

## EE402 Java Introduction Tutorial

Q1. Write a Java class that uses an array of names e.g. {"Teresa Green", "Rick O'Shea", "Robin Banks", "Barry Cade", "Sam Pull"} and prints out a Hello message to each of the names on an individual line. There can be any number of names in the array. In this case:

```
Hello Teresa Green
Hello Rick O'Shea
Etc.
```

Q2. Write a Java function of the form `String firstTwo(String)`; that when given a string, returns the string made of its first two chars, so the String "Hello" yields "He". If the string is shorter than length 2, return whatever there is, so "X" yields "X", and the empty string "" yields the empty string "".

```
firstTwo("Hello") → "He"
firstTwo("abcdefg") → "ab"
firstTwo("ab") → "ab"
```

Q3. Write a Java function of the form `boolean hasBad(String)`; that when given a string, returns true if "bad" appears starting at index 0 or 1 in the string, such as with "badxxx" or "xbadxx" but not "xxbadxx". The string may be any length, including 0. Note: use the Java API documentation to find a suitable String method.

```
hasBad("badxx") → true
hasBad("xbadxx") → true
hasBad("xxbadxx") → false
```

Q4. Write a function in Java of the form `String nTwice(String, int)`; that when given a string and an int n, returns a string made of the first and last n chars from the string. The string length will be at least n characters long.

```
nTwice("Hello", 2) → "Helo"
nTwice("Chocolate", 3) → "Choate"
nTwice("Chocolate", 1) → "Ce"
```

Q5. Write a function of the form `int countX(String)`; that when given a string, computes **recursively** (no loops) in Java the number of lowercase 'x' chars in the string.

```
countX("xxhixx") → 4
countX("xhixhix") → 3
countX("hi") → 0
```

Q6. Write a function in Java that returns the winning numbers in next week's EuroMillions lottery draw. The function should have the form: `int[]`

lotteryNumbers(); where the function uses the Math class random() method to return an array of 7 integers, the first 5 between 1 and 50 and the last two between 1 and 11. The numbers need not be unique.

Q7. Write a Java command line application that takes two Integer numbers from the command line, multiplies them together and displays the result.

Q8. Write a Java class called Statistics that has two states, an array of doubles and an int arrayLength state that will store the length of the array. Add a constructor and a display method that displays the contents of the array. You should be able to construct an object of your Statistics class by using:

```
double[] anArray = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
Statistics s = new Statistics(anArray);  
s.display();
```

and this will display:

```
Array [1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 ]
```

Q9. Modify your class in Q8 to add an average(), max() and min() methods, which each return the average of the array, the maximum and the minimum value in the array respectively.

Q10. Add a double[] sort() and double median() methods to your statistics class that take advantage of the sort algorithm that is available in the Arrays class (java.util.Arrays). The median is the centre value after sorting an array. However, if there is an even number of elements then it is the average of the pair of values at the centre.