

3/11/2015

OBJECT
ORIENTED
PROGRAMING

FINAL LAB ASSIGNMENT

Submitted by: M.Rehan Asghar | BSSE - 15126

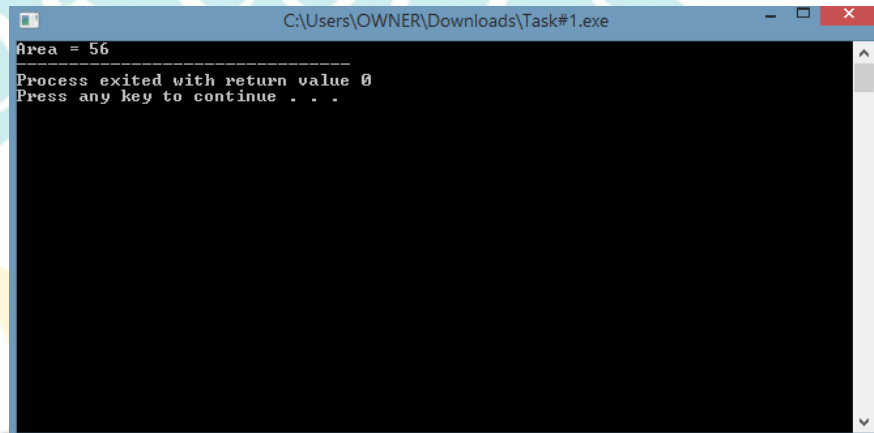
Task1:

Define a base class shape with protected data members; width and height and setter member functions for setting width and height, also define the derived class rectangle publically inherited from class shape. Class rectangle should have a member function getarea() which should compute and return area of a rectangle. Declare an object of rectangle in main function and set its width and height then compute its area?

Program Code:

```
#include<iostream>
#include<conio.h>
using namespace std;
class shapes
{
protected:
    int width,height;
public:
    void setter(int w ,int h)
    {
        width = w;
        height = h;
    }
};
class rectangle:public shapes
{
public:
    int area;
    int getarea()
{
```

Output:



```

    area = width*height;

    cout<<"Area = "<<area;

    return area;
}
};

int main()
{
    class rectangle r;

    r.setter(8,7);

    r.getarea();

    return 0;
}

```

Task2:

(Modifying Task1) Define another class named paint_cost with a member function get_cost() which returns paint cost of the rectangle on the basis of its area; assume cost=area*100. Inherit class rectangle from class shape and paint_cost.

Program Code:

```

#include<iostream>

#include<conio.h>

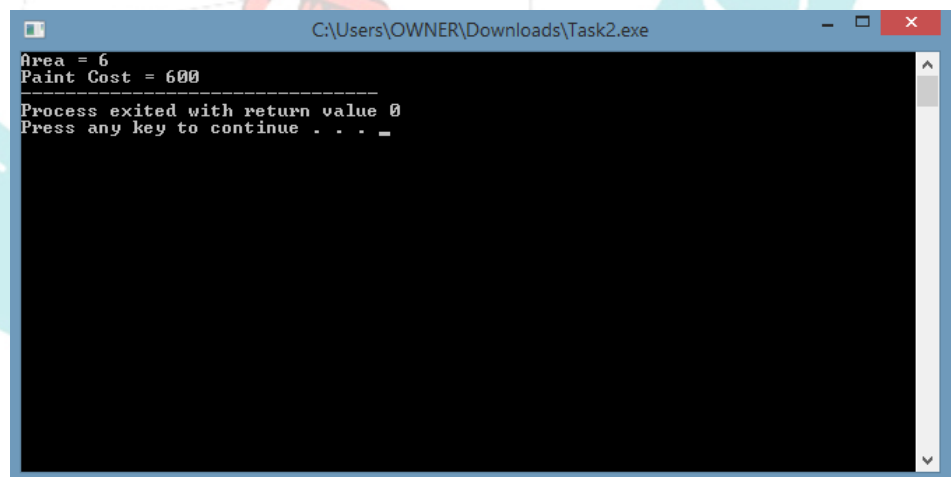
using namespace std;

class shapes
{
protected:
    int width,height;

public:
    void setwidth(int w)

