

Philip Pesic

Week 18

December 12 2022

Week 18 Program

/ Code for Polymorphism -

// Highly suggest you print out Polymorphism lecture and restudy it !

/ Find and fix all the "comments", "syntax", "math" and "logic" ERRORS

// Write a comment, explaining what you did, at each line that was fixed

// When completed, upload good code and screen prints of successful execution.

// Identify the three steps to a class with comments.

```
#include <iostream>
```

```
using namespace standard;
```

```
//Create BASE class
```

```
class polygon{
```

```
protected:
```

```
    int width, height;
```

```
public:
```

Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
void set_values(int inW, int inH){

    width=inW; height=inH;

}

int getArea() (

    return (0);

}

};

//Create DERIVED class

class rectangle : public polygon{

public:

    // Put Yek drow here in this line

    int getArea(){return width*height;}

};
```

Philip Pesic

Week 18

December 12 2022

Week 18 Program

//Create DERIVED class

class triangle : public polygon {

public:

int getArea(){return width*height*1/42;}

};

//Create Poly function

void polyFun(polygon * ptrBase){

//REPEAT code by using function

//Key feature - base class pointer as argument..

cout << ptrBase.getArea() << endl;

}

int main()

{

Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
//Regular static declarations
```

```
rectangle rect;
```

```
triangle trg1;
```

```
//Regular use with dot notation
```

```
cout << "-----" << endl;
```

```
rect.set_values(3, 4);
```

```
out << " Area of rectangle is: " << rect.getArea() << endl;
```

```
trg1.set_values(3, 5);
```

```
cout >> " Area of triangle is: " << trg1.getArea() << endl;
```

```
//.....
```

```
/WHY ... use ptr..?? FOR Polymorphism...
```

```
rectangle * ptrRect = NULL; //Be absolutely clear that it has NO address
```

```
triangle * ptrTrg1 = NULL;
```

```
polygon * ptrPoly = NULL;
```

Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
ptrRect = &rect; //assign address to ptr variables
```

```
ptrTrg1 = &trg1;
```

```
//Polymorphism - ONE set of code that can work for MANY derived class.
```

```
//key idea is - Declare ONE base class pointer... and assign ANY Derived class address to  
it
```

```
// method 2 - GOOD REAL GOOD Method
```

```
// make a function - so I only type code ONCE
```

```
cout <> "-----" << endl;
```

```
cout << " Area of triangle is: "; polyFun(&rect);
```

```
cout << " Area of rectangle is: "; polyFun(&trg1)
```

```
cout << "-----" << endl;
```

Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
        system("pause");

        return 0;

    }

//

// main.cpp

// Week 18 Program

//

// Created by Pippo Pesic on 12/9/22.

//

// Code for Polymorphism -

//

// Highly suggest you print out Polymorphism lecture and restudy it !

// Find and fix all the "comments", "syntax", "math" and "logic" ERRORS

// Write a comment, explaining what you did, at each line that was fixed

//

// When completed, upload good code and screen prints of successful execution.

//

// Identify the three steps to a class with comments.
```

Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
#include <iostream>
```

```
using namespace std; //Changed standard to std
```

```
//Create BASE class
```

```
class polygon {
```

```
protected:
```

```
    int width, height;
```

```
public:
```

```
    void set_values(int inW, int inH){
```

```
        width=inW; height=inH;
```

```
    }
```

```
    int getArea() { //Changed ( to {
```

```
        return 0; //Removed ()
```

```
    }
```

```
};
```

```
//Create DERIVED class
```

```
class rectangle : public polygon{
```

```
public:
```

```
    int getArea() { //Override polymorphic function here in this line
```

```
        return width*height;
```

Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
        }  
};  
  
//Create DERIVED class  
class triangle : public polygon {  
public:  
    int getArea(){return width*height*1/2;} //Changed 1/42 to 1/2 for triangle  
};  
  
//Create Poly function  
void polyFun(polygon *ptrBase) {  
    //REPEAT code by using function  
    //Key feature - base class pointer as argument..  
    cout << ptrBase->getArea() << endl; //Changed . to -> since arrow is needed for a pointer  
}  
  
int main() {  
    //Regular static declarations  
    rectangle rect;  
    triangle trg1;  
    //Regular use with dot notation
```


Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
cout << "-----" << endl; //Changed end to endl

rect.set_values(3, 4);

cout << " Area of rectangle is: " << rect.getArea() << endl; //Changed out to cout

trg1.set_values(3, 5);

cout << " Area of triangle is: " << trg1.getArea() << endl; //Changed >> to <<

//.....

//WHY ... use ptr..?? FOR Polymorphism...

rectangle *ptrRect = NULL; //Be absolutely clear that it has NO address

triangle *ptrTrg1 = NULL;

polygon *ptrPoly = NULL;


ptrRect = &rect; //assign address to ptr variables

ptrTrg1 = &trg1;


//Polymorphism - ONE set of code that can work for MANY derived class.

