

# **Java Object-Oriented Concepts Assignment**

**Due Date: 15-11-2022**

## **Important Instructions:**

- Consider relevant data types for each variable.
- Make getters and setters for each class.
- display() method should print all the available details.
- Each class should be defined in a separate file.
- Create two objects of each class in the Main class.
- After creating objects, call all the available methods.
- Implementation of methods is up to you.

1- Create a Shape class having one data member- name. Add the getters and setters methods, a display() method. Create two objects of Shape class in the main method.

(Implementation of class, getters and setters and object creation)

2- In Shape class, create as many constructors as you want. Use the concept of constructor chaining. Create two objects of Shape class in the main method.

(Implementation of Constructor and Constructor chaining)

3- Create a class Circle that extends Shape class. Circle class has radius as data member, getters and setters and a display method(). Create constructors and call the constructor of Shape class appropriately. Create two objects of Circle class in the main method.

(Implementation of Single Inheritance, Use of Super keyword)

4- Create a class Rectangle that extends Shape class. Rectangle class has length and width as data members, getters and setters and a display() method. Create constructors and call the constructor of Shape class appropriately. Create two objects of Rectangle class in the main method.

(Implementation of Hierarchical Inheritance)

5- In Shape class, add two new methods calArea() and calParameter() which display the message "Area/Perimeter will be calculated soon".

(preparation for Overriding)

6- In Circle and Rectangle class give specific implementations of calArea() and calParameter() methods which actually calculate Area/Perimeter of respective classes. Create two objects of Circle and Rectangle classes in the main method.

(Overriding)

7- In Shape class, create a new static method Info() which prints "Area refers to space covered by 2D object and perimeter refers to the length of its boundary". Call this method from Main class.

(Use of static keyword)

8- In the Main class, create a method that prints "It is a Shape/Circle/Rectangle" depending upon the type of object passed as an argument.

(Use of Upcasting, instanceof keyword)

---

9- Create an abstract class Shape with data member- name, getters and setters, constructors and an abstract method calArea().

(Use of Abstract class and method)

10- Create two child classes Shape\_2D and Shape\_3D which implement calArea() method. Create two objects of Shape\_2D and Shape\_3D classes in the main method.

(Use of Abstract class and method)

11- Make calArea() method abstract in Shape\_2D. Create one more abstract method calParameter(). Create a class Circle which extends Shape\_2D and has radius, getters and setters, display() method and implements calArea() and calParameter() methods. Create two objects of Circle class in the main method.

(Use of Abstract class and method)

12- Make calArea() method abstract in Shape\_3D. and create one more abstract method calVolume(). Create a class Sphere which extends Shape\_3D and has radius, getters and setters, display method and implements calArea() and calVolume() methods. Create two objects of Sphere class in the main method.

(Use of Abstract class and method)

---

13- Create a class Human\_body with data members height and weight, getters and setters and constructors. Create two objects of Human\_body class in the main method.

(Use of Nested class, composition relationship)

14- Create a nested class Eyes with data members color and size, getters and setters and constructors. Create two objects of Eyes class in the main method.

(Use of Nested class, composition relationship)

---

15- Create a class Resolution with data members pixelRow and PixelColumn, getters and setters, display() method and constructors. Create two objects of Resolution class in the main method.

(Aggregation relationship)

16- Create a class Monitor with data members color, dimension and resolution, getters and setters, display() method and constructors. Create two objects of Monitor class in the main method.

(Aggregation relationship)

17- Create a class InputDevice with data members name and price, getters and setters, display() method and constructors. Create two objects of InputDevice class in the main method.

(Aggregation relationship)

18- Create a class Computer with data members brand, price, Monitor and InputDevice, getters and setters, display() method and constructors. Create two objects of Computer class in the main method.

(Aggregation relationship)

---

19- Create Interface RBI which has a data member- minInterestRate and method calculateInterest(). Create a class Bank which implement RBI interface. Create two objects of Bank class in the main method.

(Interface implementation)

20- Modify Bank class which implements Interface RBI but does not implement setInterest() method. Bank class also has its own abstract method showAnnualRevenue(). Create three classes SBI, ICICI and HDFC classes which extend Bank class. Create two objects of SBI, ICICI and HDFC classes in the main method.

(Interface and Abstract class)