University of Windsor

SCHOOL OF COMPUTER SCIENCE
03-60-212 – OOP USING JAVA
SUMMER 2014

# LAB – 3

[To get the **full marks**, complete and show your works to the Lab Instructor before the end of the Lab period. Lab works submitted in the next Lab will get a maximum of **75% marks**, if you attend this Lab, and a maximum of **50% marks**, if you do not attend this Lab.]

## Objective:

Polymorphism and Abstract Classes

## Problem A:

1. Define an ***abstract*** class ***Shape*** with the following members and constructors:
   a. A static integer ***shapeCount*** to keep record of all the different shapes to be created with a initial value of 0.
   b. Two integers ***width*** and ***height*** for width and height of the shape
   c. A two-arguments ***Constructor*** to initialize the instance variables ***width*** and ***height***
   d. A method ***incrementShapeCount*** to increment ***shapeCount*** each time an object of any subclass of shape is created
   e. Accessor for ***shapeCount*** and accessors and mutators for ***width*** and ***heigh***. Mutators should validate the values of the instance variables so that the values never become less than 0.
   f. An ***abstract*** method ***getArea()***.

2. Define a new class called ***Rectangle*** by extending class ***Shape***. The class ***Rectangle*** should have the follwing additional members and constructors:
   a. A two-arguments constructor that will call the superclass constructor with the two arguments. It will also call the ***incrementShapeCount*** method to increment the value of ***shapeCount***
   b. Override the method ***getArea()*** of the super class so that the method returns the area of the rectangle
   c. A ***toString()*** method that returns
      i. The string "Area: ", followed the area of the rectangle

3. Define a new class called ***Triangle*** by extending class ***Shape***. The class ***Triangle*** should have the follwing additional members and constructors:
   a. A two-arguments constructor that will accept the length of base and height and will call the superclass constructor with the two arguments. It will also call the ***incrementShapeCount*** method to increment the value of ***shapeCount***

   b. Override the method ***getArea()*** of the super class so that the method returns the area of the triangle
   c. A ***toString()*** method that returns
      i. The string “Area: “, followed the area of the triangle

4. Define a new class called ***Square*** by extending class ***Rectangle***. The class ***Square*** should have the follwing additional members and constructors:
   a. A one-arguments constructor that will accept the length of a side of the square and that will call its superclass constructor with the two arguments.
   b. A ***toString()*** method that returns
      i. The string “Area: “, followed the area of the square

5. Write an application ***TestShape*** that will
   a. First declare a 3 element array of ***Shape*** objects.
      i. The first element should be a ***Rectangle*** object of width = 8 and height = 6.
      ii. The second element a ***Trianlge*** object with base = 8 and height = 6.
      iii. The third element a ***Square*** object with the length of a side = 6. [Of course, you can try with any other valid values for all the shapes.]
   b. It will then print the number shapes created by calling the ***getShapeCount*** method.
   c. Then, using a ***for*** loop (or a ***for-each*** loop), the program should print the string “Object of “ followed by the class name of each shape by calling method ***getClass***, followed the areas of the object.

The outputs from your application programs should match the sample outputs provided below.

**The expected output from the program TestShape.java is as follows:**

```
Total shapes created: 3
Object of class Rectangle, Area: 48.0
Object of class Triangle, Area: 24.0
Object of class Square, Area: 36.0
```