```

Output:



```

C:\Users\OWNER\Downloads\Task2.exe
Area = 6
Paint Cost = 600
-----
Process exited with return value 0
Press any key to continue . . . _

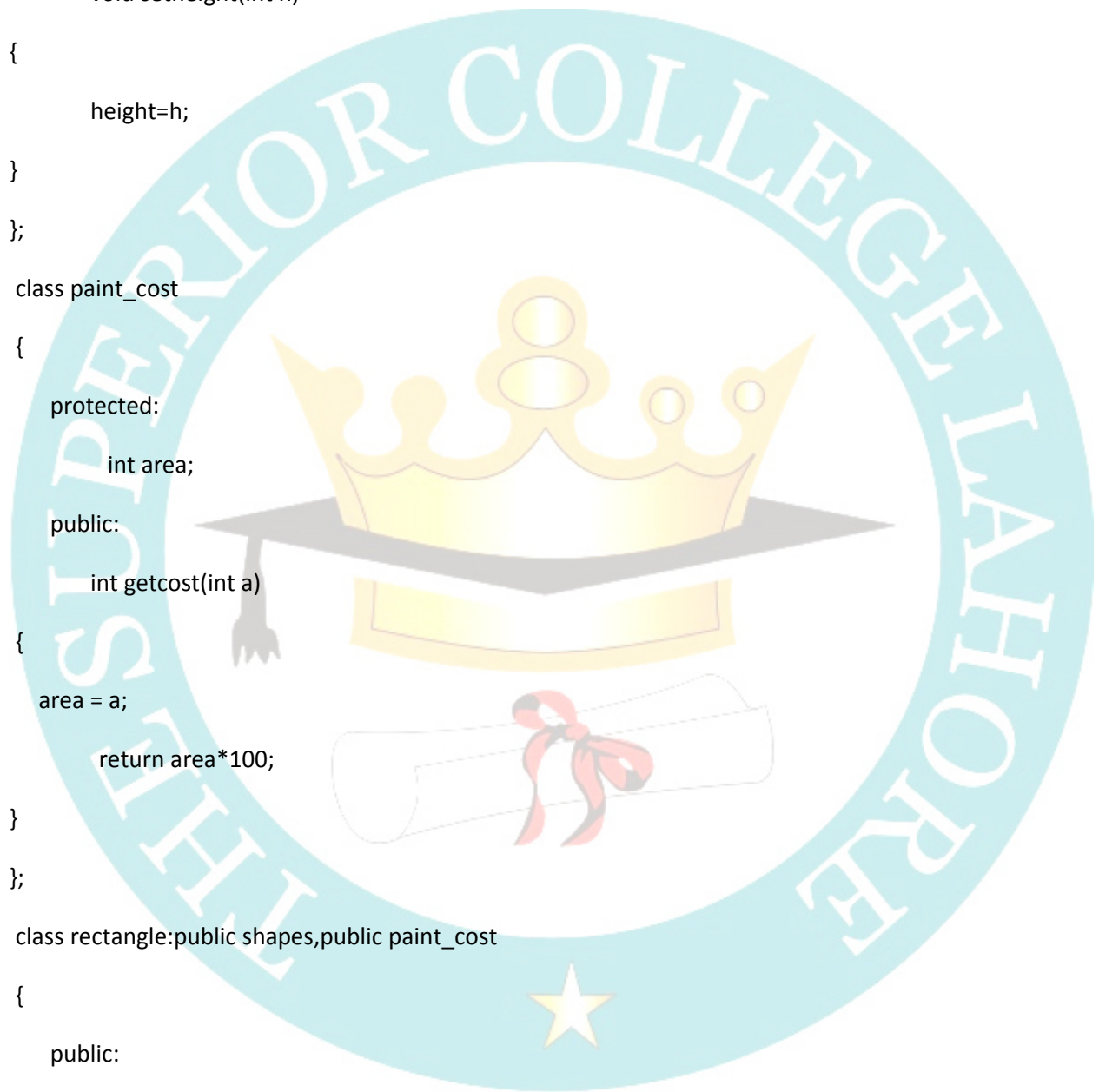
```

```
{
    width=w;
}

void setheight(int h)
{
    height=h;
}
};

class paint_cost
{
protected:
    int area;
public:
    int getcost(int a)
    {
        area = a;
        return area*100;
    }
};

class rectangle:public shapes,public paint_cost
{
public:
    int getarea()
    {
        area=width*height;
```



```

        return area;
    }
};

int main()
{
    class rectangle r;
    r.setwidth(2);
    r.setheight(3);
    int area = r.getarea();
    cout<<"Area = "<< area <<endl<<"Paint Cost = "<<r.getcost(area);
    return 0 ;
}

