clrscr( );
cout<<"1.Savings Account"<<endl;
cout<<"2.Current Account"<<endl;
cout<<"Choose the option"<<endl;
cin>>op;
if(op==1)
```

```
{
savings s;
s.input1( );
s.input2( );
s.output1( );
s.output2( );
}
else if(op==2)
{
current c;
c.input1( );
c.input2( );
c.output1( );
c.output2( );
}
else
{
cout<<"Invalid Option"<<endl;
}
getch( );
}
```

Output :

1.Savings Account

2.Current Account

Choose the option

2

Enter the name and account no :

John

123456789

Enter the balance :

6000

Account holder Name : John

Account Number : 123456789

Your account is Current account

Program(f) : Hybrid Inheritance

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
 private :
 char name[20];
 int rollno;
```

```
 public :
 void input1( )
 {
cout<<"Enter the name and rollno : "<<endl;
cin>>name>>rollno;
 }
 void output1( )
 {
cout<<"Name : "<<name<<endl;
cout<<"Rollno : "<<rollno<<endl;
 }
};
class marks:public student
{
 protected :
 int m1,m2,m3,m4,m5,m6;
 public :
 void input2( )
 {
cout<<"Enter six subjects marks : "<<endl;
cin>>m1>>m2>>m3>>m4>>m5>>m6;
 }
```

```
 void output2( )
 {
cout<<"Maths : "<<m1<<endl;
cout<<"EDC   : "<<m2<<endl;
cout<<"DSD   : "<<m3<<endl;
cout<<"SS    : "<<m4<<endl;
cout<<"NT  : "<<m5<<endl;
cout<<"OOPS  : "<<m6<<endl;
 }
};
class sports
 {
 protected :
 int sportsmarks;
 public :
 void input3( )
 {
cout<<"Enter the sports marks for 5 marks : "<<endl;
cin>>sportsmarks;
 }
 void output3( )
 {
```

```
cout<<"Sports Marks : "<<sportsmarks<<endl;
 }
};
class result:publicmarks,public sports
{
private :
 int total;
 float avg,asp;
 public :
 void output4( )
 {
 total = m1 + m2 + m3 +m4 + m5 + m6;
avg = total/6;
 asp =  avg + sportsmarks;
cout<<"Total : "<<total<<endl;
cout<<"Average : "<<avg<<endl;
 if(asp>100)
 {
cout<<"ASP = 100"<<endl;
 }
 else
 {
```

```
cout<<"ASP =  "<<asp<<endl;
 }
 }
};
void main( )
{
result r;
clrscr( );
r.input1( );
r.input2( );
r.input3( );
r.output1( );
r.output2( );
r.output3( );
r.output4( );
getch( );
}
```

Output :

```
Enter the name and rollno :
John
223
Enter the six subjects marks :
```

95

92

94

96

97

99

Enter the sports marks for 5 marks :

5

Name : John

Rollno : 223

Maths : 95

EDC : 92

DSD : 94

SS : 96

NT : 97

OOPS : 99

Sports Marks : 5

Total : 573

Average : 95

ASP : 100

Program(g): Constructor in Inheritance

```
#include<iostream.h>
#include<conio.h>
class base
{
 public :
 base( )
 {
cout<<"Base class constructor"<<endl;
 }
};
class derived:public base
{
 public :
 derived( )
 {
cout<<"Derived class cnstructor"<<endl;
 }
};
void main( )
{
clrscr( );
```

```
derived d;
getch( );
}
```

Output :

```
Base class constructor
Derived class constructor
```

Program(h): Destructor in Heritance

```
#include<iostream.h>
#include<conio.h>
class base
{
 public :
 ~base( )
 {
cout<<"Base class constructor"<<endl;
getch( );
 }
};
class derived:public base
{
 public :
 ~derived( )
```

```
 {
cout<<"Derived class constructor"<<endl;
  }
};
void main( )
{
clrscr( );
derived d;
}
```

Output :

Derived class constructor

Base class constructor

Result: Thus the C++ programs using different types of inheritancehave been executed and verified successfully.

