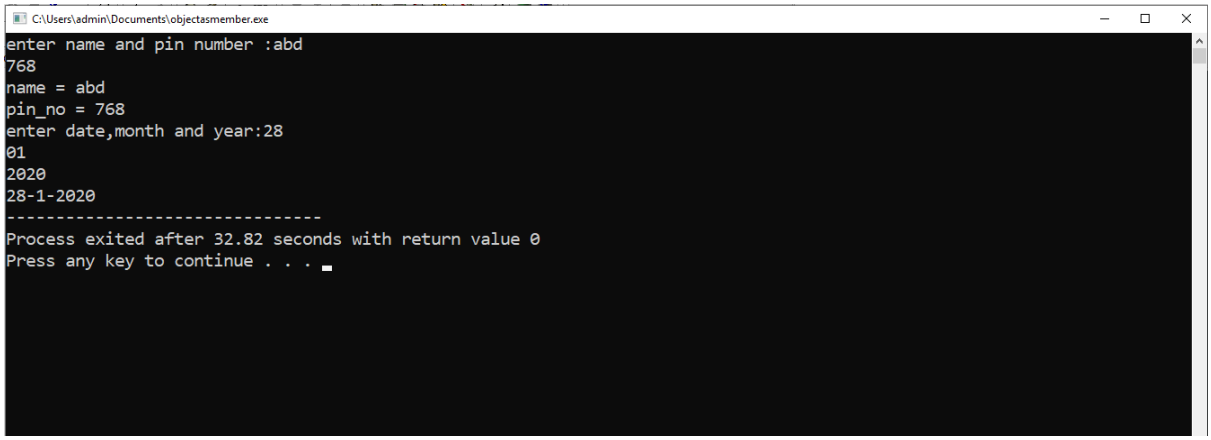
PROGRAM:

```
#include<iostream>
using namespace std;
class DOB
{
	private:
		int dd,mm,yy;
	public:
		void show()
		{
			cout<<"enter date,month and year:";
			cin>>dd>>mm>>yy;
			cout<<dd<<"-"<<mm<<"-"<<yy;
		}
};
class student
{
	private:
		string name;
		int pin_no;
		DOB d;
	public:
		void display()
		{
			cout<<"enter name and pin number :";;
			cin>>name>>pin_no;
			cout<<"name = "<<name<<endl;
			cout<<"pin_no = "<<pin_no<<endl;
			d.show();
		}
};
```

```
int main()
{
	student s;
	s.display();
	return 0;
}
```

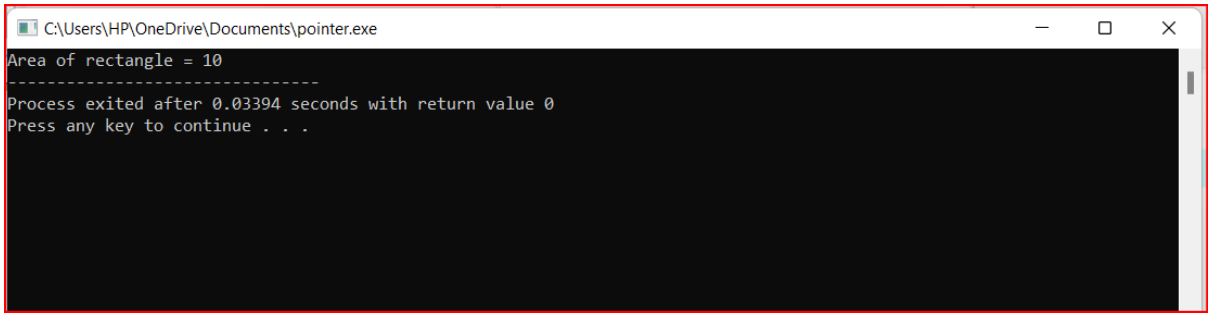
OUTPUT:



