Introduction to Java

Learning Objectives

- To introduce and become familiar with the following:
  - The structure of a Java program (application)
  - Java class naming standards
  - Program LAYOUT
  - Program COMMENTS
  - Program READABILITY
  - The main METHOD
  - Printing String output from a program

What is a Computer Program?

- A computer program is a sequence of statements intended to accomplish a task
- Programming is a process of planning and creating a program

MyFirstJavaProgram.java

```
1   /**
2    * Created by: Martin
3    * Created on: 09/09/17
4    * Program to print out "My First Java Program"
5    */
6
7   public class MyFirstJavaProgram {
8      public static void main (String [] args){
9         System.out.println("My First Java Program");
10     }//main
11  }//class
```

Comment
(lines 1-5)

Class
(lines 7-11)

Comments

- Comments provide extra information to make a program easier to understand and more maintainable

- **Comments** are not really lines of code
  - They are NOT executed (performed)
  - They are used to improve the READABILITY of our code

- Typically you will find comments in the:
  - Header (lines 1-5)
  - Declarations
  - Bracket labels (lines 10, 11)
  - Complex sections of a program

Comments

Multiple-line comments (lines 1-5)

- Enclosed between /* and */
- The compiler ignores everything between /* and */

Single-line comments (lines 10 and 11)

- Begin with //
- **Can be placed anywhere in the line**
- Everything encountered in that line following the

  // is ignored by the compiler

Important

Good Programming Practice requires

ALL programs to have

a header comment (cf lines 1-5)

identifying information

such as

the name of the **author**,

the **creation date** and

a **brief description of what the program does**

Class

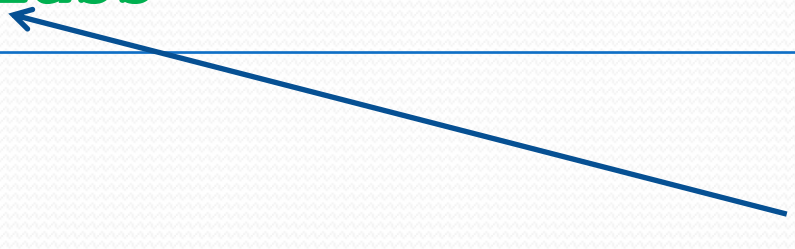
- The basic unit of a Java program is called a class
- The first line of a class consists of the reserved words **public** **and** **class** followed by the class identifier (chosen by the programmer)
- This **class identifier** should be meaningful to the purpose of the program
- The contents of a class must be placed inside a pair of curly brackets { . . . }

Class Structure

```
public class <class identifier> {

      contents of class;

}//class
```



Executing a Java program

- There is a specific instruction that will be looked for to begin execution
- To enable the program to be executed (or run) the class **MUST** contain a **main() method**
- A main method is a section of program
- A simple program will consist of a single class containing a main method and can be referred to as an **Application Program** **or** **Class**

The main() method In Java

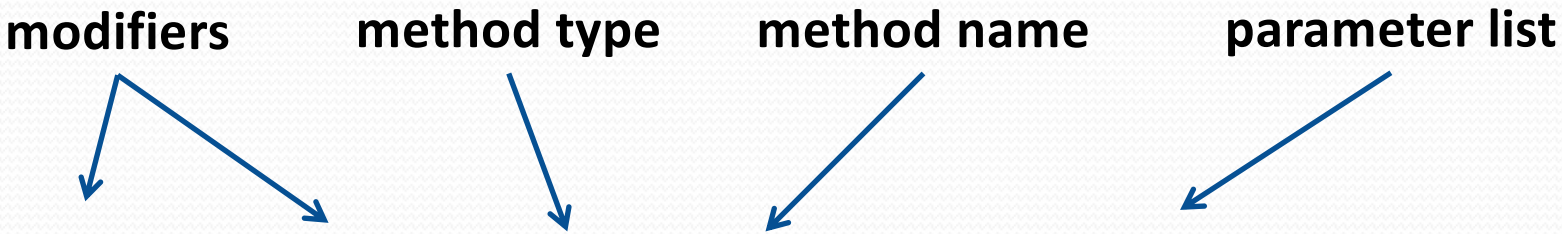
- A main method consists of a header (which never changes) followed by a sequence of statements held inside a set of curly brackets

- The header of a main method *is always the same:*

```
public static void main (String [] args)
```

- **`public`**, **`static`**, **`void`** and **`main`** are RESERVED WORDS
  - Dealt with later
  - For now simply ALWAYS use the same header for a main method