10.OOPSProgramsusingFriendfunctions

Aim : Write C++ Program using Friend function

(a)To Print a = 10 & b = 20

(b)To find largest of two numbers

Program(a) : Print a = 10 & b = 20 using Friend Function

```
#include<iostream.h>
#include<conio.h>
class test
{
private:
int a,b;
public:
friend void print(test);
};
void print(test t)
{
t.a=10;
t.b=20;
cout<<"a = "<<t.a<<endl;
cout<<"b = "<<t.b<<endl;
}
void main( )
```

```
{
test t;
clrscr( );
print(t) ;
getch( );
}
```

Output :

```
a : 10
b : 20
```

Program(b) : To find largest of two numbers

```
#include<iostream.h>
#include<conio.h>
class test2;
class test1
{
private:
int a;
public:
void input1( )
{
cout<<"Enter the value of a : "<<endl;
cin>>a;
```

```
}
friend void big(test1,test2);
};
class test2
{
private:
int b;
public:
void input2( )
{
cout<<"Enter the value of b : "<<endl;
cin>>b;
}
friend void big(test1,test2);
};
void big(test1 t1,test2 t2)
{
if(t1.a>t2.b)
{
cout<<"Biggest = "<<t1.a<<endl;
}
else if(t2.b>t1.a)
```

```
{
cout<<"Biggest = "<<t2.b<<endl;
}
else
{
cout<<"Both are equal"<<endl;
}
}
void main( )
{
test1 t1;
test2 t2;
clrscr( );
t1.input1( );
t2.input2( );
big(t1,t2);
getch( );
}
```

Output :

```
Enter the value of a :
2
Enter the value of b :
```

3

Biggest = 3

Result: Thus the C++ programs using friend functionhave been executed and verified successfully.

11.ProgramsusingThisPointerinOOPS

Aim : Write C++ program to

(a)Illustrate concept of this pointer

(b)Differentiate actual variable from local variable

Program(a) : Illustrate concept of this pointer

```
#include<iostream.h>
#include<conio.h>
class test
{
private:
int a,b;
public:
void input( )
{
cout<<"Enter the value of a and b : "<<endl;
cin>>a>>b;
}
void output( )
{
cout<<"a  =  "<<(*this).a<<endl;
cout<<"b  =  "<<(*this).b<<endl;
cout<<"The address of the object  =  "<<this<<endl;
```

```
}
};
void main( )
{
test t;
clrscr( );
t.input( );
t.output( );
getch( );
}
```

Output :

```
Enter the value of a and b :
6
8
a = 6
b = 7
The address of the object = 0x8f82fff2
```

Program(b) : Differentiate actual variable from local variable

```
#include<iostream.h>
#include<conio.h>
class test
{
```

```
private:
int a,b;
public:
void input(int a,int b)
{
(*this).a=a;
(*this).b=b;
}
void output( )
{
cout<<"a = "<<a<<endl;
cout<<"b = "<<b<<endl;
cout<<"Address of the object = "<<this<<endl;
}
};
void main( )
{
test t;
clrscr( );
t.input(10,20);
t.output( );
getch( );
```

}

Output :

a = 10

b = 20

The address of the object = 0x8f82fff2

Result: Thus the C++ programs using this pointer have been executed and verified successfully.

