

# **CSC 1302 Principles of Computer Science II**

## **Lab 8: Class Construction Practice**

(Due on 11:59 pm, 7/2/2021)

### **Purpose:**

A class is a blueprint or prototype from which objects/instances are created. It includes fields or methods that are common to all objects of the same blueprint/class. In general, a class declaration has components: class header, constructors, methods such as getter/setter, and so on. The constructors are used for initializing new instances. Fields are variables representing the state of the class and its objects. The methods are used to implement the behavior of the class and its objects.

In this lab, we will practice how to define classes and create objects. Once a class is available, it's like we have a new data type. Variables can be defined with a type of this class and can refer to an object of this class. Methods can use this class as parameter type or return type.

### **Criteria:**

1. Upload all of the .java and the .class files to the CSc1302 dropbox on <http://icollege.gsu.edu>.
2. Your assignment will be graded based on the following criteria: (a) Are your programs runnable without errors? (b) Do your programs complete the tasks with specified outputs? (c) Do you follow the specified rules to define your methods and programs? (d) Do you provide necessary comments include the programmer information, date, title of the program and brief description of the program.
3. Please comment the important lines in the .java file as shown in the template. The important lines including but not limited to i) variables, ii) for-loop, iii) while-loop, iv) if-else statement, iv) methods. Please use your own words to describe what is your purpose to write this line. A .java file without comment will be graded under a 40% penalty.
4. Make sure that both the .java and .class files are named and uploaded to icollege correctly. If any special package is used in the program, be sure to upload the package too. Should you use any other subdirectory (whatsoever) your program would not be graded, and you will receive a 0 (zero).
5. No copying allowed. If it is found that students copy from each other, all of these programs will get 0.

## **Task:**

### **1. Define a class named Circle.**

2. A `Circle` object stores a **radius** and the **(x, y)** coordinates of its center point using `Point` class (`Point` class should be similar to the one we wrote during lectures. It should have at least two **private** fields to store x and y coordinates of a point, one constructor, a `toString` method and a distance method).
3. Each `Circle` object should have 2 private fields, a `Point` object and radius. Each `Circle` object should have the following public methods:
  - `Circle(p, radius)`  
Constructs a new circle with a center specified by the given `Point p` and with the given integer radius.
  - `getCenter()`  
Returns point object for the center of the circle.
  - `getRadius()`  
Returns the circle's radius.
  - `getArea()`  
Returns the area occupied by the circle, using the formula  $\pi r^2$ .
  - `getCircumference()`  
Returns the circle's circumference (distance around the circle), using the formula  $2\pi r$ .
  - `toString()`  
Returns a string representation of the circle, such as `"Circle[center=(75, 20), radius=30]"`.
  - `contains(p)`  
Returns true if the point `p` lies within the circle else returns false. (Hint: calculate the distance between the center and the point `p` and compare it with the radius)

Write a client class named `CircleClient`. It should create an object of class `Circle` and initialize it to point (10,5) and radius 7. Print the object. Print out its circumference and area. If point (5,7) lies within the circle print “(5,7) lies within the circle” else print “(5,7) does not lie within the circle”.