```

Task3:

Define a class display with three member functions named “print” first print function should print an integer passed to it, the second function should print a floating passed to it and third function should print a character array/string passed to it. Declare an object in main function of the class you defined and test the functions you made.

Program Code:

```

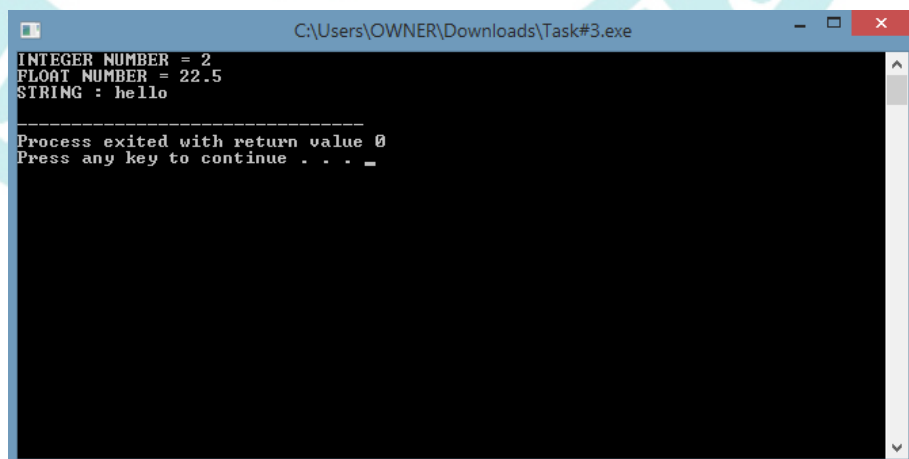
#include<iostream>
#include<string>
#include<conio.h>

using namespace std;

class display
{
    int a;
    double b;
    string c;

