Lab 12 - Polymorphism, Virtual Functions

Start Assignment

Due Monday by 11:59pm **Points** 20 **Submitting** a file upload

Advanced C++ Programming

Module 12 – Dynamic or Runtime

Polymorphism / Virtual Functions

(25 points)

Perform this lab individually

DYNAMIC OR RUNTIME
POLYMORPHISM
AND
VIRTUAL FUNCTIONS IN C++

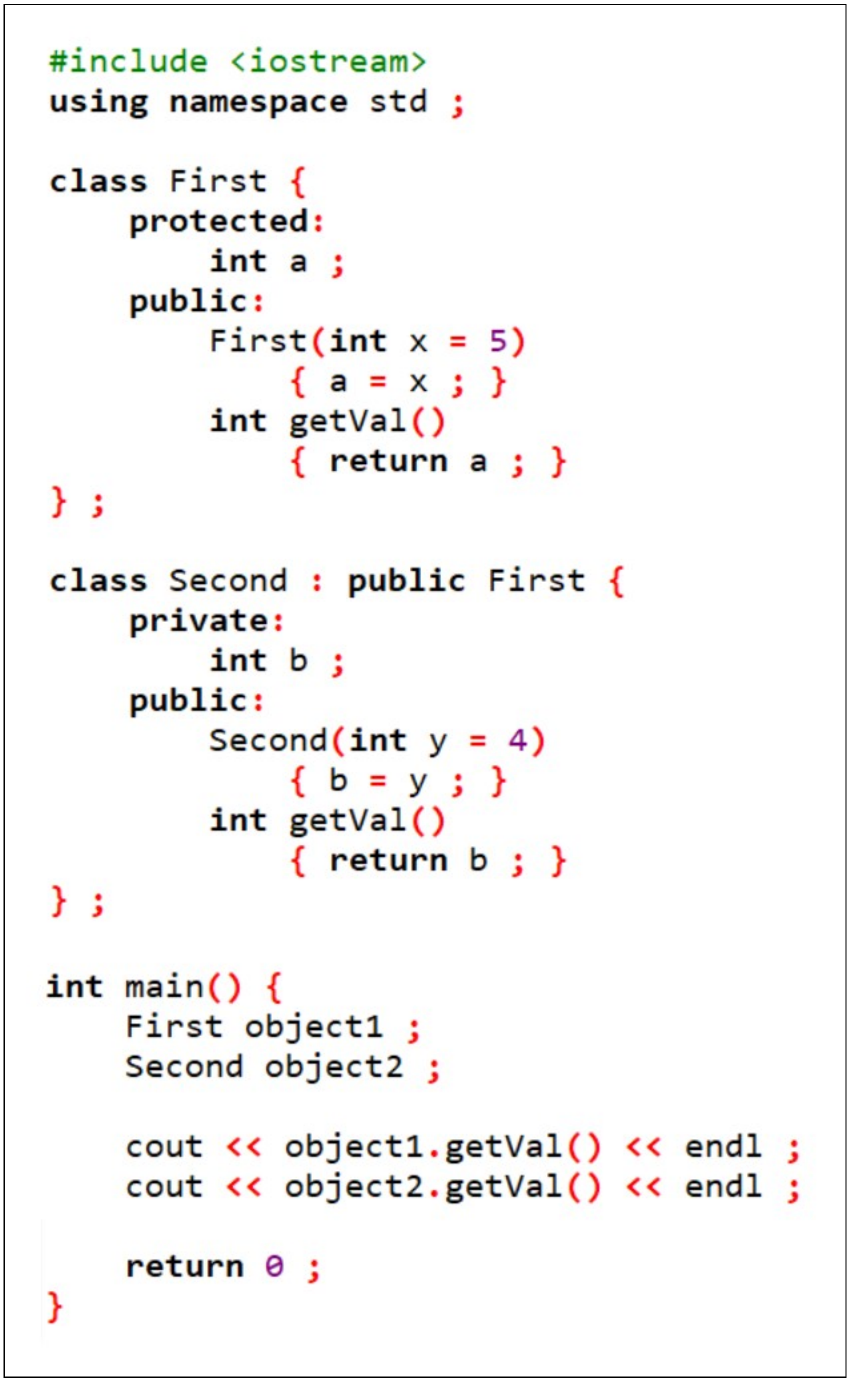
Summary

Determine the output of several (slightly modified) Checkpoints from Chapter 15 of the textbook.