PROGRAM:

```
#include<iostream>
using namespace std;
class Rectangle
{
	private:
		int length,breadth;
	public:
		Rectangle(int l,int b)
		{
			length=l;
			breadth=b;
		}
		void getarea()
		{
			cout<<"Area of rectangle = "<<length*breadth;
		}
};
int main()
{
	Rectangle r(5,2),*rptr;
	rptr=&r;
	rptr->getarea();
	return 0;
}
```

OUTPUT:


C:\Users\HP\OneDrive\Documents\pointer.exe
Area of rectangle = 10

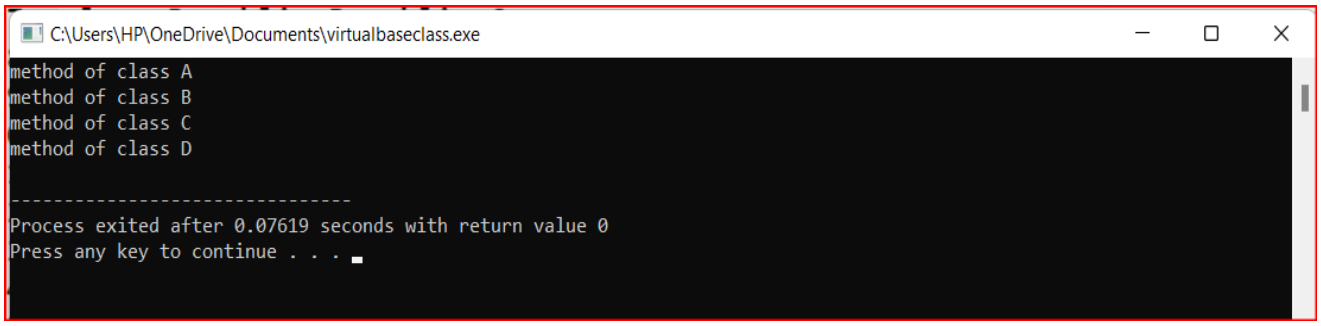
Process exited after 0.03394 seconds with return value 0
Press any key to continue . . .

PROGRAM:

```
#include<iostream>
using namespace std;
class A
{
	public:
		void showA()
		{
			cout<<"method of class A"<<endl;
		}
};
class B:virtual public A
{
	public:
		void showB()
		{
			cout<<"method of class B"<<endl;
		}
};
class C:virtual public A
{
	public:
		void showC()
		{
			cout<<"method of class C"<<endl;
		}
};
class D:public B,public C
{
	public:
		void showD()
```

```
        {
            cout<<"method of class D"<<endl;
        }
};
int main()
{
    D d;
    d.showA();
    d.showB();
    d.showC();
    d.showD();
    return 0;
}
```

OUTPUT:



PROGRAM:

```
#include <iostream>
using namespace std;
class base
{
        Public:
                virtual void print()
                {
                        cout << "print base class" << endl;
                 }
                 void show()
                {
                        cout << "show base class" << endl;
                }
};
class derived : public base {
public:
        void print()
        {
                cout << "print derived class" << endl;
        }

        void show()
        {
                cout << "show derived class" << endl;
        }
};
int main()
{
        base* bptr;
        derived d;
```

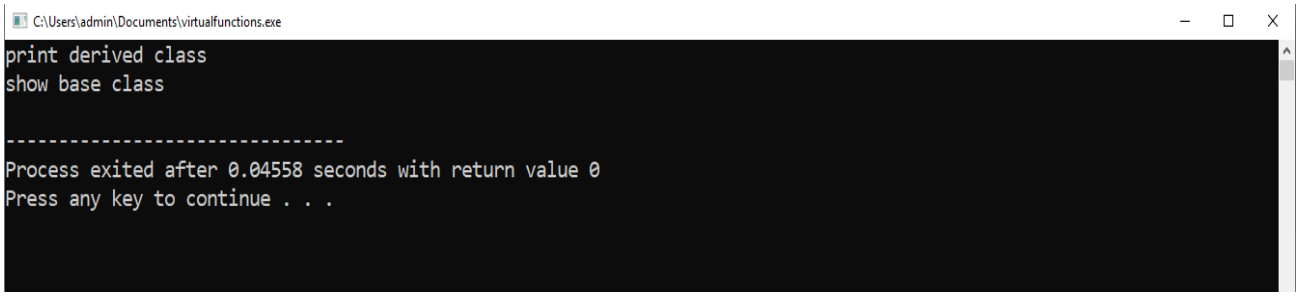
```
    bptr = &d;

    bptr->print(); //virtual function

    bptr->show();  // Non-virtual function


    return 0;
}
```