Structure of a main() method




```
public static void main (String [] args) {
    Sequence of Instructions;
    One per line;
    Each separated by a semicolon;
}//main
```

- The *main method* is a pointer to Java, to *highlight to the computer* where the instructions to be performed start (application entry point)

- The instructions are designed by you (the programmer)

Putting it all together

- Putting all the components together, we have:

  The **header comment**

  The structure of a class

  (which contains a **main** method)

- The result is shown overleaf

Typical Structure

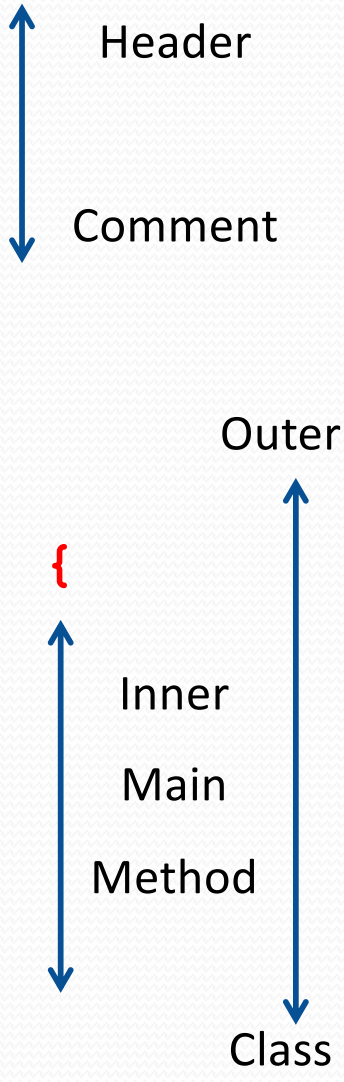
```
/**
 * Created by:     Your Name
 * Created on:     Creation Date
 * General description of program
 */

public class <class identifier> {

   public static void main (String [] args) {

      Sequence of Instructions;

      One per line
      Each separated by a semicolon;
   }//main
}//class
```



Good Design

- Note the position of the curly brackets for BOTH the class and the main() method
  - Be consistent - choose a style and set it up in IntelliJ via the preference settings
- We label the closing curly bracket with identification
  - Using a suitable comment e.g. the name of the class
  - This helps as programs get bigger
- Note the indentation of the main method within the class
  - Set in IntelliJ via the preference settings
- Statements within the main method are indented accordingly

LetterE1.java

```
1   /**
2    * Created by:        Martin
3    * Created on:        09/09/18
4    * Program to display a capital 'E' using asterisks
5    */
6
7   public class LetterE1 {
8
9      public static void main (String [] args) {
10        System.out.println("*****");
11        System.out.println("*");
12        System.out.println("*");
13        System.out.println("****");
14        System.out.println("*");
15        System.out.println("*");
16        System.out.println("*****");
17     }//main
18  }//class
```

File Names

- What file name should we save this program as?

- QUESTION:

  Why would Java not accept the following instruction?

  ```
  public Static void Main (String [] args)
  ```

- ANSWER: JaVa Is CaSe SeNsItIvE

- The name or **identifier** of a **program or class** is chosen by the programmer - for this program it is **LetterE1**

File Name (Class Identifier)

The program name (class identifier) selected should conform to the following rules:

- MUST begin with an **UPPERCASE letter** of the alphabet
- This may be followed by any number of letters or numerals or the underscore character (_)
- The program should use a class identifier which is **meaningful** i.e. indicates the purpose of the program

Using The Underscore

- Underscores may be used to break up long identifiers
- Often used when an identifier is really 2 or more English words
- EG A program to add a list of numbers might be given the name

  `Add_numbers`

- A preferable way of breaking up a long name is to use a mixture of upper and lower case letters
- Hence, the same program could have been called: `AddNumbers`
- Make your choice, but be consistent

Remember

- Programs or classes are saved in a file with the same primary name as the IDENTIFIER name given to the class within the program
- Programs written in Java will be given the following file extension .java

- Our first program will automatically be saved as:

  LetterE1.java

- Our second program will be saved as:

  LetterE2.java

Running a Java Program

- When you run LetterE1 you get the following output:

```
*****
*
*
****
*
*
*****
```

- **The following file is produced:**

LetterE1.class

LetterE1.java (revisited)

```
/** Created by: Martin Created on:09/09/18 Program
to display a capital E using asterisks */ public
class LetterE1 {public static void main (String []
args){System.out.println("*****");System.out.println
("*");System.out.println("*");System.out.println("**
**");System.out.println("*");System.out.println("*")
;System.out.println("*****");}//main
}//class
```