Each program is worth 5 points.

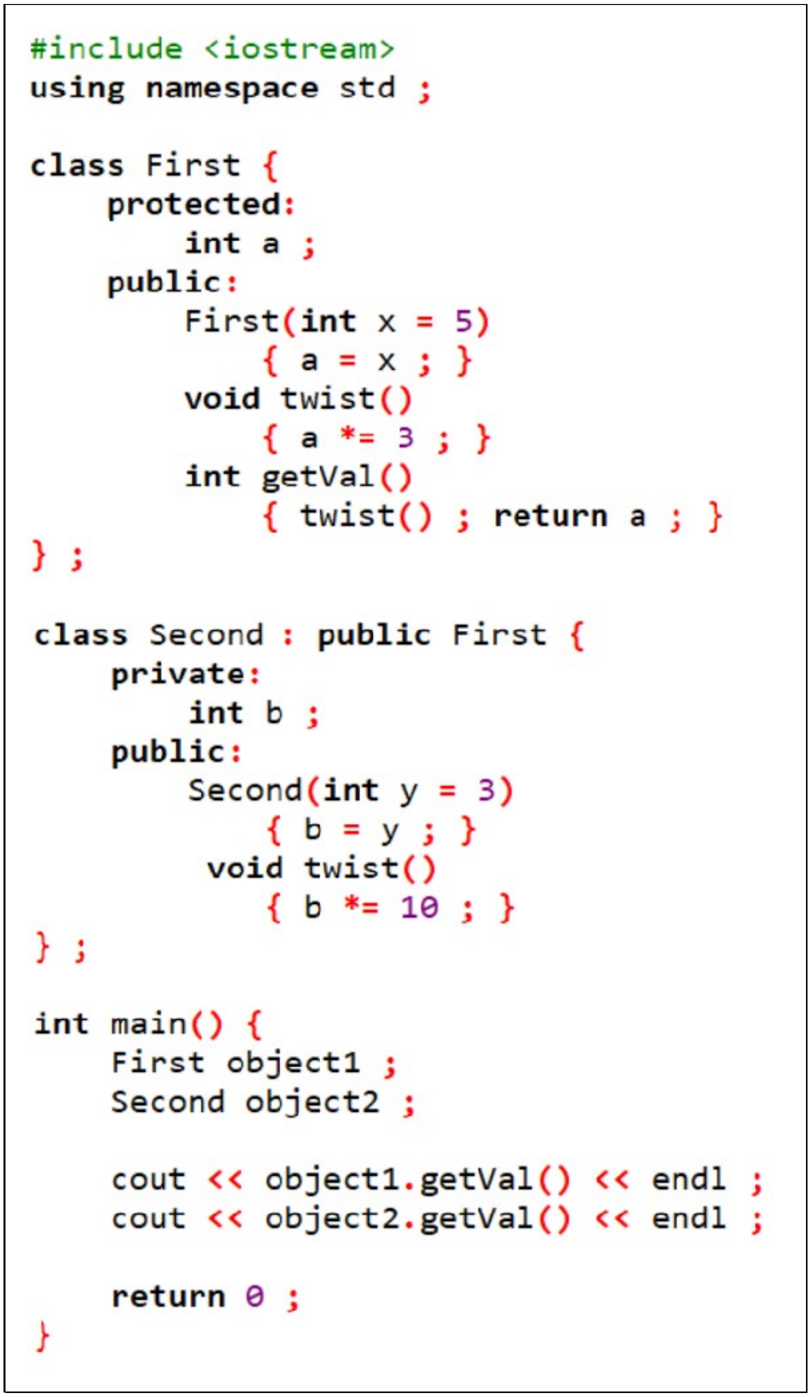
(Note: your answers are final, no re-submissions on this lab.)

