Getting Started with Java

Hello World program

for

Microsoft Windows

"Hello World!" for Microsoft Windows

- The following instructions are for users of Windows Vista, Windows 7, Windows 8 and Windows 10
- If you encounter problems with the instructions, go to on https://docs.oracle.com/javase/tutorial/getStarted/problems/index.html to find out about common problems and solutions
- These instructions consist of
  - A Checklist
  - Creating Your First Application
    - Create a Source File
    - Compile the Source File into a .class File
    - Run the Program

A Checklist

To write your first program, you'll need:

- **The Java SE Development Kit 8 (JDK 8)**
  - **This should have already been installed on your PC.**

- **A text editor**
  - In this example, we'll use Notepad, a simple editor included with the Windows platforms. You can easily adapt these instructions if you use a different text editor.

These two items are all you'll need to write your first application.

Creating Your First Application

Your first application, HelloWorldApp, will simply display the greeting "Hello world!". To create this program, you will:

- **Create a source file**
  - A source file contains code, written in the Java programming language, that you and other programmers can understand. You can use any text editor to create and edit source files.
- Compile the source file into a .class file
  - The Java programming language compiler (javac) takes your source file and translates its text into instructions that the Java virtual machine can understand. The instructions contained within this file are known as bytecodes.
- Run the program
  - The Java application launcher tool (java) uses the Java virtual machine to run your application.

Create a Source File

First, start your editor. You can launch the Notepad editor from the **Start** menu by selecting **Programs > Accessories > Notepad**.

In a new document, type in the following code:

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

Create a Source File – Cont.

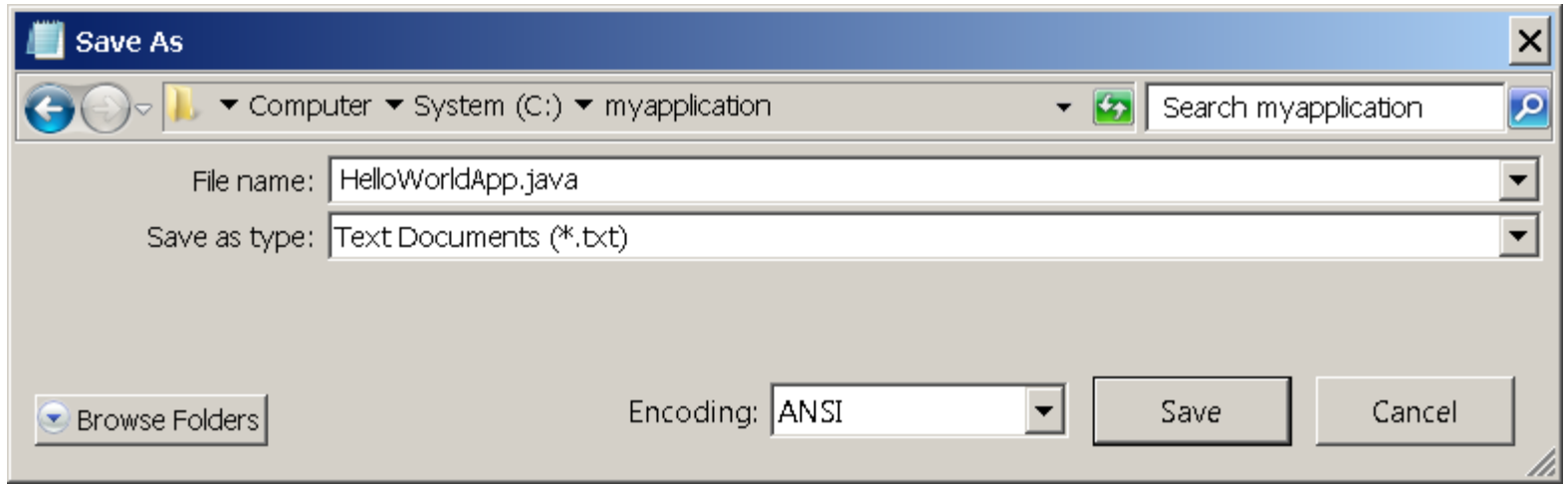
Save the code in a file with the name HelloWorldApp.java. To do this in Notepad, first choose the File > Save As menu item.

Then, in the Save As dialog box:

1. Using the Save in combo box, specify the folder (directory) where you'll save your file. In this example, the directory is myapplication on the C drive.
2. In the File name text field, type "HelloWorldApp.java", including the quotation marks.
3. From the Save as type combo box, choose Text Documents (*.txt).
4. In the Encoding combo box, leave the encoding as ANSI.

Create a Source File - Cont.