12.ProgramsusingNew & DeleteOperatorinOOPS

Aim : Write C++ program to

(a)Illustrate concept of New operator(Type 1)

(b) Illustrate concept of New operator(Type 2)

(c) Illustrate concept of New operator using for loop

(d)Illustrate the concept of Delete operator

Program(a) : Illustrate concept of New operator(Type 1)

```
#include<iostream.h>
#include<conio.h>
void main( )
{
int *p;
p=new int;
*p=10;
clrscr( );
cout<<"Answer = "<<*p<<endl;
getch( );
}
```

Output :

Answer = 10

Program(b) : Illustrate concept of New operator(Type 2)

```
#include<iostream.h>
#include<conio.h>
void main( )
{
int *p;
p=new int[5];
p[0]=10;
p[1]=20;
p[2]=30;
p[3]=40;
p[4]=50;
clrscr();
cout<<"   "<<p[0]<<endl;
cout<<"   "<<p[1]<<endl;
cout<<"   "<<p[2]<<endl;
cout<<"   "<<p[3]<<endl;
cout<<"   "<<p[4]<<endl;
getch( );
}
```

Output :

10

20

30

40

50

Program(c) : Illustrate concept of New operator using for loop

```
#include<iostream.h>
#include<conio.h>
void main( )
{
int *p;
int i;
p=new int[5];
p[0]=10;
p[1]=20;
p[2]=30;
p[3]=40;
p[4]=50;
clrscr( );
for(i=0;i<5;i++)
{
```

```
cout<<"   "<<p[i]<<endl;
}
getch( );
}
```

Output :

```
10
 20
 30
 40
 50
```

Program(d) : Illustrate the concept of Delete operator

```
#include<iostream.h>
#include<conio.h>
void main( )
{
int *p;
p=new int;
*p=10;
cout<<"Answer = "<<*p<<endl;
delete p;
cout<<"The answer after delete = "<<p<<endl;
getch( );
```

}

Output :

Answer = 10

The answer after delete = 0x8f830dd0

Result: Thus the C++ programs using new & delete operator have been executed and verified successfully.

13.ProgramsbasedonOperatoroverloadinginOOPS

Aim : Write C++ program

(a)Using Unary operator overloading

(b)Reversing values usingUnary operator overloading

(c)Using Binary operator overloading

(d)String Concatenation using operator overloading

Program(a) : Unary operator overloading

```
#include<iostream.h>
#include<conio.h>
class test
{
private :
int a;
public :
test( )
{
a = 1;
}
void operator++( )
{
a++;
}
```

```
void operator--( )
{
a--;
}
void output( )
{
cout<<"a : "<<a<<endl;
}
};
void main( )
{
test t;
clrscr( );
t++;
t.output( );
t--;
t.output( );
getch( );
}
```

Output :

```
a : 2
a : 1
```

Program(b) : Reversing values usingUnary operator overloading

```
#include<iostream.h>
#include<conio.h>
class test
{
private :
int a,b,c;
public :
test( )
{
a = 10;
b = 20;
c = 30;
}
void show( )
{
cout<<"a : "<<a<<endl;
cout<<"b : "<<b<<endl;
cout<<"c : "<<c<<endl;
}
void operator-( )
{
```

```
a = -a;
b = -b;
c = -c;
}
};
void main( )
{
test t;
clrscr( );
t.show( );
-t;
t.show( );
getch( );
}
```

Output :

```
a : 10
b : 20
c : 30
a : -10
b : -20
c : -30
```

Program(c) : Using Binary operator overloading

```
#include<iostream.h>
#include<conio.h>
class test
{
private :
int a;
public :
void input( )
{
cin>>a;
}
void operator==(test t2)
{
if(a==t2.a)
{
cout<<"Objects are equal"<<endl;
}
else
{
cout<<"Objects are not equal"<<endl;
}
```

```
}
};
void main( )
{
test t1,t2;
clrscr( );
cout<<"Enter t1 object value : "<<endl;
t1.input( );
cout<<"Enter t2 object value : "<<endl;
t2.input( );
t1==t2;
getch( );
}
```

Output :

```
Enter the t1 object value :
2
Enter the t2 object value :
4
Objects are not equal
```

Program(d): String Concatenation using operator overloading

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

```
#include<string.h>
class test
{
private :
char st[50];
public :
void input( )
{
cout<<"Enter the string : "<<endl;
cin>>st;
}
void output( )
{
cout<<"String : "<<st<<endl;
}
test operator+(test t2)
{
test t3;
strcpy(t3.st,st);
strcat(t3.st," ");
strcat(t3.st,t2.st);
return t3;
```

```
}
};
void main( )
{
test t1,t2,t3;
clrscr( );
t1.input( );
t2.input( );
t3 = t1+t2;
t3.output( );
getch( );
}
```

Output :

```
Enter the string :
Ram
Enter the string :
kumar
String : Ram Kumar
```

Result: Thus the programs using operator overloading u have been executed and verified successfully.

