Classes and Objects

CLASSES

A class is a user defined program that creates objects. It contains the set of properties attributed to a certain object.

The structure of a class

- **Constructor(s)**
  - Methods that initialize objects
  - A default constructor does not have parameters and sets the object to a predetermined default identity
- **Instance variable(s)/field(s)**
  - Usually private variables (int, double, String, boolean, float, etc)
    - Private variables can only be accessed within the program they are declared in
  - These are the parameters of the object
- **Mutators/Setters**
  - Methods that allow the user to change a parameter of the object
  - Void methods, every instance variable has one
- **Accessors/Getters**
  - Methods that allow the user to access a parameter of the object
  - Have the same return type as the parameter, every instance variable has one
- **ToString**
  - Provides the String representation of an object, usually displays the parameters of the object
- **Other methods pertaining to the object**

Class tester

To test your class, you must write a new Java program that creates the object of the class and implements the methods of the class to see if they work how you want them to.

Example class: Telescope

```
public class Telescope   {

//A class that models a field telescope.

//private instance variables

   private double diameter;

   private double mainLength;

   private double eyeLength;
```

```
//Constructors

   public Telescope (){
    Diameter = 1;
    mainLength = 1;
    eyeLength = 1;
   }

   public Telescope (double inDiameter,double inMainLength,double inEyeLength)
   {
      diameter   = inDiameter;
      mainLength = inMainLength;
      eyeLength  = inEyeLength;
   }
```

```
// Accessors (These methods retrieve information without changing it)

   public double getDiameter ()
   {
      return diameter;
   }

// Add two more accessors to get information about mainLength and
eyeLength

   public double getMainLength ()
   {
      return mainLength;
   }

   public double getEyeLength ()
   {
      return eyeLength;
   }
```

```
// Mutators (These methods change information related to an object)

   public void setDiameter (double diam)
   {
      diameter = diam;
   }

// Add two more mutators to change the values mainLength and eyeLength
// Methods: Note these methods are not static because they are associated
with a class

   public void setMainLength (double mainLeng)
   {
      mainLength = mainLeng;
   }

   public void setEyeLength (double eyeLeng)
   {
      eyeLength = eyeLeng;
   }
```

```
// return a String with diameter, mainLength, and eyeLength

public String toString()     {
      return "Diameter: " + diameter + "\nMain Length: " + mainLength +
      "\nEye Length: " + eyeLength;
   }

}
```

```
// calculate the magnification or power of the telescope
//The formula to use is: magnification = mainLength/eyeLength

   public double  calcMagnification()
   {
      return mainLength/eyeLength;
   }

//The formula to use is: fNumber = mainLength/diameter

   public double  calcFNumber(){
   // calculate the f-number of the telescope
      return mainLength/diameter;
   }

}
```

Telescope tester

```
public class TelescopeTester
{
   public static void main ( String[] args )
   {
      Telescope tele = new Telescope(3.0,6.5,0.8);

   //Use the accessor methods to print out the telescope's properties.

      System.out.println( "Diameter: "+ tele.getDiameter());

      System.out.println( "Length of the main lens: "+ tele.getMainLength());

      System.out.println( "Eyepiece: "+ tele.getEyeLength());

      System.out.println(tele);
```

```
System.out.println("Power: "+ tele.calcMagnification() + "  F-number: " + tele.calcFNumber() );

   // Use mutator method to change the main length to 7.5 inches

      tele.setMainLength(7.5);

      System.out.println(tele);

   // and check the results

      System.out.println("New Power: "+ tele.calcMagnification());

   }

}
```

Exercises for classes and objects

https://www.w3schools.com/java/exercise.asp?filename=exercise_classes1

Write and test a class called rectangle that:

-Has 2 fields: double length and double width

- Has a constructor

- Has Mutators and Accessors for each Field

-Has a toString method

- Has 2 methods:

    - one that calculates the area of the rectangle

    -One that calculates the perimeter of the rectangle

use all methods in the tester

Write and test a class called speedometer that:
-has 3 fields:
-int maxSpeed, int currentSpeed, and String units
-has a constructor
-Has mutators and accessors for each field
-Has a toString method that prints the current speed with units
-Has a method that calculates and returns what percent of the max speed the user is travelling at

*test all methods in the class tester *