Think Java Chapter 1

Monday, August 26, 2019 2:42 PM

1.1 What is a Computer?

- People usually think of computers as just Desktops or Laptops.
- The term *computer* can refer to any device that stores and processes data.
- While computer designs differ, there are two components all devices have in common.
  - Processor (CPU) - Perform simple calculations
  - Memory (RAM) - Temporarily Stores Information
- While users usually interact with screens and keyboards, the CPU and RAM perform the actual Computations.
- Fairly standard for a phone to have eight processors and four gigabytes memory.

1.2 What is Programming?

- A *program* is a sequence of instructions that say how to perform a computation on computer hardware.
  - This computation might be *mathematical* like finding the roots of a polynomial, or *symbolic*, like replacing text in a document.
- Details of languages look different, but a few basic instructions exist in every language.
  - Input - Get data from a keyboard, file, etc.
  - Output - Display data on a screen or send it to a file.
  - Math - perform basic mathematical operations like addition and division.
  - Decisions - Check for certain conditions and execute the appropriate code.
  - Repetition - Perform some action repeatedly.
- Overall, *programming* is the process of breaking down a large, complex task into several smaller and smaller tasks. The process repeats until the tasks are simple enough to be performed with the electronic circuits provided by the hardware.

1.3 The "Hello World!" Program

- The first program you write in a language is typically called the *Hello World Program.*
  - It outputs the words "Hello, World!" to the screen.
- Java Programs are made of *class* and *method* definitions, and methods are made up of *statements*.
  - In the Java Hello world program, this line is a *print statement* that displays a message on the screen:
    - `System.out.println("Hello, world!");`
  - "println" stands for "print line".
  - Like most statements, this statement ends with a semicolon (;).
- Java is "Case-Sensitive", meaning capitalization matters.
- A *method* is a named sequence of statements. This program has one method named *main*.
  - `public static void main(String[] args)`
  - "main" is special: when the program runs, it starts at the first statement in main and ends at the last one.
- A *class* is a collection of methods. This program defines a class named "Hello".
  - Classes can be named anything, but typically are capitalized.
- Java uses curly braces ({ and }) to group things together. In Hello.java, the outermost braces contain the class, and the inner braces contain the method.
- "//" is a comment, used to explain the code. Any line starting with "//" is ignored upon execution of the program.

1.4 Compiling Java Programs

- Java is a "high-level" language, which must be translated into a lower-level language, called "machine language".

- Advantages of High-Level languages:
    - It is *much easier*.
    - Programs made with High-level languages are portable, meaning they work from Computer to computer with few modifications.
- Two types of tools to translate high to low level:
    - An *Interpreter* reads a high-level program and executes it a little at a time.
    - A *Compiler* reads the entire program and translates it before running it. Compilation only needs to be done once, meaning these programs run faster.
- Java is *both* Compiled and Interpreted. Instead of translating directly into machine code, the Java compiler produces "byte code". Similar to object code, byte code is fast, but it is also portable, which is a main advantage of Java.
- For example, a programmer writes the code in Hello.java and uses the command *javac* to compile it. The byte code is saved in Hello.class. To run the program, the programmer uses the command *java* to interpret the byte code. The result of the program is displayed on the screen.

1.5 Displaying two messages

- You can put as many statements as you like in *main*.
- You can also put comments at the end of a line.
- System.out.println appends a *newline*, which moves to the beginning of the next line. If you don't want a newline, you can use *System.out.print*.

1.6 Formatting Source Code

- In Java, some spaces are required, such as between words.
- Indents and Newlines are optional, however.
- Including Newlines and Spacing helps make your program easier to read.

1.7 Escape Sequences

- It is possible to display multiple lines of output in one line of code. You just have to tell Java where to put the line breaks.
- \n is an "escape sequence" that allows you to "escape" the String to write special characters
- Most commonly used escape sequences:
    - \n: newline
    - \t: tab
    - \": quote
    - \\: backslash

1.8 What is Computer Science?

- Being able to understand computation is much more valuable than just learning to write code.
- One of the most interesting aspects of writing programs is deciding how to solve the problem, especially if there are multiple solutions.
- *Algorithm:* A sequence of steps that specifies how to solve a problem.
- Computer Science is the study of algorithms, including their discovery and analysis. As you learn to develop algorithms for problems you haven't solved before, you will learn to think like a computer scientist.
- Writing code and designing algorithms is difficult and error-prone. Programming errors are called *bugs*, and the process of finding and fixing them is called *debugging*. As you debug programs, you will develop new problem-solving skills.

1.9 Debugging Programs

- When you are experimenting with a new feature, you should try to make mistakes, because it helps teach you what various error messages mean and what could go wrong.
- Debugging is like an experimental science: once you narrow down the problem, you make a hypothesis and test it. If your hypothesis was wrong, you can try again and make a new one.
- Every programmer has had to debug a program at some point, so don't hesitate to ask questions.