**Following on from Hello World...**

**Output to console**

Some **special characters****:** A newline (line feed) character (**'\n'**) and the Tab delimiter (‘**\t’**)

|  |
| --- |
| The **\n** is an escape sequence commonly used in programming languages, including Java, to represent a newline character. It is a special character that indicates the end of a line of text and moves the cursor to the beginning of the next line. When the \n escape sequence is encountered in a string, it is replaced by the newline character during execution or display.  System.out.println("Line 1\nLine 2"); System.out.print("Line 3\n");  System.out.print("Line 4");  Would output :  Line 1 Line 2 Line 3 Line 4  The **\t** is an escape sequence in Java and many other programming languages that represents the horizontal tab character. It is one of several escape sequences used to include special characters in strings or character literals. The horizontal tab character is a control character that causes the cursor to move to the next tab stop, which is typically at fixed intervals (e.g., every 4 or 8 spaces).  When you use \t in a Java string, it is replaced by the actual horizontal tab character when the string is displayed or printed. This character moves the cursor to the next tab stop, aligning the output to the next available position.  System.out.println("Hello\tworld!");  System.out.println("This\tis\ta\ttab\texample.");  Would output :  Hello world! This is a tab example. |

Modify the **HelloWorld.java** code to output the following text to the **console**:

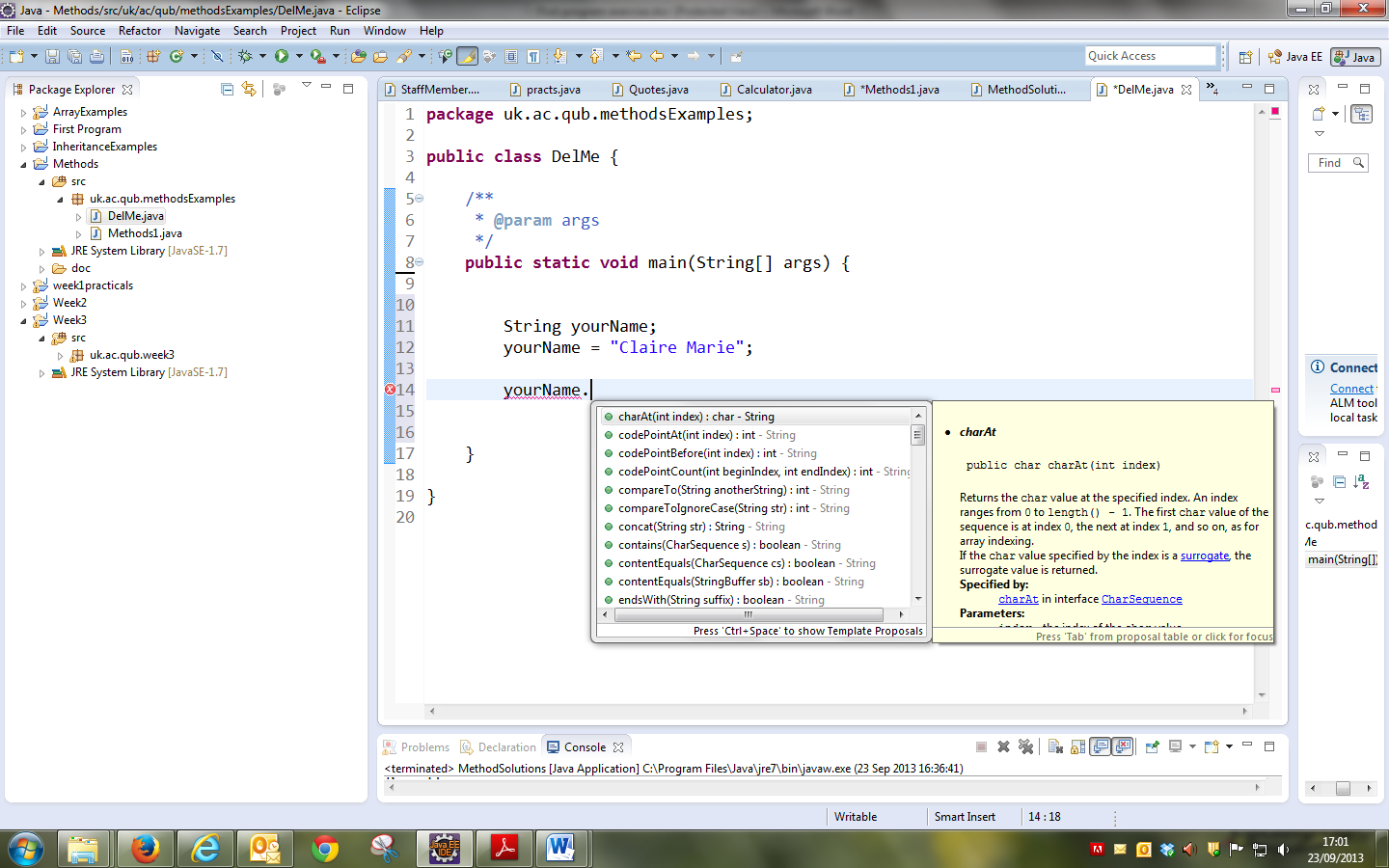
|  |
| --- |
| Hello World! I am alive  **Adding more text to the line**  **Adding a new line**  **Adding a blank line**  **Multiple lines**  **Hint use the tab \t character**  Hello World! I am alive  I can write on two lines!  Hello World! I am alive  I can write on two lines!  \*  \*\*\*  \*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  Happy Christmas  Macbeth  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  First Witch When shall we three meet again?  In thunder, lightning, or in rain?  Second Witch When the hurlyburly's done,  When the battle's lost and won.  Third Witch That will be ere the set of sun  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Strings