14.ProgramsbasedOperatoroverloadingusingFriendFunctionsinOOPS

Aim : Write the C++ program to

(a)overload greater than (>) operator using friend function

(b)overload comparison(==) operator using friend function

(c)overload multiplication(*)operator using friend function

Program(a) : overload greater than (>) operator using friend function

```
#include<iostream.h>
#include<conio.h>
class test2;
class test1
{
private :
int a;
public :
void input1( )
{
cout<<"Enter the value of a : "<<endl;
cin>>a;
}
friend void operator>(test1,test2)
};
```

```
class test2
{
private :
int b;
public :
void input2( )
{
cout<<"Enter the value of b : "<<endl;
cin>>b;
}
friend void operator>(test1,test2);
};
void operator>(test1 t1,test2 t2)
{
if(t1.a>t2.b)
{
cout<<"a is big"<<endl;
}
else
{
cout<<"b is big"<<endl;
}
```

```
}
void main( )
{
test1 t1;
test2 t2;
clrscr( );
t1.input1( );
t2.input2( );
t1>t2;
getch( );
}
```

Output :

```
Enter the value of a :
4
Enter the value of b :
5
b is big
```

Program(b) : overload comparison(==) operator using friend function

```
#include<iostream.h>
#include<conio.h>
class test2;
class test1
```

```
{
private :
int a;
public :
void input1( )
{
cout<<"Enter the value of a : "<<endl;
cin>>a;
}
friend void operator==(test1,test2)
};
class test2
{
private :
int b;
public :
void input2( )
{
cout<<"Enter the value of b : "<<endl;
cin>>b;
}
friend void operator==(test1,test2);
```

```
};
void operator==(test1 t1,test2 t2)
{
if(t1.a==t2.b)
{
cout<<"Both are equal"<<endl;
}
else
{
cout<<"Both are not equal"<<endl;
}
}
void main( )
{
test1 t1;
test2 t2;
clrscr( );
t1.input1( );
t2.input2( );
t1==t2;
getch( );
}
```

Output :

Enter the value of a :

4

Enter the value of b :

4

Both are equal

Program(c) : overload multiplication(*)operator using friend function

```
#include<iostream.h>

#include<conio.h>

class test2;

class test1

{

private :

int a;

public :

void input1( )

{

cout<<"Enter the value of a : "<<endl;

cin>>a;

}

friend void operator*(test1,test2)

};
```

```
class test2
{
private :
int b;
public :
void input2( )
{
cout<<"Enter the value of b : "<<endl;
cin>>b;
}
friend void operator*(test1,test2);
};
void operator*(test1 t1,test2 t2)
{
int t;
t = t1.a * t2.b;
cout<<"Answer : "<<t<<endl;
}
void main( )
{
test1 t1;
test2 t2;
```

```
clrscr( );
t1.input1( );
t2.input2( );
t1*t2;
getch( );
}
```

Output :

```
Enter the value of a :
5
Enter the value of b :
6
Answer : 30
```

Result: Thus the C++ programs using operator overloading using friend function have been executed and verified successfully.

