Lecture 1: Introduction.

Based on Chapter 1 in Introduction to Java Programming, Y. Daniel Liang, Brief Version, 9/E

Syllabus

The course syllabus is located at http://cs.nyu.edu/~joannakl/cs101.06_s14/syllabus.php.

Chapter 1: Introduction to Computers, Programs, and Java

You should be familiar with most, if not all, of the material in this chapter (from CSCI-UA 002 or equivalent).

We will review quickly selected sections:

- 1.6. The Java Language Specification, API, JDK, and IDE.
- 1.7. A Simple Java Program
- 1.8. Creating, Compiling and Executing a Java Program.

Chapter 1 sec. 6

Java language specification - technical definition of the language that includes the syntax and semantics of the Java programming language;
http://docs.oracle.com/javase/specs/

Application program interface (API) - predefined classes and interfaces for developing Java programs;
http://docs.oracle.com/javase/7/docs/api/

Java is a full-fledged and powerful language that can be used in many ways. It comes in three editions:

- Java Standard Edition (Java SE) can be used to develop client-side standalone applications or applets.
- Java Enterprise Edition (Java EE) can be used to develop server-side applications, such as Java servlets and JavaServer Pages. Java Micro Edition (Java ME) can be used to develop applications for mobile devices, such as cell phones.
- Java Development Toolkit (JDK) - a set of separate programs, each invoked from a command line, for developing and testing Java programs.

Integrated development environment (IDE) - software that provides/combines many tools for rapidly developing (Java) programs: editing, compiling, building, debugging, and online help are integrated in one graphical user interface. Examples:

- NetBeans,
- Eclipse,
- Dr. Java,
- TextPad.
Chapter 1 sec. 7: A Simple Java Program

Listing 1: What does this code do: what does each line mean? what is the output?

```java
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Programming is fun!");
        System.out.println((10.5 + 3 * 2.5) / (7.5 - 4.5/3));
    }
}
```

Listing 1, ver.2:

```java
// name of the class, every Java program needs at least one class
public class Welcome {
    // main method of the class Welcome, that's where the program starts
    public static void main(String[] args) {
        // display the string
        System.out.println("Programming is fun!");
        // compute and display the number
        System.out.println((10.5 + 3 * 2.5) / (7.5 - 4.5/3));
    }
}
```

Output:

```
Programming is fun!
3.0
```
Chapter 1 sec. 8: Creating, Compiling and Executing a Java Program

<table>
<thead>
<tr>
<th>action</th>
<th>result</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create/Modify source code: use plain text editor</td>
<td>Source code files, ex. Welcome.java</td>
<td>high level language, human readable</td>
</tr>
<tr>
<td>Compile source code: javac Welcome.java</td>
<td>Bytecode, ex. Welcome.class</td>
<td>If errors occur go back to step 1 (compilation errors) low level language, similar to machine code, runs on JVM, plus: machine independent, minus: slow</td>
</tr>
<tr>
<td>Run bytecode: java Welcome</td>
<td></td>
<td>If errors occur go back to step 1 (run-time errors)</td>
</tr>
</tbody>
</table>

Chapter 1: Homework