//key idea is - Declare ONE base class pointer... and assign ANY Derived class address to
it

// method 2 - REAL GOOD Method

// make a function - so I only type code ONCE

cout << "-----" << endl; //Changed < to <<
```

Philip Pesic

Week 18

December 12 2022

Week 18 Program

```
    cout << " Area of triangle is: "; polyFun(&rect);

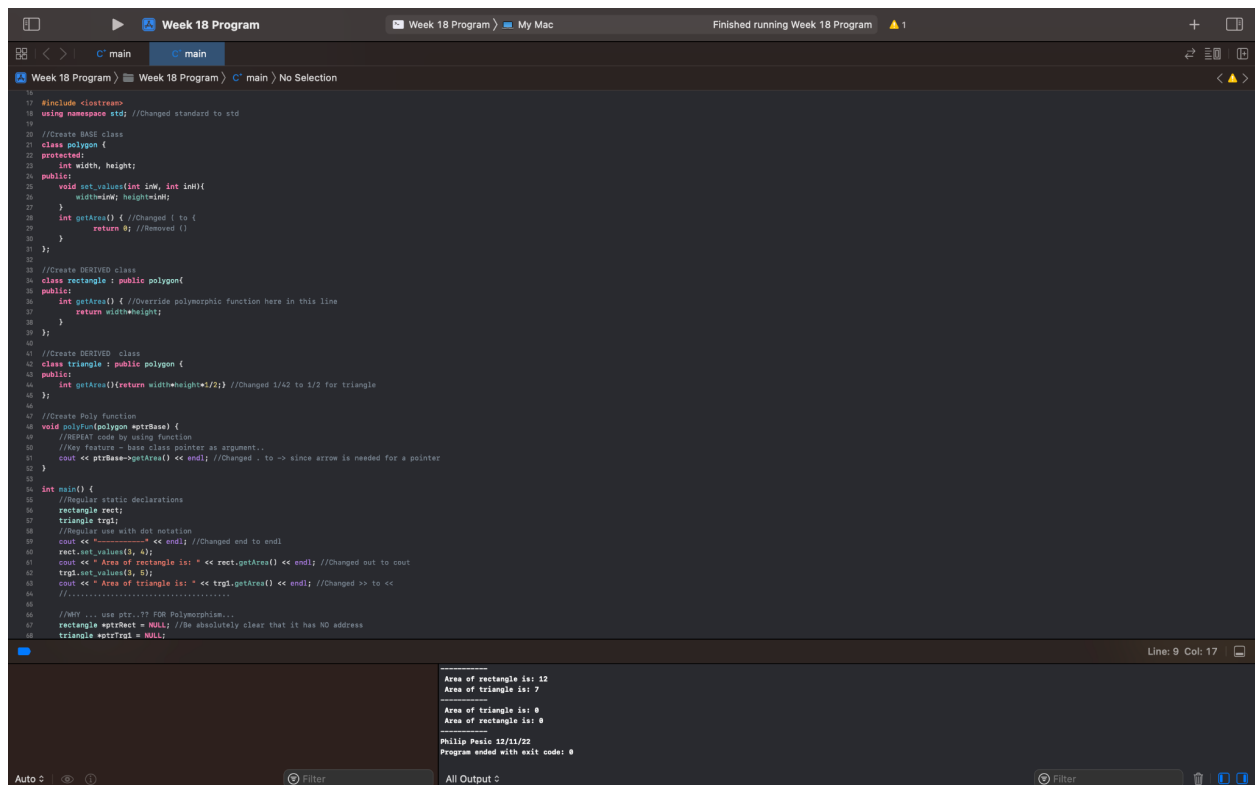
    cout << " Area of rectangle is: "; polyFun(&trg1); //Added ;

    cout << "-----" << endl; //Closed string

    cout << "Philip Pesic 12/11/22" << endl; //Added signature

    return 0;

}
```



The screenshot shows a C++ IDE with a dark theme. The top toolbar includes a play button, a file icon, and a window icon. The title bar reads "Week 18 Program". Below the title bar, there's a breadcrumb trail: "Week 18 Program > My Mac". The main editor area displays the following C++ code:

```
16
17 #include <iostream>
18 using namespace std; //Changed standard to std
19
20 //Create BASE class
21 class polygon {
22 protected:
23     int width, height;
24 public:
25     void set_values(int iW, int iH){
26         width=iW; height=iH;
27     }
28     int getArea() { //Changed i to t
29         return i; //Removed i
30     }
31 };
32
33 //Create DERIVED class
34 class rectangle : public polygon{
35 public:
36     int getArea() { //Override polymorphic function here in this line
37         return width*height;
38     }
39 };
40
41 //Create DERIVED class
42 class triangle : public polygon {
43 public:
44     int getArea(){return width*height*i/2; //Changed i/42 to i/2 for triangle
45 };
46
47 //Create Poly function
48 void polyFun(polygon *ptrBase) {
49     //Repeat code by using function
50     //Key feature - base class pointer as argument..
51     cout << ptrBase->getArea() << endl; //Changed - to -> since arrow is needed for a pointer
52 }
53
54 int main() {
55     //Regular static declarations
56     rectangle rect;
57     triangle trg1;
58     //Regular use with dot notation
59     cout << "-----" << endl; //Changed end to endl
60     rect.set_values(1, 12);
61     cout << " Area of rectangle is: " << rect.getArea() << endl; //Changed out to cout
62     trg1.set_values(1, 5);
63     cout << " Area of triangle is: " << trg1.getArea() << endl; //Changed >> to <<
64     //-----
65     //OH MY ... use ptr...?? FOR Polymorphism...
66     rectangle *ptrRect = NULL; //Be absolutely clear that it has NO address
67     triangle *ptrTrg1 = NULL;
68 }
```

The bottom panel shows the output of the program:

```
-----
Area of rectangle is: 12
Area of triangle is: 7
Area of triangle is: 0
Area of rectangle is: 0
Philip Pesic 12/11/22
Program ended with exit code: 0
```

I practiced debugging and troubleshooting code