STATIC METHODS AND FIELDS

Fundamentals

- A **static variable** or **class variable** is any field declared with the `static` modifier; this tells the compiler that there is exactly one (1) copy of this variable in existence, regardless of how many times the class has been instantiated.
- A **static method** belongs to the class rather than the object of a class.
    - It can be invoked without creating an instance of a class.
    - It can access static members and change their values.
- To call a static method or variable, put the class name before the method or variable.
  Example:
```
class NumberClass {
    public static int num = 0;
    public static void displayNumber() {
        System.out.print(num);
    }
}
public class NumberClassDemo {
    public static void main(String[] args) {
        System.out.print(NumberClass.num);
        NumberClass.displayNumber();
    }
}
```
  Output: 00
- An instance variable can be used to call a static method or update a static variable.
  Example:
```
public class NumberClassDemo {
    public static int num = 0;
    public static void main(String[] args) {
        NumberClass nc1 = new NumberClass();
        NumberClass nc2 = new NumberClass();
        nc1.num = 6;
        nc2.num = 5;
        System.out.print(NumberClass.num);
    }
}
```

Output: 5
Explanation: There is only one (1) num variable since it is static.

Static versus Instance

- An **instance variable or method** requires an object of its class to be created before it can be used or called.

Type	Accessing	Legal?	How?
Static method	Another static method or variable	Yes	Using the class name
Static method	An instance method or variable	No	
Instance method	A static method or variable	Yes	Using the class name or reference variable
Instance method	Another instance method or variable	Yes	Using a reference variable

Example of a static method calling an instance method:
```
public class NumberClass {
    public static int num = 0;
    public static void main(String[] args) {
        displayNumber(); //does not compile
    }
    public void displayNumber() {
        System.out.print(num);
    }
}
```

Example of a static method trying to access an instance variable:
```
public class NumberClass {
    int num = 0;
    public static void main(String[] args) {
        displayNumber();
    }
    public static void displayNumber() {
        System.out.print(num); //does not compile
    }
}
```

Reference:
Oracle Docs (n.d.). *Citing sources.* Retrieved from https://docs.oracle.com/javase/tutorial/java/javaOO/index.html