Program 1



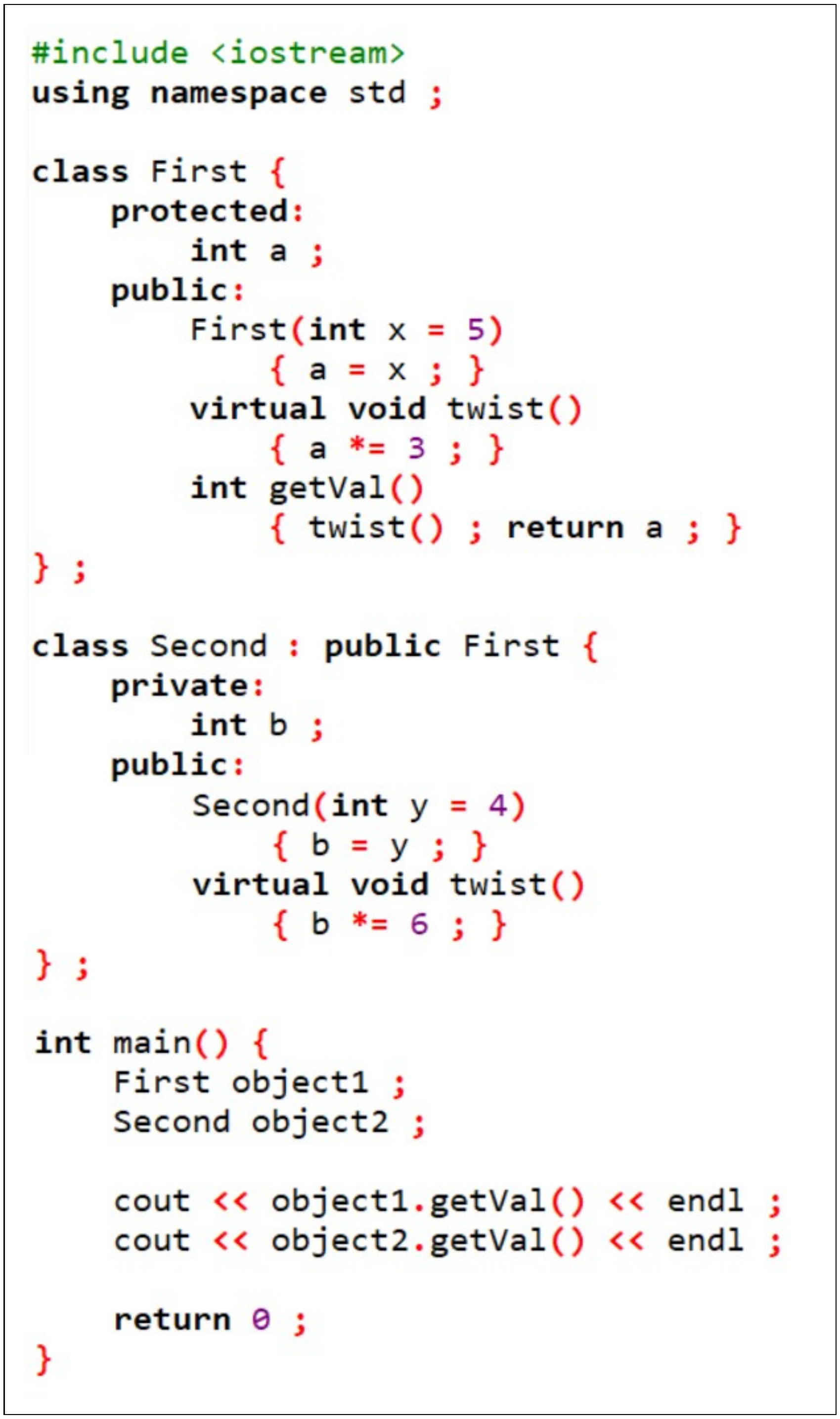
```
#include <iostream>
using namespace std ;

class First {
    protected:
        int a ;
    public:
        First(int x = 5)
            { a = x ; }
        int getVal()
            { return a ; }
} ;

class Second : public First {
    private:
        int b ;
    public:
        Second(int y = 4)
            { b = y ; }
        int getVal()
            { return b ; }
} ;

int main() {
    First object1 ;
    Second object2 ;

    cout << object1.getVal() << endl ;
    cout << object2.getVal() << endl ;

    return 0 ;
}
```

