**Practical No.6**

**Virtual functions and abstract classes**

**Program 6(a):** Implement the concept of method overriding.

**Example:** Write a program to demonstrate the function overriding.

**Coding:**

#include<iostream.h>

#include<conio.h>

class base

{

protected:

int a,b;

public:

void read()

{

cout<<"Enter two values:";

cin>>a>>b;

}

void display()

{

cout<<"The values are :"<<a<<"\n"<<b;

}

};

class sub:public base

{

protected:

int c,d;

public:

void read()

{

cout<<"Enter 4 values:";

cin>>a>>b>>c>>d;

}

void display()

{

cout<<"The values are :"<<a<<"\n"<<b<<"\n"<<c<<"\n"<<d;

}

};

void main()

{

clrscr();

sub s;

s.read();

s.display();

getch();

}

**Output:**



**Program 6(b):** Show the use of virtual function.

**Example:** Write a program to demonstrate the dynamic binding using virtual function.

**Coding:**

#include<iostream.h>

#include<conio.h>

class base

{

protected:

int a,b;

public:

virtual void read()

{

cout<<"Enter two values:";

cin>>a>>b;

}

virtual void display()

{

cout<<"\nThe values are :"<<a<<"\n"<<b<<endl;

}

};

class sub:public base

{

protected:

int c,d;

public:

virtual void read()

{

cout<<"Enter 4 values:";

cin>>a>>b>>c>>d;

}

virtual void display()

{

cout<<"\nThe values are :"<<a<<"\n"<<b<<"\n"<<c<<"\n"<<d<<endl;

}

};

void main()

{

clrscr();

base \*ptr;

base b;

sub s;

ptr=&b;

ptr->read();

ptr->display();

ptr=&s;

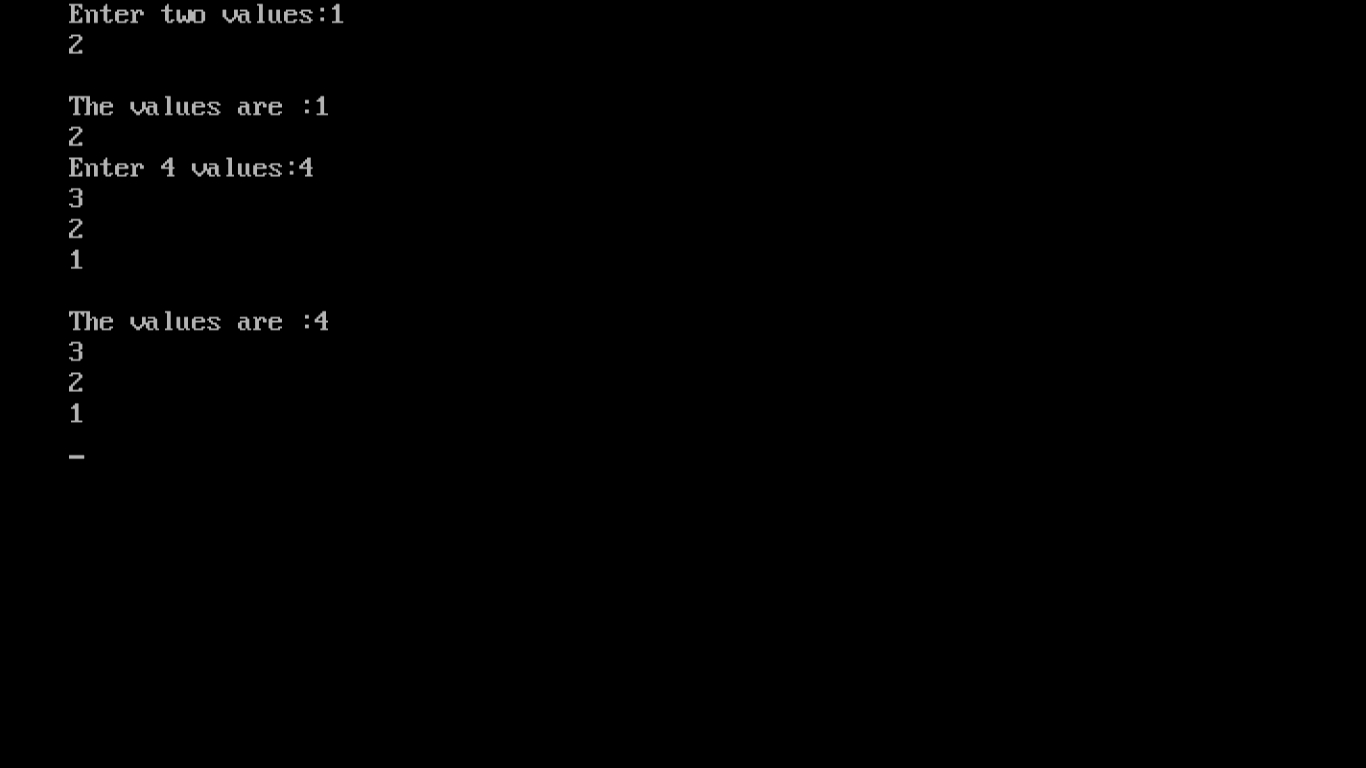
ptr->read();

ptr->display();

getch();

}

**Output:**



**Program 6(c):** Show the implementation of abstract classes.

**Example:** Write a program to demonstrate virtual class or abstract class.

**Coding:**

#include<iostream.h>

#include<conio.h>

class base

{

protected:

int a,b;

public:

void read()

{

cout<<"Enter two values:";

cin>>a>>b;

}

void display()

{

cout<<"The values are :"<<a<<"\n"<<b;

}

};

class sub:public virtual base

{

protected:

int c,d;

public:

void read()

{

cout<<"Enter 4 values:";

cin>>a>>b>>c>>d;

}

void display()

{

cout<<"The values are :"<<a<<"\n"<<b<<"\n"<<c<<"\n"<<d;

}

};

void main()

{

clrscr();

sub s;

s.read();

s.display();

getch();

}

**Output:**

