Presentation 2 Topics

All classes in this presentation should contain private data.

1. Create a Circle class to define circle objects. A circle has a center and a radius. Use the Point class to define the center of the circle. Your class must contain two constructors – one without arguments and the other with two arguments. The constructor without arguments will create a default circle with radius 5 and center (0,0). The constructor with arguments will create a circle with custom radius and center as specified by the arguments. Your class must also include a toString() method to give a string representation for your circle.
Create a main class to test your circle class. Create a default circle and another circle with radius 10 and center (5,0). Print both circles.

2. Work with group 1 to enhance the circle class. You will add getArea() and getPerimeter() methods to compute the area and perimeter of the circle respectively. Test both methods in the main class for different circles and display the results.

3. Create a 3D point class. A 3-dimensional point has a z-coordinate in addition to the x, y coordinate. Add a constructor with no arguments and a constructor that takes three inputs. Also, add a toString() method to give a String representation for your point. Create a main class to test your class. Create at least two 3D points and display them.

4. Create a Sphere class to describe Sphere objects. A sphere is a 3D solid that has a center and a radius. Work with group 3 and use their 3D Point class to represent the center of the sphere. Your class must contain two constructors – one without arguments and the other with two arguments. The two arguments will serve as inputs for the center and the radius of the sphere. Your class must also include a toString() method to give a String representation for your Sphere. Create a main class to test your Sphere class. Create a default sphere and another sphere with center (-2,-10) and radius 50. Print both spheres.

5. Work with group doing topic 4 to enhance the Sphere class. You will add getSurfaceArea() and getVolume() methods to compute the surface area and the volume of the sphere respectively. Test both methods in the main class for both the spheres described in topic 4 and display the results.

6. Create an Employee class to describe employee objects. An Employee class contains information about an Employee such as the ID number

and the name. Add a constructor with no arguments and a constructor with two arguments. The two arguments will correspond to the ID and the name of the employee. Also, add a toString() method to give a String representation for your Employee. Create a main class to test your class. Create two employees by asking the user for their ID numbers and names and display both employees.

7. Create a Department class to describe departments. You will use the Employee class created by group 6 to describe employees of your department. A department must have a name and two employees. Create a constructor that takes as input the name of the department, and two employee objects as input. The name and the employee objects will be initialized in the constructor. Your Department class must include a toString() method to give a String representation for the department. Create a main class to test your class. Create at least two different departments and display them.

8. Work with group doing topic 7 and enhance their Department class to include a new method, verifyEmployee(), that takes ID number as input and checks whether the corresponding employee belongs to the department. So you must check whether the id matches either of the two employees and return true if there is a match otherwise return false.

Create a main class to test your method. First, create a department with two different employees and display the department. Then use verifyEmployee() to check if an employee exists.