**Online Lab 3**

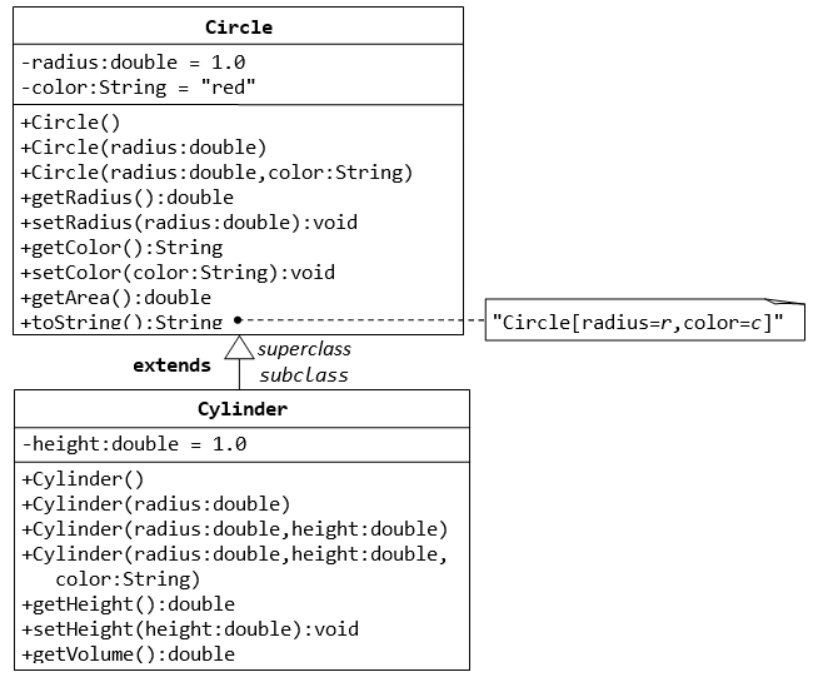
**Object Oriented Programming**

**Learning Objectives**

1. The students will be able to

* Implement the inheritance relationship between two classes having an Is-A relationship
* Override and reuse base class behavior in the derived class
* Override the common object class methods such as toString.
* **What is Method Overriding? Method Overriding vs Method Overloading**
* **Uses of Super Keyword!**

|  |
| --- |
| **Lab Walkthrough/Demo** |



* **Add an overridden method to getArea() in Cylinder class to calculate the surface area of the Cylinder object.**
* **Using ‘super’ in child class to call the parent class constructor.**
* **Using ‘super’ in the child class to call the parent class version of the overridden method.**

**THE Test CLASS**

public class **TestCylinder** { // save as "TestCylinder.java"

public static void main (String[] args) {

// Declare and allocate a new instance of cylinder

// with default color, radius, and height

Cylinder c1 = new Cylinder();

// Declare and allocate a new instance of cylinder

// specifying height, with default color and radius

Cylinder c2 = new Cylinder(10.0);

// Declare and allocate a new instance of cylinder

// specifying radius and height, with default color

Cylinder c3 = new Cylinder(2.0, 10.0);

System.out.println("Cylinder1 data:" + c1

+ " Cylinder2 data=" + c2

+ " Cylinder3 data=" + c3);

}

}

|  |
| --- |
| **Home Tasks** |

Now complete the following tasks inside the **Employee** class:

1. Consider a base class named **Employee** and its derived classes **HourlyEmployee** and **PermanentEmployee** while taking into account the following criteria.
   1. Employee class has three data fields i.e. a name (of type string) and specific empID (of type integer), and hourlyincome.
   2. Two Constructors i) No argument Constructor ii) Three argument Constructor
   3. A Method earnings() that returns the income of employee.
   4. It has a toString method, which returns a string in the following format.
      1. Employee: Ali (ID: 786), Income: 90000
   5. The derived classes (HourlyEmployee) has an attribute named totalHours.
   6. Both the classes (HourlyEmployee and PermanentEmployee) have a no argument constructor and a three-argument constructors to initialize the hourlyIncome as well as data fields of the base class.
      1. The id, name, and hourlyincome should be initialized using the constructor of the employee class (Show the appropriate use of super keyword)
      2. The totalHours for HourlyEmployee should be input from user right inside the constructor
   7. Override Method earning() in HourlyEmployee and permanenetEmployee to calculate the income of an employee for the actual number of hours he or she worked. Note, the income for Employee/PermanentEmployee objects are paid for exact 180 hours, no matter how many actual hours he or she worked.
   8. Implement the toString Method for both classes. The method should return the string formatted according to following for each class:

The Test Class:

1. Develop the main() method inside the employee class, create an instance of Employee class in addition to both classes (i.e. HourlyEmployee and PermanentEmployee) with the data of your own choice.
2. Test the working of functions that calculate the total income of an employee.
3. Test the toString methods of both classes by printing both objects in a print statement.