Program 2



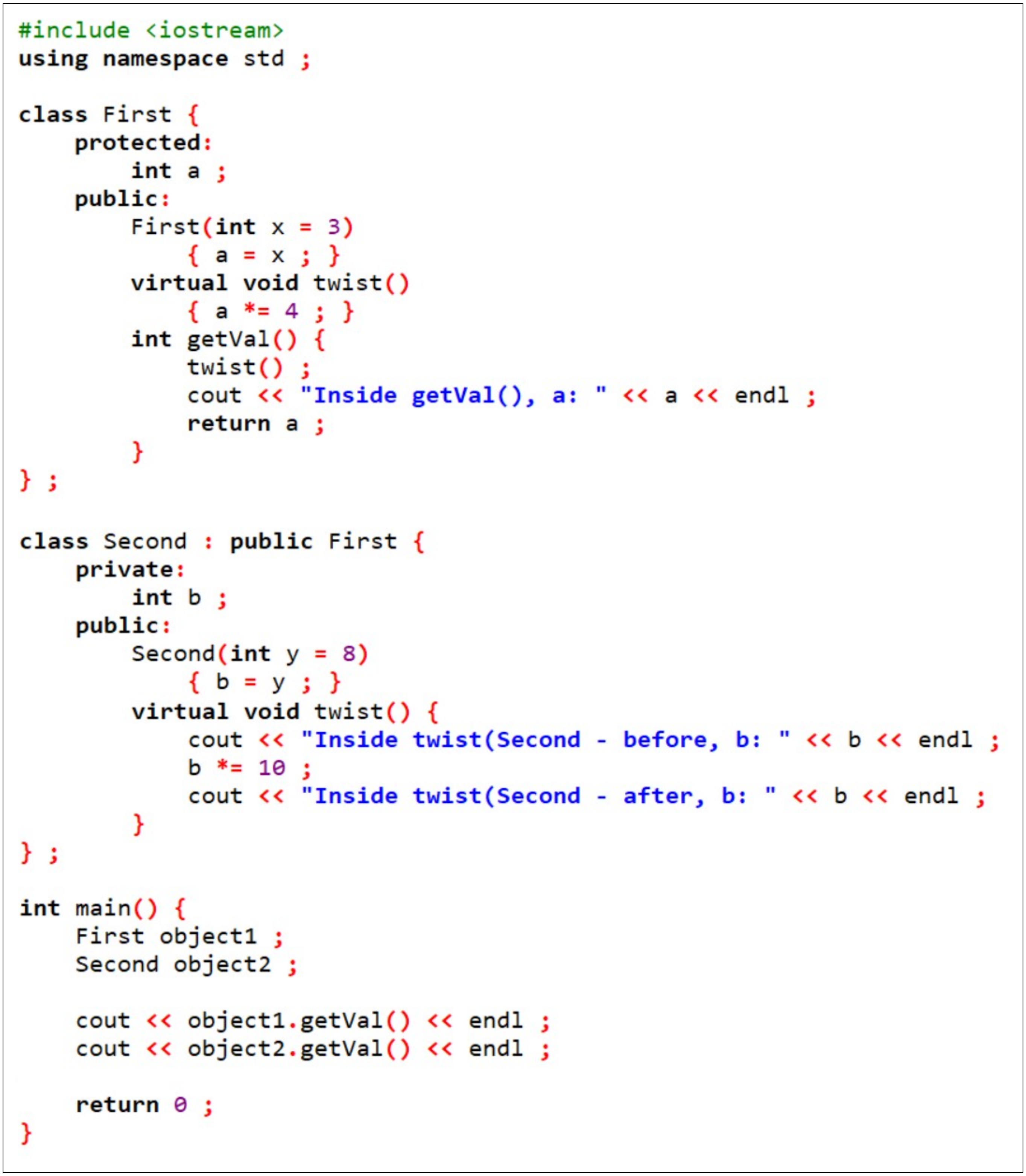
```
#include <iostream>
using namespace std ;

class First {
    protected:
        int a ;
    public:
        First(int x = 5)
            { a = x ; }
        void twist()
            { a *= 3 ; }
        int getVal()
            { twist() ; return a ; }
} ;

class Second : public First {
    private:
        int b ;
    public:
        Second(int y = 3)
            { b = y ; }
         void twist()
            { b *= 10 ; }
} ;

int main() {
    First object1 ;
    Second object2 ;

    cout << object1.getVal() << endl ;
    cout << object2.getVal() << endl ;

    return 0 ;
}
```

