C++ Programming
Lab 4

● You may optionally work on lab 4 with one other person. If you are working alone, follow normal directions for uploading your files on your own. If you are working in a group of 2, please write BOTH names on the report and in the comments of the code. Only one student needs to upload code.

● Define a class hierarchy for Shapes. Define a base class Shape with methods to draw(), which draws this shape, and area(), which computes and returns the area of this shape:

- ○ (2 points) Define an abstract base class, called *Shape*, with two pure virtual function (double *area()* and void *draw()*). Be sure this class has a constructor which takes parameters to initialize the three data members in this order (int *centerX*, int *centerY*, string *name*). DO NOT GIVE DEFAULT PARAMETERS TO THE CONSTRUCTOR PARAMETERS! Be sure area returns a double as integer would not be precise enough for practical use. **(You may choose to use either int or double for centerX and centerY. Also, feel free to enter a center value in your main program.)**

- ○ (4 points) Derive a total of four classes from Shape: derive three classes (*Circle*, *Square*, and *Triangle*) directly from your abstract base class and make them concrete by providing implementations for the virtual functions introduced in your abstract base class *Shape*. The methods for each of these three classes must actually behave differently from one another (example: Circle area vs. Square area vs. Triangle area).

  Derive the fourth class, *Rectangle*, from *Square* and add a width. For *draw()* use character graphics. The shapes should resemble the shape defined by the class. **(Feel free to use whatever is the easiest route in drawing the shapes. You can draw them as solid or just an outline of the shape. It will be your choice as this is not a graphics course.)**

- ○ (2 points) Write a class *Picture* that holds a list of Shapes. Write a method, called add(Shape *sp) that adds the shape pointed to by sp to this picture. Also define two polymorphic methods that operate on a Picture: void *drawAll()* and double *totalArea()*. Implement Picture as a LinkedList of Shapes. Be sure your destructor cleans up your Picture when it dies.

- ○ (2 points) Write a main program that builds a Picture and fills it with two triangles, two circles, two squares, and two rectangles (Specified below). Then have it call drawAll(), then have it print out the totalArea() of the shapes on that picture.

  FirstTriangle: height=5, base=5
  SecondTriangle: height=4, base=3

  FirstCircle: radius = 5
  SecondCircle: radius = 10

FirstSquare: side=5
SecondSqaure: side = 10

FirstRectangle: height = 4 width=8
SecondRectangle: height=8 width=4

/* various pictures of the shapes omitted because they are too hard to draw by hand */
The total area of the shapes on this picture is 600.199 square units.

Lab 4 Specifics:

The output shapes may be simple and unimpressive. That's OK. Do not use any external libraries for the drawing graphics, simple character graphics is what we want. You can still draw almost standard shapes with character graphics (e.g. a rectangle).

```
*********
*       *
*       *
*       *
*********
```

or

```
----------
|        |
|        |
|        |
----------
```

Put each class (full definition) in a separate header file, but put include guards on each .h file. You can read about include guards here. Each .h file should include the class it is derived from. main.cpp will include all the .h files in the correct order (to be determined by you). Your file organization would look something like this:

```
lab4.zip
|
| - main.cpp
|
| - Shape.h
|
| - Circle.h
|
| - Square.h
|
| - Triangle.h
|
| - Rectangle.h
|
| - Picture.h
```