Static Variables

Do now

What if you want to keep track of something across an entire class?

What if you want to keep track of something across an entire class?

If a variable or method belongs to the **class** and not the **instance**, it is considered a class variable or class method.

These can also be called **static variables** or **static methods.**

Class Variables or Static Variables

A variable or attribute of a class that is common to all instances of a class

Class Methods or Static Methods

A method of a class that is common to all instances of a class, and is not called on an object instance.

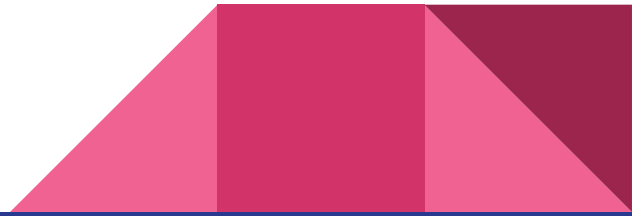
Static and Non-static methods

A **static method** is a method that belongs to a class, but it does not belong to an instance of that class and this method can be called without the instance or object of that class. They may not use non-static methods. Example: static methods in class “Math.abs()”, “Math.pow()”, “Math.PI”

```
public static void printMsg(){
  System.out.println("Hi! Rectangle class");
}
```

Non-static belongs to each object that is generated from the class. Methods can access any **static** method and **static** variable.

```
public int calcArea(){
  return width * height;
}
```



Static variable example

```
public class Rectangle{

    // This variable is the same for all instances.
    // It tracks how many total rectangle objects have been created.

    private static int totalRectangles = 0;

    public Rectangle(int myWidth, int myHeight){

        width = myWidth;

        height = myHeight;

        // every time we make a new rectangle, it will increment by 1

        totalRectangles++;

    }

}
```

Static Method Example

```
class Rectangle{
        // other code here ……
        public static int getTotalRectangles(){
                return totalRectangles;
        }
}
```

What is printed?

```
public class SomeClass{
     private int x = 0;
     private static int y = 0;
     public SomeClass(int pX){
          x = pX;
          y++;
     }
     public void incrementY(){
          y++; }
     public void incrementY(int inc){
          y += inc; }
     public int getY(){
          return y; }
}
```

The following code segment appears in a class other than SomeClass.

```
SomeClass first = new SomeClass(10);
SomeClass second = new SomeClass(20);
SomeClass third = new SomeClass(30);
first.incrementY();
second.incrementY(10);
System.out.println(third.getY());
```

A. 0 B. 1 C. 11 D. 14 E. 30

Exercise

Create a class Employee (Employee.java)

Define instance variables

Define at least 3 construtor (1 of them default constructor)

Declare static and non-static methods

Create multiple objects from your Driver.java and test the constructors and methods.

Save here:
.../APCSA1/apcsa-assignments-fall-YourUsername/classwork/10_14_classes/