Tutorial 2

Classes & Objects

1. Write a class `Circle` that has the following instance variables and methods:

```
public class Circle
{
   private double radius;         // radius of circle
   private static final double PI = 3.14159;

   // constructor
   public Circle(double rad) {...}
   // mutator method - set radius
   public void setRadius(double rad){...}
   // accessor method - get radius
   public double getRadius(){...}
   // calculate area
   public double area(){...}
   // calculate circumference
   public double circumference() {...}
   // print area
   public void printArea(){...}
   // print circumference
   public void printCircumference(){...}
}
```

The UML class diagram for the Circle class is given below:

Circle
- radius: double
+ Circle(rad: double) + setRadius (rad: double): void + getRadius(): double + area(): double + circumference(): double + printArea(): void + printCircumference(): void

Write an application class `CircleApp` to test the `Circle` class. The class `CircleApp` should display a menu. The user can then select an option of the following: (1) create a new circle; (2) print area; (3) print circumference; and (4) quit. Implement the operations for each option.

A sample program run is given below:

```
 ----jGRASP exec: java CircleApp

==== Circle Computation =====
|1. Create a new circle     |
|2. Print Area              |
|3. Print circumference     |
|4. Quit                    |
============================
Choose option (1-3):
1
Enter the radius to compute the area and circumference
```

```
5
A new circle is created
Choose option (1-3):
2
Area of circle
Radius: 5.0
Area: 78.53975
Choose option (1-3):
3
Circumference of circle
Radius: 5.0
Circumference: 31.4159
Choose option (1-3):
4
Thank you!!
```

2. Write a class `Dice` that has the following instance variables and methods:

   The UML class diagram for the Dice class is given below:

Dice
- valueOfDice: int
+ Dice() + setDiceValue(): void + getDiceValue(): int + printDiveValue (): void

   Write an application class `DiceApp` to test the class `Dice`. The class `DiceApp` interacts with the user to generate the numbers randomly from rolling a pair of dices. The generated numbers from the pair of dices and the total are then displayed on the screen.

   A sample program run is given below:

```
 ----jGRASP exec: java DiceApp

Press <key> to roll the first dice
1
Current Value is 3
Press <key> to roll second dice
2
Current Value is 3
Your total number is: 6
```