Java has many built-in **methods** (actions that can be performed) on an object – in this case a String. Once a string is declared, for instance like this …

String yourName;

yourName = “Claire Marie”;

You can then access methods that can be carried out on the String … Using the **dot operator** directly after the **variable identifier** (in this case **yourName**) in Eclipse will enable you to view the associated methods (**Code completion**).

Write a Java program that declares a **String**. Assign your **own name** to the String. Then output to screen the following information and perform the following operations on the string:

* Print out the **string**
* Print out the **number of characters** in the string
* Print out the **first** and **last** characters in the string (note the first character in the string is counted as zero the next is one etc…)
* Print out the string as **uppercase** characters
* Print out the string with all occurrences of the letter ‘**a**’ replaced by ‘**\***’ (if you don’t have an ‘a’ in your name then use a difference letter).
* Print out the position within the string of the **first** occurrence of the character **‘n**’.

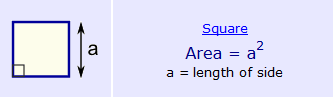
For example, if the input string is: **Aidan McGowan**

|  |
| --- |
| The input String : Aidan McGowan The number of characters in the string 13 The first character in the string is A and the second character in the string is n  The string in uppercase : AIDAN MCGOWAN The string with the letter a replaced with \* is : Aid\*n McGow\*n The character 'n' first occurs in position 4 |

**Declaring and using vars**

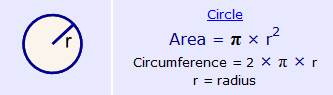
Primitive types – remember there are **8 types** in Java

**int**



Write a Java program that will declare the length of a side of a square as an **int** and calculate and print out the **area**. Set the length to **6**.

**double**



Write a Java program that will declare the radius of a circle as a **double** and calculate and print out the **area** (to three decimal points). Hint use **System.out.printf()…** consider **PI** to be **3.142.** Set the radius to **9.**

**Making decisions -** if statements

Write a Java program that will declare two **int** variables. Use the variable identifiers ***number1*** and ***number2***. Initialise ***number1*** to the value 3 and ***number2*** to the value 10. Use a *series* of **if** statements to print to screen the biggest value.

|  |
| --- |
| Numbers to compare 3 and 10 Biggest number 10 |

Now change ***number1*** to 44. Run the program again and check if the biggest value is identified.

|  |
| --- |
| Numbers to compare 44 and 10 Biggest number 44 |

Now change both numbers to be the same. Run the program again does your program recognise this? If not change it to do so. (**hint** : use the **==** operator)

|  |
| --- |
| Numbers to compare 44 and 44 Numbers are the same |