

## CS 12 - Lab 8

### Virtual Functions

You will be using the code you wrote from last weeks lab and adding to it to demonstrate the use of virtual functions.

An Abstract Base Class is a class in which you never intend to declare an object of that type. It is solely used with inheritance to provide an appropriate base class from which classes may inherit interface or implementation. Abstract Base Classes typically do not contain data members. The following is an example of an Abstract Base Class:

```
class Shape
{
    public:
        virtual float area()
        {
            return 0.0;
        };
        virtual float volume()
        {
            return 0.0;
        };
        virtual void printShapeInfo() {};
};
```

This class has a couple virtual functions. The first two are called area and volume and simply return zero. These functions will be used in the following way. The Point class will inherit the Shape class, thus inheriting these functions. This is appropriate since the area and volume of a point are both zero. However, Circle inherits Point and provides its own area function which will override the zero returning area function. However, the volume of a Circle is still zero. Cylinder inherits from Circle and redefines its own area and volume functions.

The virtual function printShapeInfo does not do anything in the Shape function and each function will override this function to print the type of shape it is.

- Add the Shape class given above to your code. Make Point inherit the Shape class.
- Add a virtual function called printShapeInfo to the classes Point, Circle, and Cylinder that prints the type of object to the screen and the information stating the location of the object (note, do not include the printing of the area or volume). You may use your overloaded insertion operator if you'd like. To do this you would have to dereference the *this* pointer.

- Write a member function to the class Cylinder that computes and returns the surface area of a cylinder. The formula is:

$$2 * circle\_area + 2 * \pi * radius * height$$

- Check your code and verify for yourself how the virtual functions are working. You may use the following main program for testing your code.

```
void main()
{
    Point p(7, 11);
    Circle cir(22, 8);    // enter 3.5 for radius when prompted
    Cylinder cyl(10, 10); // enter 3.3 for the radius and 10 for the height

    p.printShapeInfo();
    cout << endl;
    cir.printShapeInfo();
    cout << endl;
    cyl.printShapeInfo();
    cout << endl;

    Shape *ShapeArray[3];
    ShapeArray[0] = &p;
    ShapeArray[1] = &cir;
    ShapeArray[2] = &cyl;

    for(int i=0; i<3; i++)
    {
        ShapeArray[i]->printShapeInfo();
        cout << "Area = " << ShapeArray[i]->Area() << endl;
        cout << "Volume = " << ShapeArray[i]->Volume() << endl;
        cout << endl;
    }
}
```

The output of this code should look something like:

```
Enter the radius of the circle: 3.5
Enter the radius of the circle: 3.3
Enter height: 10
Point:
[7, 11]

Circle:
Center = [22, 8]; Radius = 3.5

Cylinder:
```

Bottom Center = [10, 10]; Radius = 3.3; Height = 10

Point:

[7, 11]

Area = 0

Volume = 0

Circle:

Center = [22, 8]; Radius = 3.5

Area = 38.465

Volume = 0

Cylinder:

Bottom Center = [10, 10]; Radius = 3.3; Height = 10

Area = 275.629

Volume = 341.946

You must use Unix, separate your program into separate files, and use a makefile.