**OO Programming** **Lab Exercise 5**

Before you start:

Create a folder called l**ab5** inside your personal **java** folder you created at the start. Save all your work for lab 5 in this folder.

Using JCreator (or a text editor of your choice):

Your understanding of the programs and concepts covered in the revisions are examined here, in particular:

* Writing method definitions
* Passing parameters to methods
* Returning values from methods
* Understanding method definitions from the Java API reference

### 

### Instructions:

* The following exercises should be saved under the filenames lab7q1.java for Q1, lab7q2.java for Q2 and so on.
* Test each method you write below by calling it from method **main**().

**Q1**. Write a method called **intro** which prints out the message

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Recap on Java methods \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Q2**. Write a method **powersOfTwo** that prints the first 10 powers of 2 (starting with 2). The method takes no parameters and does not return anything.

**Q3**. Write a method **sum100** that returns the sum of the integers from 1 to 100, inclusive.

**Q4**. Writing a method called **GBPtoEUR**() which accepts an integer value (Sterling) and returns a double value (Euros).

Conversion: Use whatever the latest exchange rate is (it keeps changing !).

**Q5**. Write a method called **power** which takes two integer values, say x and n. Method power should raise integer x to the power of n, where n is a positive integer. The calculated value of x to the power of n should be returned to the calling method.

For example, power(2,5) is 32.

**Q6**. Write a method called **search** which will search a String object for occurrences of a specified character. The user should input the string and search character which are then to be passed as parameters to method search. Finally, method search should return to the caller the number of occurrences of the character in the String. e.g., **search**("Hello", 'l') results in search returning a value of 2.

**Sample interaction 1:**

Enter a string: **This is a sample string**

Enter a search character: **s**

Character 's' occurs 4 times in the input string "This is a sample string"

**Sample interaction 2:**

Enter a string: **This is a sample string**

Enter a search character: **z**

Character 'z' does not occur in the input string "This is a sample string"

**Sample interaction 3 (user enters a space as the search character):**

Enter a string: **This is a sample string**

Enter a search character:

Character ' ' occurs 4 times in the input string

"This is a sample string"

**Q7**. Write a method called **multiConcat** that takes a String and an integer as parameters. Return a string that consists of the string parameter concatenated with itself **count** times, where **count** is the integer parameter. For example, if the parameter values are “hi” and 4, the return value is “hihihihi”. Return the original string if the integer parameter is less than 2. Demonstrate this by producing the following output from **main**() by repeated calls to **multiConcat**() :

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**Q8**. Write a method **factorial** that takes a single integer value and returns its factorial.