Java assignment on Interfaces

1. Interface **Material** defines a set of string constants for various materials. Abstract class **MatObject** has one instance variable named **material** of type String. This records the material used to construct the object. Classes **Ball, Coin** and **Ring** extend **MatObject.** The constructors initialize the **material** variable. Class **MatObject** instantiates these classes. A different material is passed to each constructor. The material of each object is displayed.

2. Interfaces **AntiLockBrakes, CruiseControl** and **PowerSteering** declare optional functionality for an automobile. Each interface declares one method that has the same name as its interface. The abstract **Auto** class is extended by **Model1, Model2** and **Model3** classes. **Power steering** is available for **Model1** objects. **AntilockBrakes** and **CruiseControl** are available for **Model2** objects. **CruiseControl** is available for Model3 objects. Instantiate each of these classes and exercise its methods.

3. Interface **LuminousObject** declares **lightOff()** and **lightOn()** methods. Class **SolidObject** is extended by **Cone** and implements **LuminousObject**. Class **LuminousCone** extends **Cone** and implements **LuminousObject**. Class **LuminousCube** extends **Cube** and implements **LuminousObject**. Instantiate **LuminousCone** and **LuminousCube** classes. Use interface reference to refer to those objects. Invoke the methods of the **LuminousObject** interface via the interface reference.

4. Interface **P** is extended by **P1** and **P2.** Interface **P12** inherits both **P1** and **P2**. Each interface declares one constant and one method. Class **Q** implements **P12**. Instantiate **Q** and invoke each of its methods. Each method displays one of the constants.

5. Interface **K1** declares **methodK()** and a variable **intK** that is initialized to one. Interface **K2** extends **K1** and declares **methobK()**. Interface **K3** extends **K2** and declares **methodK().** The return type of **methodK()** is void in all interfaces. Class **U** implements **K3**. Its version of **methodK()** displays the value of **intK**. Instantiate **U** and invoke its method.

6. Declare interface **L1** and **L2**. Interface **L3** extends both of these interfaces. Also declare interface **L4**. Class **X** implements **L3**. Class **W** extends **X** and implements **L4.** Create an object of class **W**. Use the instanceof operator to test if that object implements each of the interfaces and is of type **X**.

7. Write the following applications: an interface **Shape2D** that declares a **getArea( )** method that calculates and returns the area of an enclosed 2D shape. Interface **Shapc3D** declares a **getVolume( )** method that calculates and returns the volume of an enclosed 3D shape. **Point3D** contains the coordinates or a point. The abstract class Shape declares all abstract **display( )** method and is extended by the circle and Sphere classes.

The former implements the **Shape2D** interface and the later implements the **Shape3D** interface. The shapes classes instantiates each of these classes and exercises their methods.