15.ArraysasanobjectsinOOPS

Aim : Write C++ program using arrays inside the class

Program :

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
private :
char name[20];
int rollno,marks[6],i;
public :
void input( );
void output( );
};
void student::input( )
{
cout<<"Enter the name and rollno : "<<endl;
cin>>name>>rollno;
cout<<"Enter six subjects marks : "<<endl;
for(i=0;i<6;i++)
{
cin>>marks[i];
```

```
}
}
void student::output( )
{
cout<<"Name : "<<name<<endl;
cout<<"Rollno : "<<rollno<<endl;
int total=0;
float avg;
for(i=0;i<6;i++)
{
total = total + marks[i];
}
avg = total/6;
cout<<"Total   : "<<total<<endl;
cout<<"Average : "<<avg<<endl;
cout<<"Result  : ";
for(i=0;i<6;i++)
{
if(marks[i]<50)
{
cout<<"Fail";
goto last;
```

```
}
}
cout<<"Pass";
last :
getch();
}
void main( )
{
student s;
clrscr( );
s.input( );
s.output( );
getch( );
}
```

Output :

```
Enter the name and rollno :
John
223
Enter six subjects marks :
98
97
96
```

90

98

97

Name : John

Rollno : 223

Total : 576

Average : 96

Result : Pass

Result: Thus the C++ programs using arrays inside the class have been executed and verified successfully.

16.ProgramsusingTemplates in OOPS

Aim : Write C++ program to

(a)Add 2 integers and 2 floats using function templates

(b)Overloading Function Template

(c)Class template (inline function)

(d)Class template (outline function)

Program(a) : Add 2 integers and 2 floats using function templates

```
#include<iostream.h>
#include<conio.h>
template <class t>
t add(t  a,t  b)
{
return a+b;
}
void main( )
{
clrscr( );
cout<<"Sum of two integers = "<<add(3,4)<<endl;
cout<<"Sum of two float = "<<add(4.5,7.3)<<endl;
getch( );
}
```

Output :

Sum of two integers = 7

Sum of two floats = 11.8

Program(b) : Overload function template

```
#include<iostream.h>
#include<conio.h>
template<class t1,class t2>
float sum(t1 a,t2 b)
{
return a+b;
}
void main( )
{
clrscr( );
cout<<"Sum of two integers : "<<sum(6,7)<<endl;
cout<<"Sum of two float :  "<<sum(8.7,7.4)<<endl;
cout<<"Sum of 1 float and 1 integer  : "<<sum(4.4,8)<<endl;
cout<<"Sum of 1 integer and 1 float :  "<<sum(5,5.5)<<endl;
getch( );
}
```

Output :

Sum of two integers : 13.0

Sum of two float : 16.1

Sum of 1 float and 1 integer : 12.4

Sum of 1 integer and 1 float : 10.5

Program(c) : Class template (inline function)

```
#include<iostream.h>
#include<conio.h>
template<class t>
class test
{
private:
t a,b;
public:
void input( )
{
cin>>a>>b;
}
t sum( )
{
return a+b;
}
```

```
};
void main( )
{
clrscr( );
test <int>t1;
test <float>t2;
cout<<"Enter the two integers : "<<endl;
t1.input( );
cout<<"Sum of two integers : "<<t1.sum( )<<endl;
cout<<"Enter the two floats :  "<<endl;
t2.input( );
cout<<"Sum of two floats :  "<<t2.sum( )<<endl;
getch( );
}
```

Output :

```
Enter the two integers :
3
4
Sum of two integers : 7
Enter the two floats :3.5
6.5
Sum of two floats : 10
```

Program(d) : Class template (outline function)

```
#include<iostream.h>
#include<conio.h>
template<class t>
class test
{
t a,b;
public:
test (t x,t y)
{a=x;b=y;}
void input( );
void sum( );
};
template <class t>
void test<t>::input( )
{
cin>>a>>b;
}
template<class t>
void test<t>::sum( )
{
cout<<a+b<<endl;
```

```
}
void main( )
{
clrscr( );
int x;
float y;
test<int> t1(x,y);
test <float>t2(x,y);
cout<<"Enter two integers :" <<endl;
t1.input( );
cout<<"The sum of two integers : "<<endl;
t1.sum( );
cout<<"Enter two floats :"<<endl;
t2.input( );
cout<<"The sum of two floats : "<<endl;
t2.sum( );
getch( );
}
```

Output :