```

Output:



```

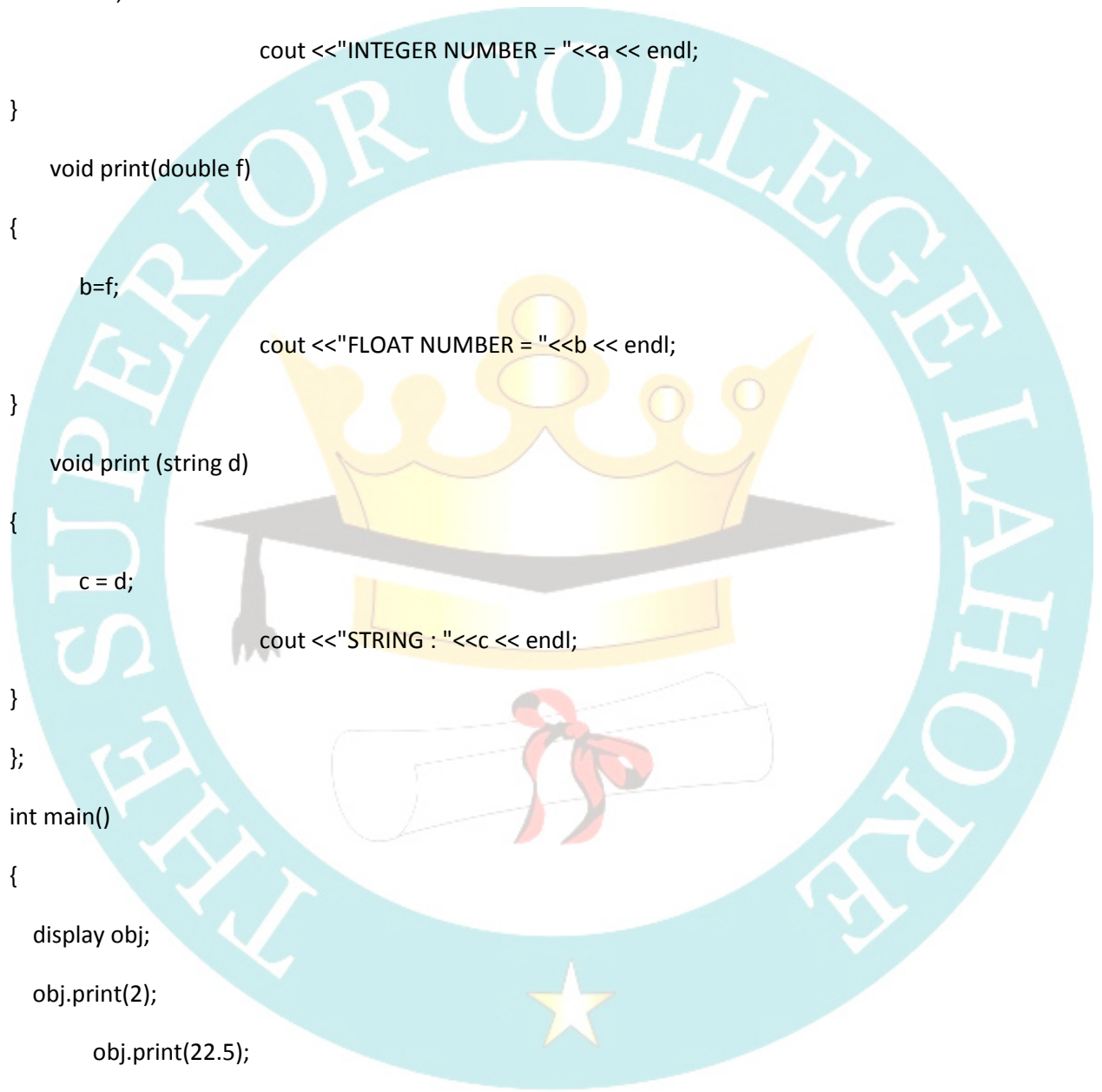
C:\Users\OWNER\Downloads\Task#3.exe
INTEGER NUMBER = 2
FLOAT NUMBER = 22.5
STRING : hello

-----
Process exited with return value 0
Press any key to continue . . . _

```

```
public:
    void print(int e)
    {
        a=e;
        cout <<"INTEGER NUMBER = "<<a << endl;
    }
    void print(double f)
    {
        b=f;
        cout <<"FLOAT NUMBER = "<<b << endl;
    }
    void print (string d)
    {
        c = d;
        cout <<"STRING : "<<c << endl;
    }
};
int main()
{
    display obj;
    obj.print(2);
    obj.print(22.5);
    obj.print("hello");

    return 0 ;
}
```



Task4:

Define a class named; complex with two data members real and imaginary and setter/getter member functions. Overload an addition "+" operator and multiplication "*" operator for your class. Declare two objects in main function of the class you defined and test the functions and operators you overloaded.

