**/\* Program No. :**

**Aim : WAP to show the working of virtual functions and abstract classes, how the derived class override the virtual functions of a normal base class and abstract base class.**

**\*/**

#include<iostream.h>

#include<conio.h>

class base1

{

public:

void display()

{

cout<<"\n\nBase1 class called.";

}

};

class base2

{

public:

virtual void display()

{

cout<<"\n\nBase2 class called.";

}

};

class base3

{

public:

virtual void display()=0;

};

class derived1:public base1

{

public:

void display()

{

cout<<"\n\nDerived1 class of Base1 class called";

}

};

class derived2:public base2

{

public:

void display()

{

cout<<"\n\nDerived2 class of Base2 class called";

}

};

class derived3:public base3

{

public:

void display()

{

cout<<"\n\nDerived3 class of Base3 class called";

}

};

void main()

{

clrscr();

base1 ob1,\*ptr1;

base2 ob2,\*ptr2;

//base3 ob3; Error : Cannot create instance of abstract class 'base3'

base3 \*ptr3;

derived1 ob4;

derived2 ob5;

derived3 ob6;

ob1=ob4;

ob2=ob5;

ptr1=&ob4;

ptr2=&ob5;

ptr3=&ob6;

ob1.display(); //Only virtual functions can be overrided

ob2.display(); //Assingment operator doesn't override base class function

ptr1->display(); //Only virtual functions can be overrided

ptr2->display();

ptr3->display();

ob4.display();

ob5.display();

ob6.display();

getch();

}

**/\***

**Name : Rohit Aggarwal**

**Roll No. : 7CS-097**

**\*/**