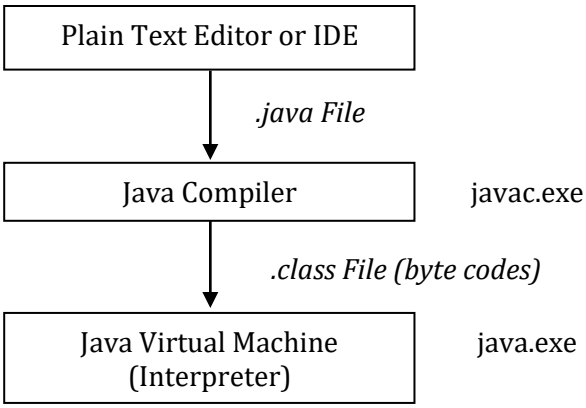
Running Java Programs

In this handout, the compilation and execution of Java programs is explored.

The Java Compilation Process

The creation of a Java program generally follows the steps shown in the figure at right.



The code is developed in either a plain text editor or an **I**ntegrated **D**evelopment **E**nvironment (IDE) and saved to a file with a **.java** extension. It is then checked for syntax errors by the Java Compiler (**javac.exe**). If successfully compiled, the resultant **.class** file is passed to the Java Virtual Machine (**java.exe**) for execution.

The act of compiling has a two-fold effect: Firstly, the code is checked for syntax errors and secondly, the code is converted to byte codes. The byte codes are instructions for an imaginary machine called the **Java Virtual Machine** (JVM). This machine is emulated by all Java interpreters and therefore allows you to execute a compiled Java program among different platforms (operating systems) with a JVM.

Setup

In order to compile and run Java programs, you must first obtain a copy of the latest version of the Java Software Development Kit (JDK), version 1.8.X. The JDK can be found at http://java.sun.com/j2se/ or http://www.oracle.com/technetwork/java/javase/overview/index.html. The JDK contains the fundamental tools needed for compiling, running, documenting and archiving Java code. All of the tools are command-line based tools and so can only be executed from the command prompt. In this handout you will be mainly concerned with the two tools mentioned previously: **javac.exe** for compiling and **java.exe** for running Java programs. After installing the JDK, you will need to update the **path** environment variable. For your own personal machines, the following works for MS-Windows XP.

1. Click the *Start* button.
2. Click *Settings* > *Control Panel* > *System*.
3. Select the tab called *Advanced* and then click on the *Environment Variables* button.
4. At the bottom, click on the *Path* variable and then click the *Edit* button.
5. At the end of the variable value add: `;c:\Program Files\Java\jdk1.8.0\bin`
6. In this example we assume that the JDK was installed in the directory `c:\Program Files\Java\jdk1.8.0`. (*you may need to update the version number here to match your download e.g. jdk1.8.1*)

7. Click the *Ok* buttons until the applet closes.

In the lab you will need to create a batch file. First create a text file called **jpath.bat** on your flash drive, desktop or preferred directory (Note: the filename does not matter as long as it ends in the **.bat** extension and is not the name of an application or a command). In this text file type the following line.

```
set path=c:\Program Files\java\jdk1.8.0\bin;%path%
```

In this example we assume that the JDK was installed in the directory `c:\Program Files\Java\jdk1.8.0\bin`. Save the file. Now each time you open a command-prompt window to use the JDK, type the following before starting.

```
jpath
```

This needs to be performed **once** for each command-prompt window opened.

A Sample Run

Now it is time to compile and run a Java program. To do so, simply perform the following steps.

1. Start NOTEPAD, any plain-text editor or your favourite Java Integrated Development Environment.

2. Type or copy the following program:

```
public class GreetingTheWorld
{
    public static void main( String args[] )
    {
        System.out.println( "Greetings to the WORLD" );
    }
}
```

3. Save the program in a file called: **GreetingTheWorld.java**.

   Note: The file must have the same name as the class and must end with **.java**. Also, remember that Java is case sensitive and so the file name must have the same case as the class name.

4. Open a command-prompt window. If you are in the lab run the **jpath** batch file as described in the previous section.

5. Before the program can be executed, it must be compiled.

   At the prompt type the following: `javac GreetingTheWorld.java`

   The program is compiled and the class file **GreetingTheWorld.class** should have been created. Remember, that in the lab you must run the **jpath** batch file as described previously.

6. To execute the program type: `java GreetingTheWorld`

   Note that the **.class** extension is not needed since the interpreter assumes that the file will have this extension.

7. The program should output the following: `Greetings to the WORLD`