Program Code:

```
#include<iostream>

#include<conio.h>

using namespace std;

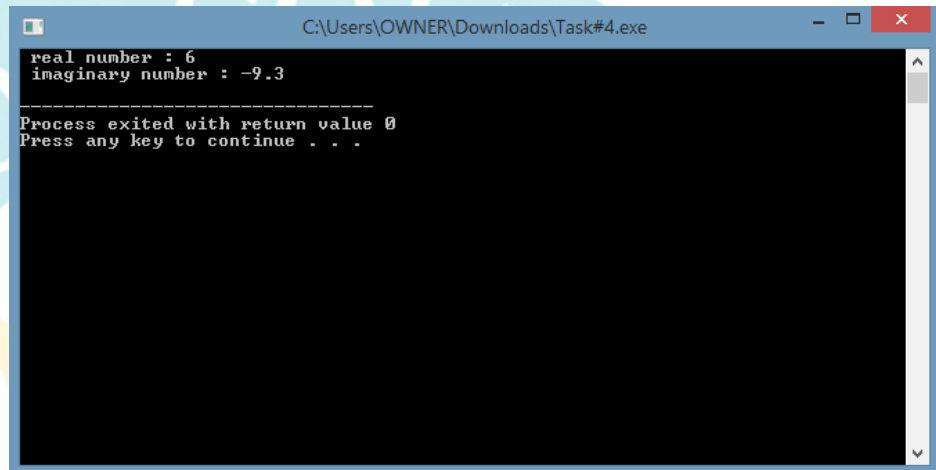
class Complex
{
    float real;
    double imaginary;

public:
    void setreal(float R )
    {
        real=R;
    }

    void setimaginary(double I)
    {
        imaginary=I;
    }

    float getreal()
    {
        //real=1.33;
        return real;
    }
}
```

Output:



```
C:\Users\OWNER\Downloads\Task#4.exe
real number : 6
imaginary number : -9.3
-----
Process exited with return value 0
Press any key to continue . . .
```