- What is the difference between the two versions?
- What do you think will happen if you typed this into IntelliJ and ran it?
- Which is the more readable?

print() and println() methods

Consider the following statement:

```
System.out.println("*****");
```

Output:

```
*****
■
```

Consider the following statement:

```
System.out.print("*****");
```

Output:

```
*****■
```

Consider the following statement:

```
System.out.print("*****\n");
```

Output:

```
*****
■
```

LetterE2.java

```
1   /**
2    * Created by:        Martin
3    * Created on:        09/09/18
4    * Program to display a capital 'E' using asterisks
5   */
6
7   public class LetterE2 {
8
9      public static void main (String [] args) {
10        System.out.print("*****\n");
11        System.out.print("*\n");
12        System.out.print("*\n");
13        System.out.print("****\n");
14        System.out.print("*\n");
15        System.out.print("*\n");
16        System.out.print("*****\n");
17     }//main
18  }//class
```

LetterE2 Output

- When you run LetterE2 you get the following output:

```
*****
*
*
****
*
*
*****
```

Escape Characters

- In Java there are a number of escape characters used to control output

	Escape Sequence	Description
\n	Newline	Cursor moves to the start of the next line
\t	Tab	Cursor moves to the next tab stop
\b	Backspace	Cursor moves one space left
\\	Backslash	Backslash is printed
\'	Single quotation	Single quotation mark is printed
\"	Double quotation	Double quotation mark is printed

Escape Characters

What would be the output from the following statements?

```
System.out.print("\tM\ta\tr\tt\ti\tn");

    M   a   r   t   i   n
```

```
System.out.print("My first name is \t\t\'Martin\'\n");

My first name is        'Martin'

```

```
System.out.println("\"\\t\" is used to tab output
   \n\"\\n\" is used to take a new line");

"\t" is used to tab output
"\n" is used to take a new line

```

LetterE3.java

```
1   /**
2    * Created by:         Martin
3    * Created on:         09/09/18
4     * Program to display two capital Es using stars
5     */
6
7   public class LetterE3 {
8
9      public static void main (String [] args) {
10        System.out.print("*****\t*****\n");
11        System.out.println("*\t\t*");
12        System.out.print("*\t\t*\n");
13        System.out.println("****\t****");
14        System.out.println("*\t\t*");
15        System.out.print("*\t\t*\n");
16        System.out.println("*****\t*****");
17     }//main
18  }//class
```

LetterE3 Output

- When you run LetterE3 you get the following output:

```
*****    *****
*        *
*        *
****     ****
*        *
*        *
*****    *****
```

Countdown.java

```
1   /**
2    * Created by:  Martin
3    * Created on:  09/09/18
4    * Program to count down from three to one
5    */
6
7   public class Countdown {
8
9      public static void main (String [] args) {
10        System.out.print("Three... ");
11        System.out.print("Two... ");
12        System.out.print("One... ");
13        System.out.print("Zero... ") ;
14        System.out.println("Liftoff!");
15        System.out.println("Houston, we have a problem!");
16     }//main
17  }//class
```

Countdown Output

- When you run Countdown you get the following output:

```
Three... Two... One... Zero... Liftoff!
Houston, we have a problem!
```

Countdown.java (with spaces)

```
public class Countdown {

   public static void main (String [] args) {
     System.out.print ("Three... ");
     System.out.print (    "Two. . . ");

     System.out.print ("One... " );

     System.out.print ("Zero... ") ;
     System.out.println(    "Liftoff!  ");
     System.out.print("   Houston, we have a problem! " );
   }//main
}//class
```

Output:

```
Three... Two. . . One... Zero... Liftoff!
Houston, we have a problem! ■
```

Think

- What does the first multi-line comment tell you about the Countdown program?
- Do you find Countdown more readable with the blank lines and additional spaces?
- Can Java cope with extra blank lines throughout Countdown?
- What is the name of class?
- What filename would this program be given?

Becoming a Successful Programmer

- Install IntelliJ on your own machine and ensure initial settings are implemented correctly
- Type in and run sample Java programs from a textbook or the web
- Ensure that as each topic is presented in lectures, adequate individual study time is allowed to read around the material and all lab exercises are completed
- Don’t be scared to make mistakes.
  - That is how you will learn
  - Practice makes perfect

Self-Questions

- What is the structure of a Java program?
- Why is it important to have comments in a program?
- What are the various forms of comments in Java?
- What information should be included in a header comment?
- What are the naming conventions for a class?
- Why is readability important in programming?
- What is the difference between print() and println()?
- What are escape characters and when are they used?