**Lab 3 18-09-2017**

**Object Oriented Concepts and Programming**

**Learning Objectives**

**Teacher perspective:**

1. Base the lab to maximum extent on students’ earlier learning in FCP course by avoiding new concepts for now such as constructors (and its types), access specifiers etc.
2. Introduce the concept of setter/getter functions
   1. Emphasize and demonstrate the importance of input validation in setter functions
3. Introduce the concept of source-class and driver/test-class

**Students’ perspective:**

The students will be able to

1. Implement classes in java along with their methods (behaviors) and attributes (properties)
2. Create object of classes for using them
3. Develop working programs in java by creating and using objects

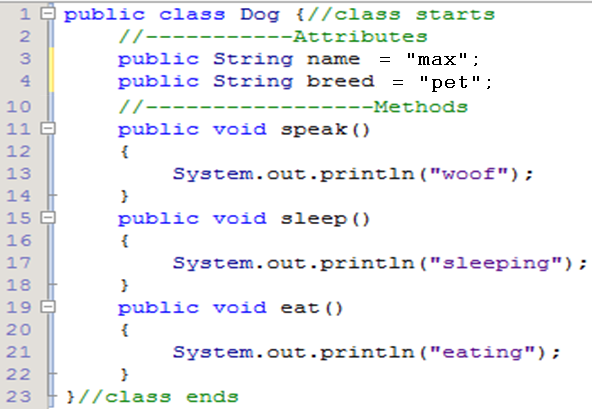
|  |
| --- |
| **Lab Walkthrough** |

**Program 3.1**

A sample Dog class.

Attributes are name and breed.

Methods are speak, sleep and eat.

****

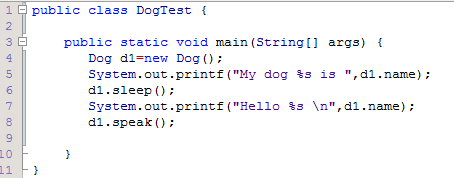
**Experiment 3.1:**

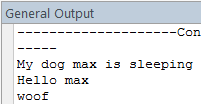
1. compile the above class (without main) using java filename.java. Does it compile? Why?

2. if above experiment is successful, run the compiled .class file. What happens? Why?

**Program 3.2**

A sample DogTest class (in separate file)

****



|  |
| --- |
| **Lab Demo** |

**Lab Demo 3.1.**

The instructor will himself demonstrate the lab demo on multimedia.

Write a class Circle which will model the functionality of a Circle.

1. Attributes
   * radius
2. Methods
   * Setter function for radius
   * Getter function for radius
   * To compute area
   * To compute perimeter

|  |
| --- |
| **Lab Task** |

**Lab Task 3.1**

* Design a class Rectangle which has following members:
  + Data:
    - length
    - height
  + Methods:
    - getArea
    - getPerimeter
* Test Rectangle inside the main function

|  |
| --- |
| **Lab Task** |

**Home Task 3.1**

* Design a class QuadraticRoots which has following members:
  + Data:
    - a
    - b
    - c
    - root1
    - root2
  + Methods:
    - Setter functions (total 3)
    - Getter functions (total 3)
    - computeRoots
      * set root1 and root2 in this function (ignore iorta case)
    - displayRoots
      * display in format: (root1, root2)
* Test QuadraticRoots inside the main function.
  + Create an object of QuadraticRoots class.
  + Input instance variables a, b, c from user.
  + Call the computeRoots and displayRoots functions.