

## 3 Your First JAVA program

*You should work through this section sitting at a terminal and make sure you get it to work.*

As is *traditional* in learning any programming language that your first program prints ‘ ‘Hello World!’ ’ to the screen. The code of such a simple program is shown below:

```
//  
//          Simple "Hello World"  
//  
  
public class Hello{  
    public static void main(String args[]){  
        System.out.println("Hello World !");  
        System.exit(0);  
    }  
}
```

Look at the program in detail:

1. The first three lines start with `//` which are *comments*. They play no part in the actual program but are there to make the code more humanly readable.
2. The line `public class Hello` declares a *class* called `Hello`, which contains your code, in this case a program to write `Hello World !`. All programs must be declared as a *class*, more on this much later. Note that the whole class is enclosed in `{ }` brackets.
3. The `public static void main(String args[])` statement is the start of the actual program. Use this as a “recipe” at the moment, all main programs in this course will start with this.
4. The whole program is enclosed in `{ }` brackets.
5. The `System.out.println();` statement says to print the string `"Hello World !"` to the standard output (which is your screen).
6. The `System.exit( 0 );` statement stops the program and return a “success” code to the operating system.

### 3.1 Your First Program

*You are strongly recommended to follow this task.*

In an *active* terminal window,

1. Make a directory called `sciprog` to hold your computing examples by typing:

```
mkdir sciprog
```

2. Make this new directory your *present working directory* by typing:

```
cd sciprog
```

---

### 3. Now type

```
emacs Hello.java &
```

to start the editor. Note the “&” character means “run the editor in background” which will allow you to use the terminal window at the same time. Type in and SAVE BUFFER the above program. For help on this see the *emacs under X-Windows* section.

### 4. *Compile* the program with

```
javac Hello.java
```

which will create an *byte code* file called “Hello.class”. (check that this file exists with the `ls` command)

### 5. To run the program type

```
java Hello <Return>.
```

If “Hello World !” appeared, then ..., congratulations, you have just written your first JAVA program.

## Examples

The following on-line examples are available:

1. Source of HelloWorld

## What Next?

You are now ready to undertake Checkpoint 1 detailed in the next section.