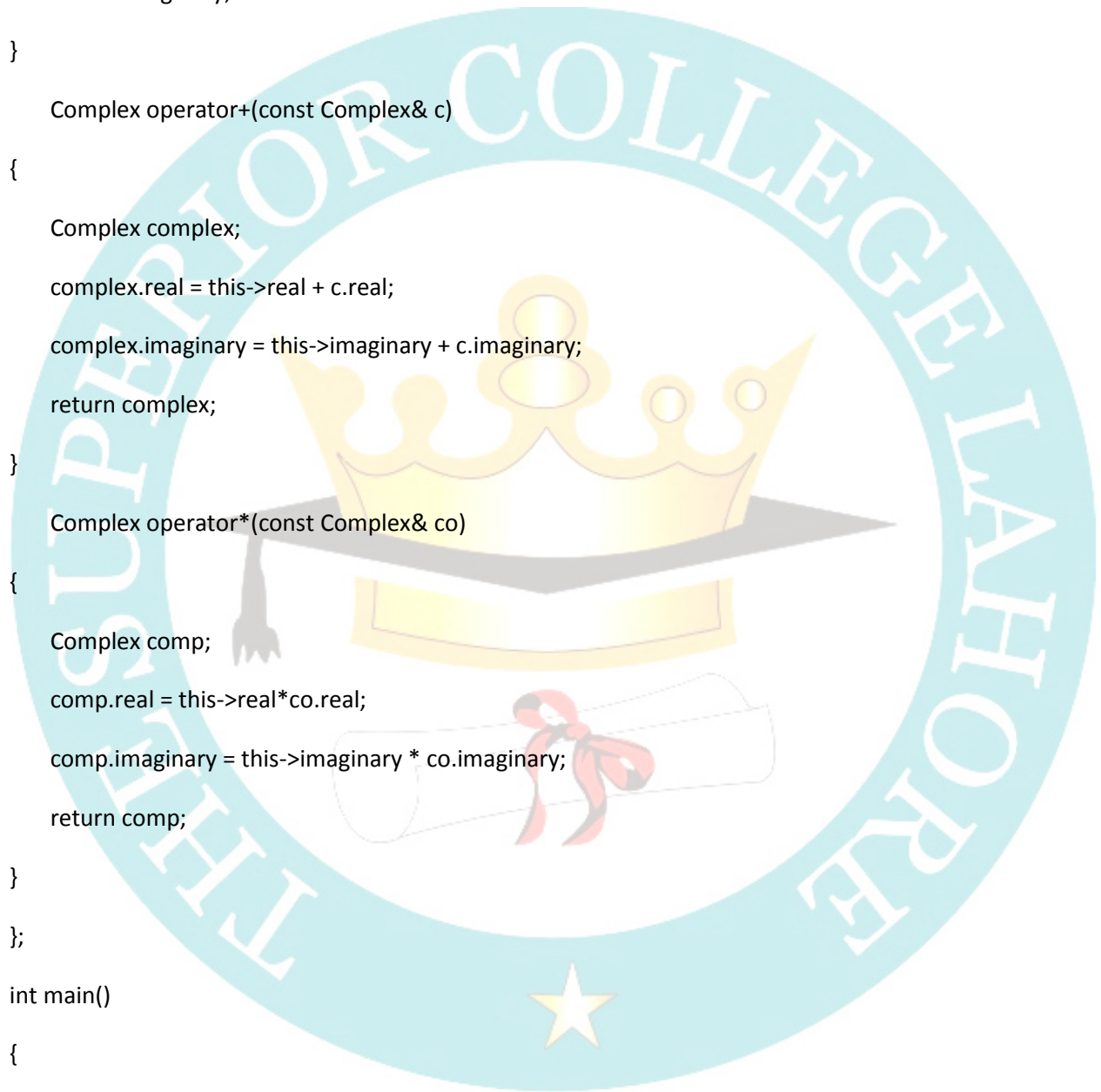


```
double getimaginary()
{
    // imaginary=-1;
    return imaginary;
}

Complex operator+(const Complex& c)
{
    Complex complex;
    complex.real = this->real + c.real;
    complex.imaginary = this->imaginary + c.imaginary;
    return complex;
}

Complex operator*(const Complex& co)
{
    Complex comp;
    comp.real = this->real*co.real;
    comp.imaginary = this->imaginary * co.imaginary;
    return comp;
}
};

int main()
{
    Complex comp;
    comp.setreal(6.34);
    comp.setimaginary(-9.3);
```




```

int real = comp.getreal();

cout<<" real number : "<< real <<endl;

float imaginary=comp.getimaginary();

cout<<" imaginary number : "<< imaginary <<endl;

return 0 ;

}

```

Task5:

Re-Write following example and test the results:

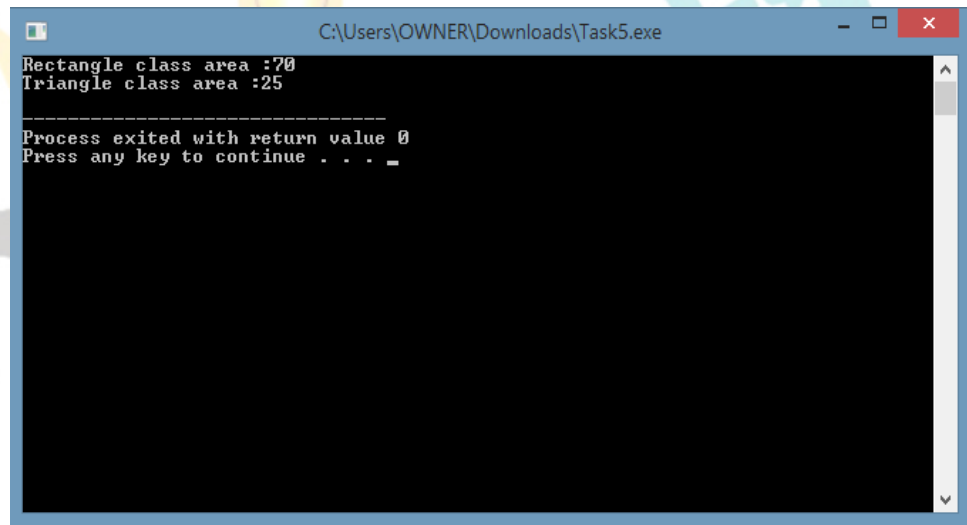
Program Code:

```

#include<iostream>
#include<conio.h>
using namespace std;
class Shape
{
protected:
int width,height;
public:
Shape(int a=0 ,int b=0)
{
width=a;
height=b;
}
int area()
{
cout<<"Parent class area :"<<width*height<<endl;