Program 3



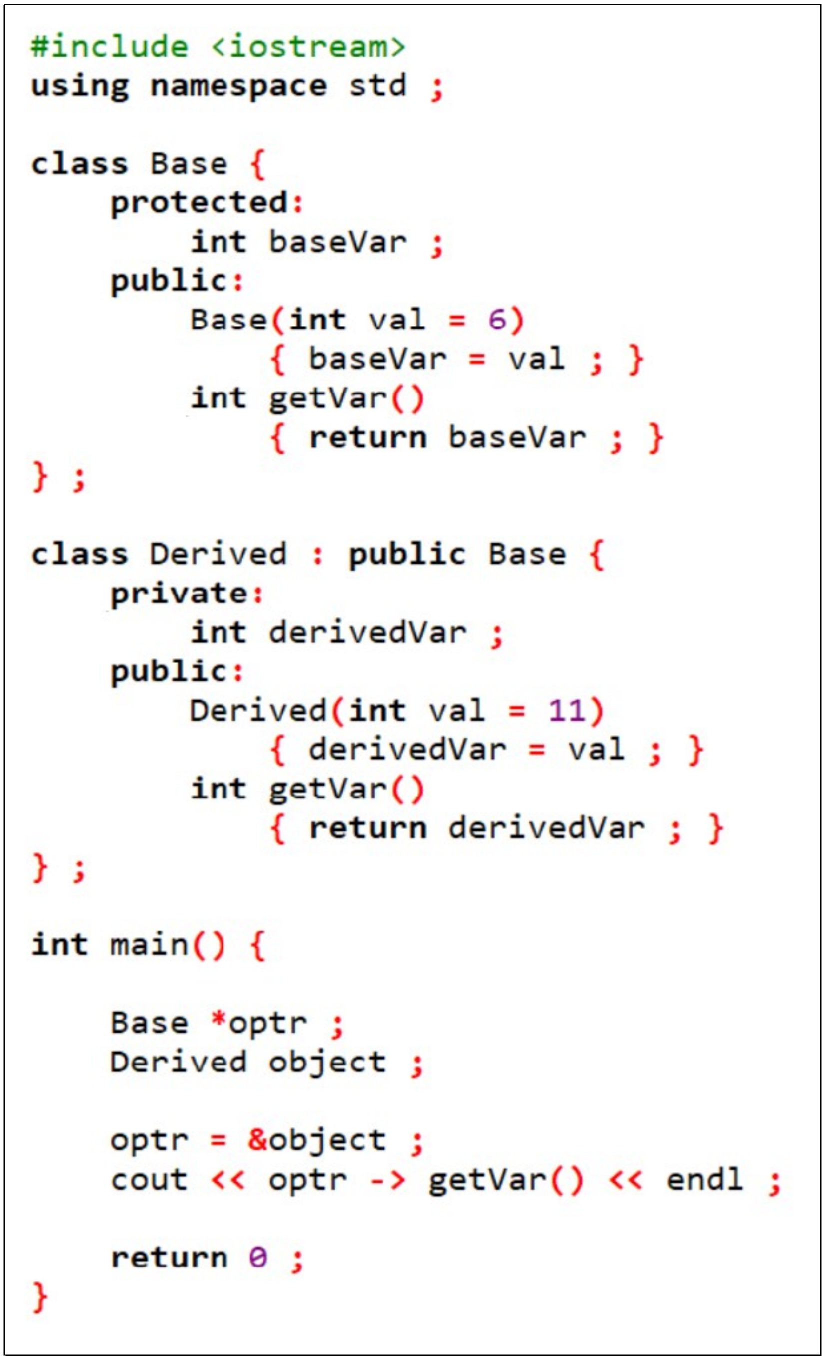
```
#include <iostream>
using namespace std ;

class First {
    protected:
        int a ;
    public:
        First(int x = 5)
            { a = x ; }
        virtual void twist()
            { a *= 3 ; }
        int getVal()
            { twist() ; return a ; }
} ;

class Second : public First {
    private:
        int b ;
    public:
        Second(int y = 4)
            { b = y ; }
        virtual void twist()
            { b *= 6 ; }
} ;

int main() {
    First object1 ;
    Second object2 ;

    cout << object1.getVal() << endl ;
    cout << object2.getVal() << endl ;

    return 0 ;
}
```

