Questions? ☺️

Object-Oriented Programming

Practice Exercise 5

Object-Oriented
Programming

Static and Final

Static Keyword

- Is a modifier that can be applied to variables, methods, and classes
- Associates the assigned entity to the class, not an instance
    - Static variable or method can be used by directly referencing the class
        - object.method() -> Class.method()

Example of Using a Static Variable

```
public static void main(String args[]){
    SomeClass temp1 = new SomeClass();
    SomeClass temp2 = new SomeClass();
    SomeClass temp3 = new SomeClass();

    System.out.println(temp1.id);
    temp2.id++;
    System.out.println(temp1.id);
    temp3.id++;
    System.out.println(temp1.id);
    System.out.println(temp2.id);
}
```

```
public class SomeClass {
    public static int id = 0;
}
```

Output

```
0


1


2
2
```

Example of Using a Static Method

```
public class SomeClass {
    public static int id = 0;

    public static int method1() {
        return 1;
    }
}
```

```
public static void main(String args[]){
    System.out.println(SomeClass.method1());

    SomeClass temp1 = new SomeClass();
    System.out.println(temp1.method1());
}
```

Output

```
1


1
```

Reflect

Are **static** methods/variables aligned with OOP?

Static violates OOP

- Removes the need for objects – for instances
- Objects with static variables do not have full control over their variable since they can be accessed anywhere
- Uses of static variables and methods is better for procedural programming, not OOP
    - BUT can be useful for certain design patterns

Use of Static in Singleton Pattern

```
public class MyDbHelper {
    private static MyDbHelper instance = new MyDbHelper();

    private MyDbHelper() {
        // private: no one can instantiate the class except for itself
    }

    public static MyDbHelper getInstance() {
        return instance; // single instance is maintained here
    }

    // Other database methods
}
```

This is a useful pattern / programming technique despite the concept not aligning with OOP

Questions? 😊

Final

- Is a modifier that can be applied to variables, methods, and classes
- Final…
  - Variables must have a value
    - Can be declared blank final but must be initialized at some point
  - Methods cannot be overridden
  - Classes cannot be extended

We will discuss more about overriding and extending when we reach Inheritance

Example of Using a Final Variable

```
public class Driver {
   final int x = 1;

   public static void main(String args[]) {
      x = 4; // this will throw an error
   }
}
```

Example of Using a Blank Final Variable

```
public class Student {
    private final int id; // blank final variable
    private String name;

    public Student(int id, String name) {
        this.id = id; // eventual assignment of the final variable
        this.name = name;
    }
}
```

Final + Static

- You could use a combination of the two to create constant global values…
- But you'll have to reflect on whether a class is doing much more than is expected of it
- Alternative for variables would be the Enum class

cont…