Object Oriented Programming

Mr. D Bylsma
Chris
Adrian

1 Preface

You are learning Object Oriented Programming. OOP is a paradigm - a "style" or "way" of designing programs. We will use Java, but OOP concepts can apply anywhere.

1.1 Please note

You must read, learn, live, and love these before continuing!

- whathaveyoutried.com

- xkcd.com/627

2 Setting up Netbeans

Just watch the video, or google it.

3 Java Syntax

3.1 Variables

Variable declarations always have the format Type variableName. They can be initialized like Type variableName = initialValue.

- When you divide integers, the decimal part is cut off. This is "integer division"

- When you add/multiply/divide/subtract an integer by/with a double, it's automatically converted to a double

- When adding a number to a String, everything is converted to a String

3.2 Control Flow

3.2.1 If Branch

- You must compare strings with if (str1.equals(str2))

3.2.2 For Loop

- Idiom: for (int i = 0; i < n; i++) when you want to do something i times.

- Always count from 0

- Use it when you need to do something a known number of times

3.2.3 While Loop

- Use it when you need to do something an unknown number of times (e.g. you don't know how many times the user will get the password wrong)

3.3 Functions

- For now, everything will be public static ReturnType

- If a function doesn't need to return a result (e.g. result of computation, error/success code, parsed data), it has a return type of void. Analogies:

    - My parents tell me to do my homework. I finish it, then go to sleep. The return type is void because I only need to do my own thing

    - My parents tell me to wash the dishes. I finish, and that's it. Again, the return type is void

    - The teacher asks me to answer the next question. I work on the problem, and return my answer. The return type may be a double if it is a math problem, or Sting if it is an explanation, or an EquilibriumEquation (we'll learn to define our own types later) if it's chemistry

    - I ask the new student for his name. He returns his name (a String)

- Functions with a void return type always have a side effect - like printing to the console

- Functions with a concrete return type may or may not have a side effect. computeAverage(double test1, test2) just does some math and returns the value. getUserNameFromConsole returns the user's input, but also has a side effect of interacting in the console