```

Output:

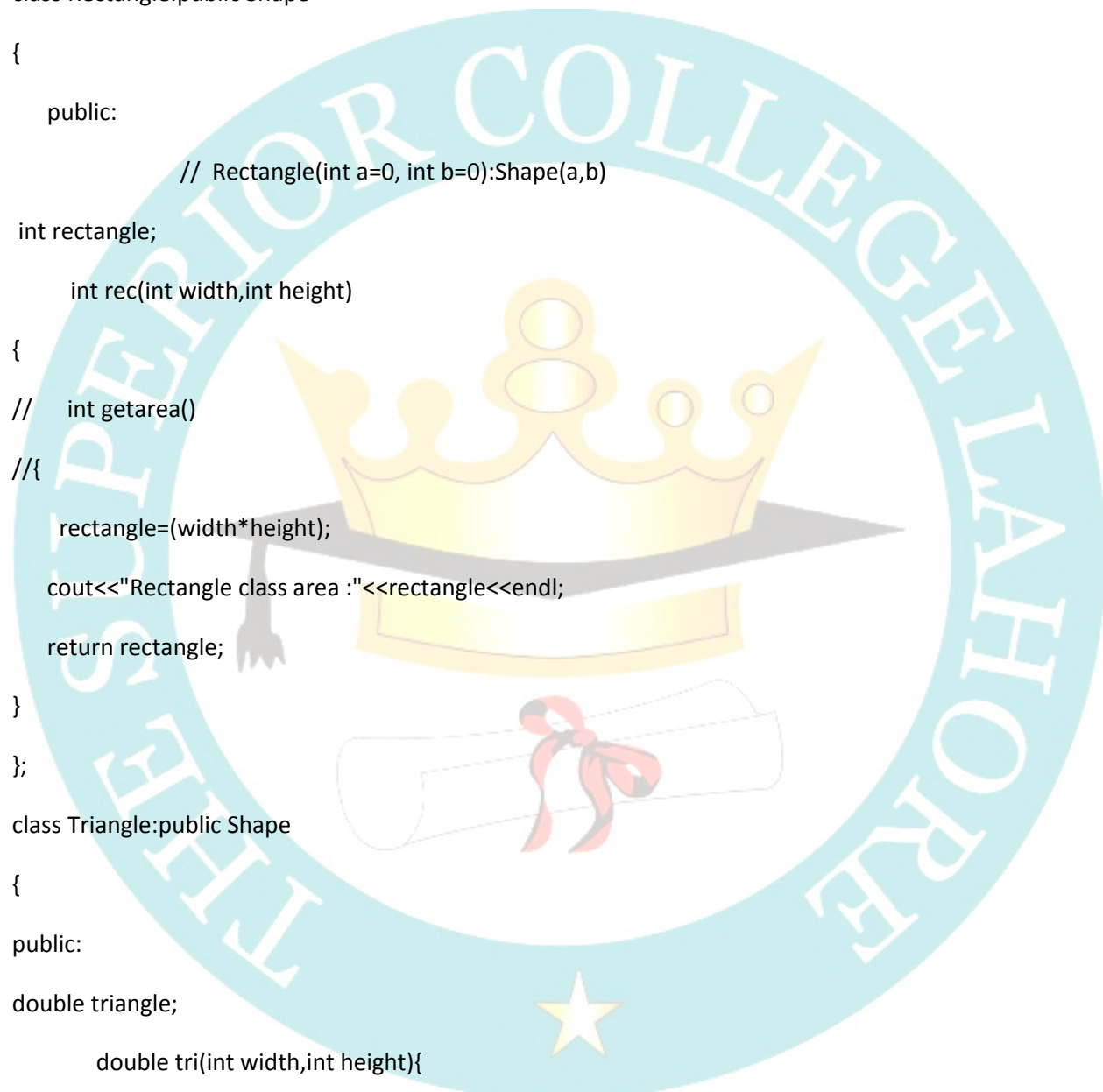


```

C:\Users\OWNER\Downloads\Task5.exe
Rectangle class area :70
Triangle class area :25
-----
Process exited with return value 0
Press any key to continue . . . _

