Single inheritance with constructor

```
#include<iostream.h>
#include<conio.h>
class circle
{
        int r;
        float area;
    public:
        circle(int a)
        {
                r=a;
        }
        void disp()
        {
                cout<<"area="<<3.14*r*r;
        }
};
class rectangle:public circle
{

        int l,b;
    public:
        rectangle(int a,int m,int n):circle(a)
    {
        l=m;
        b=n;
    }
    void result()
    {
        cout<<"area="<<l*b;
    }
};
void main()
{
        int a,c,d;
        clrscr();
        cout<<"enter radius of circle";
        cin>>a;
        cout<<"enter sides";
        cin>>c>>d;
        rectangle obj(a,c,d);
        cout<<"circle\n";
        obj.disp();
        cout<<"\n rectangle \n";
        obj.result();
}
```

Single inheritance without constructor

```
#include<iostream.h>
#include<conio.h>
class circle
{
        int r;
        float area;
    public:
        void get(int a)
        {
                r=a;
        }
        void disp()
        {
                cout<<"area="3.14*r*r;
        }
};
class rectangle::public circle
{

        int l,b;
    public:
    void init(int m,int n)
    {
        l=m;
        b=n;
    }
    void result()
    {
        cout<<"area="<<l*b;
    }
};
void main()
{
        int a,c,d;
        clrscr();
        cout<<"enter radius of circle";
        cin>>a;
        rectangle obj;
        obj.get(a);
        cout<<"enter sides";
        cin>>c>>d;
        obj.init(c,d);
        cout<<"circle\n";
        obj.disp();
        cout<<"\n rectangle \n";
        obj.result();
}
```