OUTPUT:



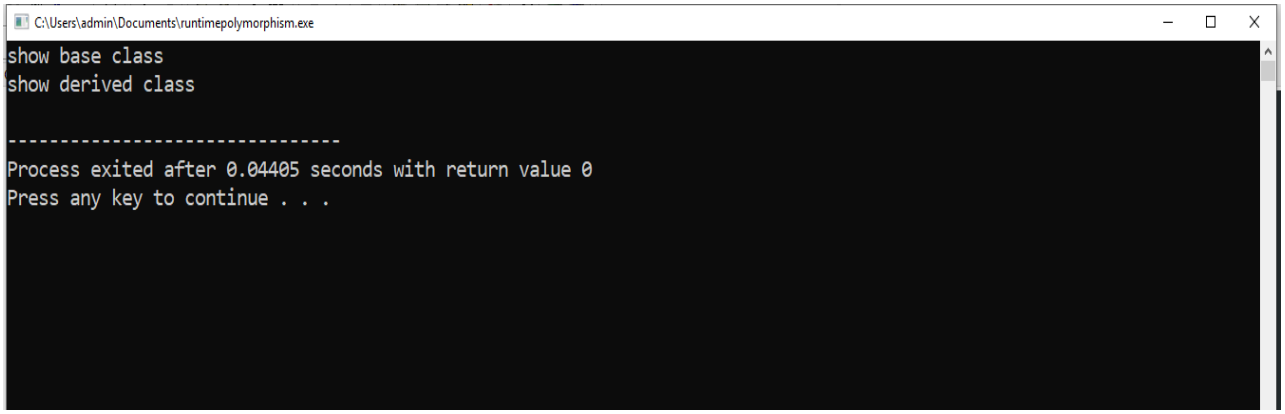
PROGRAM:

```
#include <iostream>
using namespace std;
class base {
public:
	virtual void show()
	{
		cout << "show base class" << endl;
	}
};

class derived : public base {
public:
	void show()
	{
		cout << "show derived class" << endl;
	}
};

int main()
{
	base b, *bptr;
	bptr = &b;
	bptr -> show();
	derived d, *dptr;
	bptr = &d;
	bptr -> show();
	return 0;
}
```

OUTPUT:


C:\Users\admin\Documents\runtimepolymorphism.exe
show base class
show derived class

Process exited after 0.04405 seconds with return value 0
Press any key to continue . . .

PROGRAM:

```
#include<iostream>
using namespace std;
class Shapes
{
	public:
		virtual void Area(int x)=0;
};
class Square:public Shapes
{
	public:
		void Area(int x)
		{
			cout<<"Area of square = "<<x*x<<endl;
		}
};
class circle:public Shapes
{
	public:
		void Area(int x)
		{
			cout<<"Area of circle = "<<3.14*x*x<<endl;
		}
};
class rectangle:public Shapes
{
	public:
		void Area(int x)
		{
			int b=2;
			cout<<"Area of rectangle = "<<x*b<<endl;
```

```
                    }
};
int main()
{
        circle c, *cptr;
        cptr = &c;
        cptr -> Area(4);
        Square s, *sptr;
        sptr = &s;
        sptr -> Area(3);
        rectangle r, *rptr;
        rptr = &r;
        rptr -> Area(5);
        return 0;
}
```

OUTPUT: