CS620c Structured Programming Lesson 3

Joe Duffin

Email: Joseph.Duffin@nuim.ie

Last lecture revisited..

- You should always spend time planning what your program needs to be able to do
  - decide what the component parts are and
  - how to order them
- **An algorithm is a detailed sequence of steps that is required to solve a problem**
- A recipe in a cook book is an algorithm, it details
  A. The materials (ingredients) needed to make the recipe (solve the problem)
  B. The instructions (steps) to follow.

More on Algorithms

Think about the following:

- Matching socks
- Getting the average of X numbers
- Ordering your CD collection
- Doing your weekly shopping

Java – what is Java?

- Java is a programming language
- Java can be used to write
  - Applications
  - Applets
- It is relatively simple to understand
- It is an object-oriented language
- Platform-independent because of the JVM
  - "write once, run anywhere“
- Many useful libraries available
  - for music, 3D, internet…

Java – Getting started

What we are using in class for CS620c (first two bullet points)

- Download the Java Software Development Kit (sdk) for free from http://www.java.com/en/download/manual.jsp
- Download NotePad++ for free https://notepad-plus-plus.org/download/v6.9.2.html

Other ways of developing Java programs (you may still have to download the Java SDK in addition to these IDEs).

- Download JCreator LE from: http://www.jcreator.org/download.htm
- Download BlueJ from: http://www.bluej.org/download/download.html

Writing a Java program (essential)

- Write the program in plain text format and save with a **.java** file extension
- But, the machine won't understand this.
- So, compile the program to turn it into Java **bytecode**. The bytecode is stored in **.class** file
  - Bytecode is a highly optimised set of instructions designed to be executed by a JVM.
- A JVM interprets the bytecode and runs it on the machine

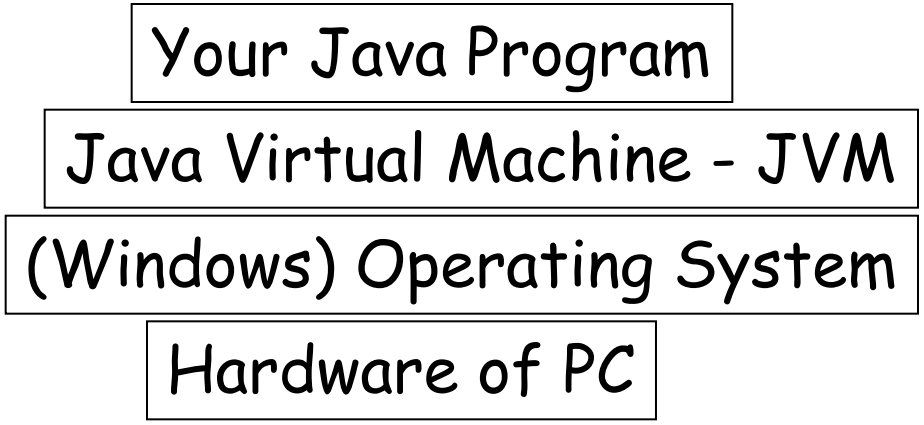
- The compiler is called javac and the JVM interpreter is called java

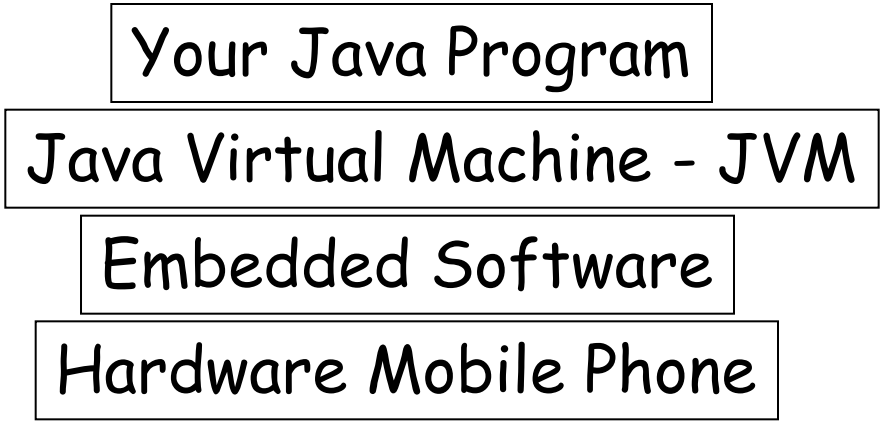
JVM – Java Virtual Machine

- The JVM is just a software program
- Allows the same `.java` files to run on any machine
  - Windows pc
  - Mobile phone
  - Mac computers
  - Linux pc…