Program 4



```
#include <iostream>
using namespace std ;

class First {
    protected:
        int a ;
    public:
        First(int x = 3)
            { a = x ; }
        virtual void twist()
            { a *= 4 ; }
        int getVal() {
            twist() ;
            cout << "Inside getVal(), a: " << a << endl ;
            return a ;
        }
} ;

class Second : public First {
    private:
        int b ;
    public:
        Second(int y = 8)
            { b = y ; }
        virtual void twist() {
            cout << "Inside twist(Second - before, b: " << b << endl ;
            b *= 10 ;
            cout << "Inside twist(Second - after, b: " << b << endl ;
        }
} ;

int main() {
    First object1 ;
    Second object2 ;

    cout << object1.getVal() << endl ;
    cout << object2.getVal() << endl ;

    return 0 ;
}
```

Program 5 (extra credit - 5 pts)



```
#include <iostream>
using namespace std ;

class Base {
    protected:
        int baseVar ;
    public:
        Base(int val = 6)
            { baseVar = val ; }
        int getVar()
            { return baseVar ; }
} ;

class Derived : public Base {
    private:
        int derivedVar ;
    public:
        Derived(int val = 11)
            { derivedVar = val ; }
        int getVar()
            { return derivedVar ; }
} ;

int main() {

    Base *optr ;
    Derived object ;

    optr = &object ;
    cout << optr -> getVar() << endl ;

    return 0 ;
}
```

Submit a Word, text, or pdf document with your answers.

Links

Additional Files and Programs	Next Lab
none	Lab 13 - Exceptions and Templates (https://miracosta.instructure.com/courses/31330/assignments/842811)

Homework Assignment	Prior Lab
Homework 12 (https://miracosta.instructure.com/courses/31330/assignments/842798)	Lab 11 - Recursion (https://miracosta.instructure.com/courses/31330/assignments/842809)