Griffin's OH 4/16/22

COMS 1004 Introduction to Computer Science and Programming in Java

Quick Announcements

Announcements for the week of April 15th 2022

- You have made it to the last month of classes, this class ends April 28th
- Quiz 5 makeup will be on April 19th so study now if you deem it necessary
- Programming Problem Set 5 is due on Monday please start on it if you haven't already done so
- Please Start Submitting Questions for the Review Video [see this link]

Topics for the Week

1. Statics in Java
2. Inheritance

1

Statics in Java

Statics in Java

The static keyword in java is used primarily for memory management. The static keyword can be used with classes, variables, methods and blocks (out of scope). The static keyword belongs to the class than instance of the class. This means if you make a member static, you can access it without an object.

If we make variables static then regardless of whatever instance of an object we have, the variable is shared amongst all of them. Meaning if one instance of a class alters a static variable, all the other instances of that class will have the same value for that variable.

Why use Statics in Java

Nowadays the real use for statics is to make sure if you want a constant value between instances of classes regardless of the specific operations performed in each separate instance.

The reality of it though is that statics are a remnant of procedural programming prior to the wide scale adoption of Object Oriented Programming.

If you want to save yourself time with testing, I would suggest making the method you wanna test static until you can confirm functionality. For PP5 specifically though, all methods should be declared as static and remain as such

2

Inheritance

Inheritance

In object-oriented design, inheritance is a relationship between a more general class (called the super class) and a more specialized class (called the subclass). The subclass inherits data and behavior from the superclass

When you use inheritance in your programs, you can reuse code instead of duplicating it. This reuse comes in two forms. First, a subclass inherits the methods of the superclass. For example, if the Vehicle class has a drive method, then a subclass Car automatically inherits the method. It need not be duplicated.

Inheritance

When you want to declare a subclass the notation is quite simple; simply add the extends clause and the name of the superclass to the class definition of the subclass

```
public class subClass extends superClass{}
```

While it is true that by default you will inherit all public methods from the superClass, if you want to override the implementation of one of the inherited methods, you are free to do so. Simply redefine the method (with the exact same method signature) and write your new implementation.

Link to Video Drive and Other Resources

Link to Video Drive:
Video Drive

Link to My Office Hour Materials:
Useful Files