Java Virtual Machine

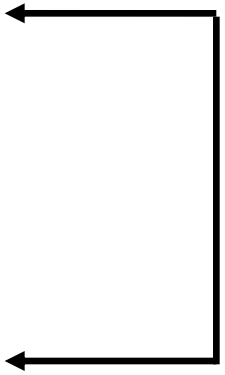
- Java is portable because it relies on layer of Software and Hardware
  - Especially the JVM
  - Each layer only interacts with neighbouring layers





Steps to writing programs

1. Develop an algorithm
2. Write a software implementation of the algorithm – a software program
3. Compile it
4. Fix any compilation errors
5. Test it – try to run it
6. Fix any runtime errors



Repeat

steps 3 to 6

as necessary

Java – Simple program

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

- Create the source file with a text editor
- Save as `Hello.java`
- Compile the source into bytecode
  - `javac Hello.java`
- Interpret the bytecode
  - `java Hello`

“Hello World” – Word by word, line by line

Comment: this code does not run it is only used to make code more readable, add a note or explanation (or perhaps disable a piece of code).

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

Comments:

1/ Consider including your name, the purpose of the program and date of last modification in your comments.

2/ // Hello World is also valid to comment individual lines.

“Hello World” – Word by word, line by line

public: this is a “modifier” placed before the word class that makes the class visible (accessible) from outside the class.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

public: 1/ In Java a source code file can contain at most one public class.

2/ In a private class methods and variables can only be accessed by methods within the same private class (but not by other methods contained in other classes, (classes can be grouped to make a “package”)

3/ Classes can also be protected (more later).

“Hello World” – Word by word, line by line

class: This is the description of an object, you can have many instances of an object. For now you could think of it as a container into which you put the description of an object. Much more about objects soon.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

class: 1/ Describes an object, you can then create an “instance” of the object.

“Hello World” – Word by word, line by line

Helloworld:This is the name the programmer has given to the class.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

class name:

1/ The class Helloworld must be contained in the file Helloworld.java (the public class and file name must have the same name).

2/ The code belonging to this class is contained between the curly braces {}.

“Hello World” – Word by word, line by line

public: Modifier makes the method visible outside the class, essentially to the JVM (Java Virtual Machine) or OS (OperatingSystem).

```
/* Hello World */

public class Hel    orld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

public: 1/ The main() method needs to be accessible to the JVM (which is outside the class) so that it can be called.

“Hello World” – Word by word, line by line

static: Forces an instance of the main() method to exist so that it can be called from outside the class.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

“Hello World” – Word by word, line by line

void: this tells us that the main method (function) does not return any values.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

void: Most functions (Java\C# programmers call them methods) that you will have met such as sine or cosine are given one number and return another number. Thus sine(90) would return 1. However the main() method does not return anything (it just prints to the screen in this example) and so we need to tell the compiler that it is a “void” function.

“Hello World” – Word by word, line by line

main: this is the method that the JVM looks for and provides the entry point of your program.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

main: the main() method is the starting point in many programming languages including Java, C# and C++.

“Hello World” – Word by word, line by line

String: is a sequence of text characters. It is the argument for the main() method

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

String args[]:

1/ If you type java Helloworld abc 123<ret> on the command line to run the program then the args array will contain “abc” in args[0] and “123” in args[1]. This provides a way of passing data into your program from the initial call to start it.

2/ args.length = the number of arguments passed in the command line.

3/ args is the name used normally but it could be anything (e.g. info).

“Hello World” – Word by word, line by line

System.out: is an object used for printing to screen.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

System.out:

1/ System is a predefined class that provides access to the system. out is the output stream that is connected to the console.

2/ System.out is an object.

3/ As you may have guessed n = System.in.read(name); is one method for reading from the keyboard, n=number of characters read, and name is the string array containing them.

“Hello World” – Word by word, line by line

println: is a method belonging to the System.out class/object that can print a string to the console followed by a line feed and carriage return

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

“Hello World” – Word by word, line by line

“Hello, World!” is an argument passed to the println() method. The argument is a string of characters contained in brackets.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

“Hello, World!”: 1/ This is the argument for the println() method. An argument is the one (or more) pieces of information passed to method.

“Hello World” – Word by word, line by line

For a beginner this is the main line of code that needs to be created and understood.

```
/* Hello World */

public class Helloworld
{
        public static void main (String args[])
        {
                System.out.println("Hello, World!");
        }
}
```

and this is “boiler plate”, code that holds it all together, we can wait for better understanding as the module proceeds.

Code Indentation

K&R: Bracket and Method call

```
/* Hello World */

public class Helloworld
{
      public static void main (String args[])
      {
            System.out.println("Hello, World!");
      }
}
```

Kernel Normal Form style

```
/* Hello World */

public class Helloworld{

       public static void main (String args[]){

       System.out.println("Hello, World!");
      }
}
```

What have we covered today?

- Re-visited algorithms
- Introduced Java
- Learnt the template for a Java program
- How to write programs to print messages to the screen
- How to compile and run a program