```
Enter two integers :
6
7
```

The sum of two integers :

13

Enter two floats :

3.3

4.4

The sum of two floats :

7.7

Result: Thus the C++ programs using templates have been executed and verified successfully.

17.ProgramusingManipulatorsinOOPS

Aim : Write C++ program

(a)Invert digits using manipulators

(b)To set precise value for a floating value

(c)Using iosflags manipulators

(d)Conversion of number systems using manipulators

Program(a) : Invert digits using manipulators

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
void main( )
{
cout<<setfill('*');
cout<<setw(5)<<"1"<<endl;
cout<<setw(5)<<"10"<<endl;
cout<<setw(5)<<"101"<<endl;
getch( );
}
```

Output :

```
****1
***10
**101
```

Program(b) : To set precise value for floating value

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
void main( )
{
clrscr();
cout<<setprecision(2)<<22/7.0<<endl;
cout<<setprecision(3)<<22/7.0<<endl;
cout<<setprecision(4)<<22/7.0<<endl;
getch( );
}
```

Output :

```
3.14
3.143
3.1429
```

Program(c) : Using iosflags manipulators

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

```
clrscr( );
cout<<setfill('*');
cout<<setw(10)<<setiosflags(ios::left)<<"RAM"<<endl;
cout<<setw(10)<<setiosflags(ios::right)<<"VENKAT"<<endl;
cout<<setiosflags(ios::showpos)<<100<<endl;
getch( );
}
```

Output :

```
RAM*******
****VENKAT
+100
```

Program(d) : Conversion of number systems using manipulators

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
void main( )
{
clrscr( );
cout<<setbase(8)<<65<<endl;
cout<<setbase(16)<<65<<endl;
getch( );
}
```

Output :

101

41

Result: Thus the C++ programs using manipulators have been executed and verified successfully.

18.ProgramsusingVirtualfunctionsin OOPS

Aim : Write C++ program

(a)Using Pure Virtual function and abstract class

(b)Using Virtual function

Program(a) : Using Pure Virtual function and abstract class

```
#include<iostream.h>
#include<conio.h>
class shape
{
protected:
float d1,d2;
public:
void input( )
{
cin>>d1>>d2;
}
virtual float area( )=0;
};
class triangle:public shape
{
public:
float area( )
```

```
{
return 0.5*d1*d2;
}
};
class rectangle:public shape
{
public:
float area( )
{
return d1*d2;
}
};
void main( )
{
triangle t;
clrscr( );
cout<<"Enter base and height :  "<<endl;
t.input( );
cout<<"Area of the triangle = "<<t.area( )<<endl;
rectangle r;
cout<<"Enter length and breadth :  "<<endl;
r.input( );
```

```
cout<<"Area of rectangle = "<<r.area( )<<endl;
getch( );
}
```

Output :

```
Enter base and height :
4
5
Area of the triangle = 10
Enter length and breadth :
2.5
6.5
Area of rectangle = 16.25
```

Program(b) : Virtual function

```
#include<iostream.h>
#include<conio.h>
class c1
{
public:
void c1function( )
{
cout<<"C1 class function "<<endl;
}
```

```
};
class c2:public virtual c1
{
public:
void c2function( )
{
cout<<"C2 class function "<<endl;
}
};
class c3:public virtual c1
{
public:
void c3function( )
{
cout<<"C3 class function "<<endl;
}
};
class c4:public c2,public c3
{
public:
void c4function( )
{
```

```
cout<<"C4 class function "<<endl;
}
};
void main( )
{
c4 x;
clrscr( );
x.c1function( );
x.c2function( );
x.c3function( );
x.c4function( );
getch( );
}
```

Output :

```
C1 class function
C2 class function
C3 class function
C4 class function
```

Result: Thus the C++ programs using virtual functions have been executed and verified successfully.

*****