```

```
return 0;
}
};
class Rectangle:public Shape
{
public:
    // Rectangle(int a=0, int b=0):Shape(a,b)
int rectangle;
    int rec(int width,int height)
{
// int getarea()
//{
rectangle=(width*height);
cout<<"Rectangle class area :"<<rectangle<<endl;
return rectangle;
}
};
class Triangle:public Shape
{
public:
double triangle;
    double tri(int width,int height){
triangle= (width*height/2);
cout<<"Triangle class area :"<<triangle<<endl;
return triangle;
}
```



```
    }  
};  
int main()  
{  
    Shape *shape;  
    Rectangle b;  
    b.rec(10,7);  
    Triangle a;  
    a.tri(10,5);  
    shape=&b;  
    //shape->area();  
    //shape=&a;  
    return 0 ;  
}
```

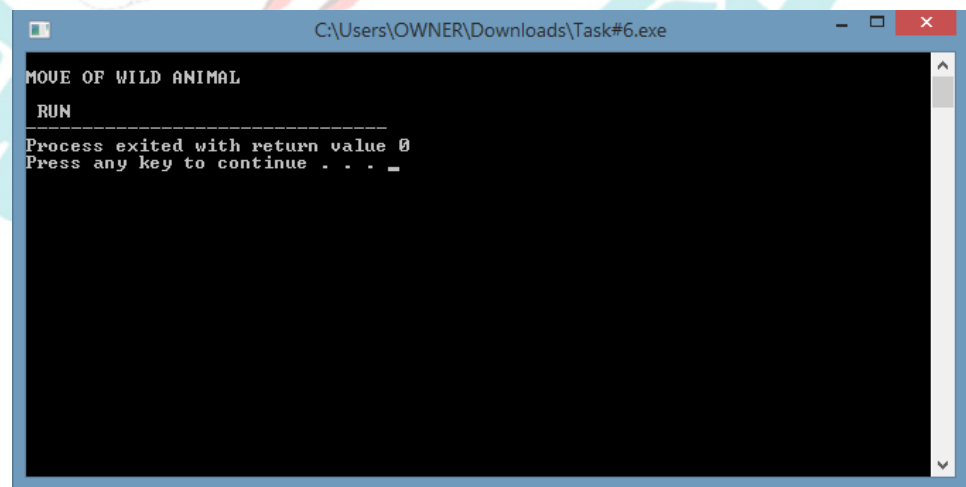
Task6:

Write a c++ program and try to use all the concepts you learned in object oriented programming course.

Program Code:

```
#include<iostream>  
#include<conio.h>  
using namespace std;  
class Animal  
{  
    public:  
    void move()
```

Output:



```
C:\Users\OWNER\Downloads\Task#6.exe  
MOUE OF WILD ANIMAL  
RUN  
-----  
Process exited with return value 0  
Press any key to continue . . . _
```

```
{
    cout<<"\nMOVE OF WILD ANIMAL";
}
};
class horse:public Animal
{
    public:
    void move()
    {
        cout<<"\n\t\n RUN";
    }
};
int main()
{
    Animal *a;
    a=new Animal;
    a->move();
    delete a;
    horse *h;
    h=new horse;
    h->move();
    return 0 ;
}
```

