Getting Started

Workshop 1

In this workshop, you'll learn:

- How to start a simple java application.

Part 1: Preparation

1. Updating the PATH Environment Variable

If you do not set the PATH variable, you need to specify the full path to the executable file every time you run it, such as:

```
C:\> "C:\Program Files\Java\jdk1.8.0\bin\javac" MyClass.java
```

It is useful to set the PATH variable permanently so it will persist after rebooting.

To set the PATH variable permanently, add the full path of the jdk1.8.0\bin directory to the PATH variable. Typically, this full path looks something like C:\Program Files\Java\jdk1.8.0\bin. Set the PATH variable as follows on Microsoft Windows:

a. Click **Start**, then **Control Panel**, then **System**.

b. Click **Advanced**, then **Environment Variables**.

c. Add the location of the bin folder of the JDK installation to the **PATH variable** in **System Variables**. The following is a typical value for the PATH variable:

```
C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files\Java\jdk1.8.0\bin
```

Note:

- The PATH environment variable is a series of directories separated by semicolons (;) and is not case-sensitive. Microsoft Windows looks for programs in the PATH directories in order, from left to right.
- You should only have one bin directory for a JDK in the path at a time. Those following the first instance are ignored.
- If you are not sure where to add the JDK path, append it.
- The new path takes effect in each new command window you open after setting the PATH variable.

2. **Let us look at a simple code that would print the words *Hello World*.**

```
publicclassMyFirstJavaProgram{
/* This is my first java program.
   * This will print 'Hello World' as the output
   */
publicstaticvoid main(String[]args){
    System.out.println("Hello World");// prints Hello World
  }
}
```

Let's look at how to save the file, compile and run the program. Follow the steps given below:

- Open notepad and add the code as above.
- Save the file as: MyFirstJavaProgram.java.
- Open a command prompt window and go to the directory where you saved the class. Assume it's C:\.
- Type **javac MyFirstJavaProgram.java** and press enter to compile your code. If there are no errors in your code, the command prompt will take you to the next line (Assumption : The path variable is set).
- Now, type **java MyFirstJavaProgram** to run your program.
- You will see **Hello World** on the window.

```
C :>javac MyFirstJavaProgram.java
C :> java MyFirstJavaProgram
HelloWorld
```

Part 2: Workshop requirements

1. Write a Java program to convert the starting time in hours and minutes to the equivalent **total minutes** :

- For example: `2` hours and `30` minutes would be `150` total minutes
- For example: `5` hours and `15` minutes would be `315` total minutes
- Hint: use multiplication and addition

2. Write a Java program that can serve as a ***ending time*** calculator. The user enters the starting time in hours and minutes, a duration in total minutes, and your program will calculate and display the ending time (as hours:minutes).

For example, if an event starts at `2 30` and lasts `125` minutes, it will end at `4 35`

Note: To simplify the problem, assume military time (0..23) rather than standard time, in which you would need to worry about a.m. and p.m.

3. Use `String` variables and string concatenation

a. Write a Java program named `NameMaker.java` with the String variables `firstName`, `middleName`, `lastName`, and `fullName`
b. Prompt the user to enter their first, middle, and last names and read the names from the **keyboard**
c. Use string **concatenation** to set and display their `fullName` as firstName + a blank char + middleName + a blank char + lastName

```
Enter your first name: Edward
Enter your middle name: Michael
Enter your last name: Gellenbeck

Hello Edward Michael Gellenbeck
```