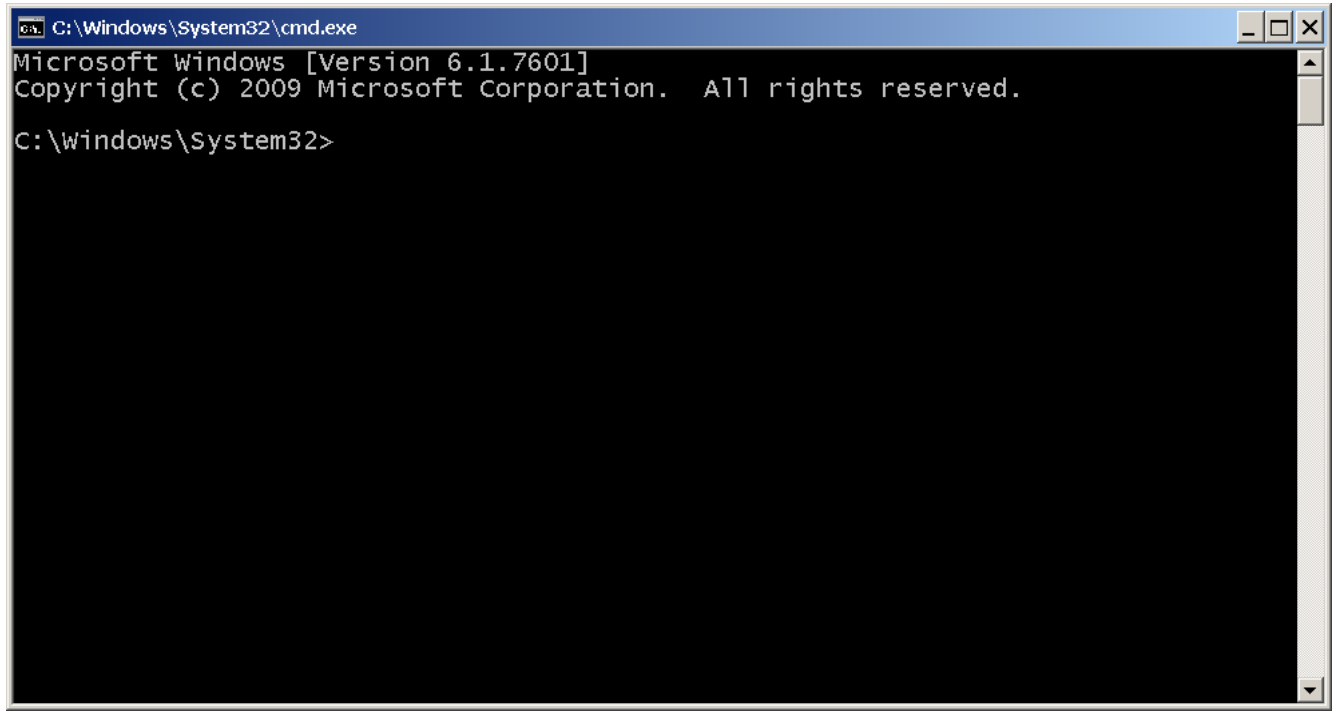
When you're finished, the dialog box should look like this



Now click Save, and exit Notepad.

Compile the Source File into a .class File - 1

Bring up a shell, or "command," window. You can do this from the Start menu by choosing Run... and then entering cmd. The shell window should look similar to the following figure.



The prompt shows your *current directory*. When you bring up the prompt, your current directory is usually your home directory for Windows

Compile the Source File into a .class File - 2

- To compile your source file, change your current directory to the directory where your file is located.
- For example, if your source directory is myapplication on the C drive, type the following command at the prompt and press Enter:

```
cd C:\myapplication
```

Now the prompt should change to C:\myapplication>.

Compile the Source File into a .class File - 3

If you enter `dir` at the prompt, you should see your source file, as follows:

```
C:\>cd myapplication

C:\myapplication>dir
 Volume in drive C is System
 Volume Serial Number is F2E8-C8CC

 Directory of C:\myapplication

2014-04-24  01:34 PM    <DIR>          .
2014-04-24  01:34 PM    <DIR>          ..
2014-04-24  01:34 PM               267 HelloWorldApp.java
               1 File(s)            267 bytes
               2 Dir(s)  93,297,991,680 bytes free

C:\myapplication>
```

Now you are ready to compile. At the prompt, type the following command and press **Enter**.

```
javac HelloWorldApp.java
```

Common Error

`'javac' is not recognized as an internal or external command, operable program or batch file`

If you receive this error, Windows cannot find the compiler (`javac`).

Here's one way to tell Windows where to find `javac`. Suppose you installed the JDK in `C:\jdk1.8.0`. At the prompt you would type the following command and press Enter:

```
C:\jdk1.8.0\bin\javac HelloWorldApp.java
```

If you choose this option, you'll have to precede your `javac` and `java` commands with `C:\jdk1.8.0\bin\` each time you compile or run a program.

Updating the PATH Environment Variable - 1

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:

Updating the PATH Environment Variable - 2

1. Click Start, then Control Panel, then System.
2. Click Advanced, then Environment Variables.
3. 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:
4. C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files\Java\jdk1.8.0\bin

NOTE: You may not have sufficient permissions to allow you to make this change.

If everything has worked…

The compiler will have generated a bytecode file, HelloWorldApp.class. At the prompt, type dir to see the new file that was generated as follows:

```
C:\myapplication>javac HelloWorldApp.java

C:\myapplication>dir
 Volume in drive C is System
 Volume Serial Number is F2E8-C8CC

 Directory of C:\myapplication

2014-04-24  02:07 PM    <DIR>          .
2014-04-24  02:07 PM    <DIR>          ..
2014-04-24  02:07 PM               432 HelloWorldApp.class
2014-04-24  01:34 PM               267 HelloWorldApp.java
               2 File(s)            699 bytes
               2 Dir(s)  93,298,032,640 bytes free

C:\myapplication>
```

Running the Program

Now that you have a .class file, you can run your program

In the same directory, enter the following command at the prompt:

java -cp . HelloWorldApp

You should see the following on your screen:

```
C:\myapplication>java -cp . HelloWorldApp
Hello World!

C:\myapplication>
```

Congratulations! Your program works!