## ECS510 Algorithms and Data Structures in an Object-Oriented Framework

## Exercise Sheet 2: Operations on arrays and specification issues

This is a set of exercises to support the "Using Arrays" section.

1) Download the file UseArrays $1 . j a v a$ from the "Code Index" in the "Using Arrays" section. Check you can compile and run this code. You can make use of the support code here to test the methods on arrays of integers you have to write for the rest of this exercise. When you are asked to "write a static method", all the data the method uses should be passed to it in arguments, it should not make any use of static variables declared outside the method. The method should not read anything from the screen or write anything; reading and writing should only be done by the front-end code that allows the method to be demonstrated. It is permissible to write a "helper method" to assist in the implementation of the method asked for.
2) Write a static method that takes an array of integers and a separate integer and returns the number of integers in the array whose value is less than the separate integer. So if the array is

| 8 | 11 | 7 | 12 | 5 | 10 | 5 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |

and the separate integer is 9 , the method should return 4.
Note, the example clarifies an ambiguity in the specification, which is what to do if a particular integer value occurs more than once. The word "integer" in the specification could mean an integer value, or it could mean an occurrence of an integer value. In the example 8,5 and 7 are less than 9 , and 5 occurs twice while 7 and 8 occur once. So when you are told 4 is returned, it indicates there is an assumption that the word "integer" is used to mean an occurrence of an integer value. If you were told that 3 should be returned, that would show that "integer" was used to mean an integer value rather an occurrence of an integer value.
If you were not given an example with an integer occurring twice, you might decide one way or the other without realising that what you decided was not what was intended. Good practice is to make a note of ambiguities, comment on your interpretation in the code, and seek clarification if possible by asking the provider of the specification for what was intended.
3) Write two static methods that both take an array of integers and two separate integers, and multiply all the integers in the array whose value is less than the first separate integer by the value of the second separate integer argument. So, for example, if the array argument is

| 7 | 11 | 13 | 8 | 5 | 10 | 20 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |

and the separate integer arguments are 12 and 5 respectively, the result will be the array:

| 35 | 55 | 13 | 40 | 25 | 50 | 20 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |

The first method should perform this operation destructively, and the second method should perform it constructively.
4) Write a static method that takes an array of integers and an integer (the "target"), finds the position of the integer in the array that is closest to the target, and returns that position. So, if the target is 25 and the array is

the meth | 19 | 9 | 30 | 47 | 5 | 10 | 20 | 36 | 21 | 11 | 13 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | always position 0 , not 1). You may use Java's method abs in class Math to assist.

5) Write a static method that takes an array of integers and two separate integers and returns an array of integers which has the numbers of the portion of the original array starting with the first separate integer argument and ending with the second.
So if the array is:

| 7 | 11 | 13 | 8 | 5 | 8 | 10 | 27 | 30 | 11 | 5 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

and the separate integer arguments are 8 and 27 , the result should be:

| 8 | 5 | 8 | 10 | 27 |
| :--- | :--- | :--- | :--- | :--- |

This example clarifies how to handle a case where the first separate integer argument occurs twice.
6) Write a static method that takes an array of integers and an integer and filters the array so that the resulting array has the elements of the argument array in the same order except that those lower than the argument integer are removed. For example, if the argument array is

| 17 | 11 | 20 | 34 | 5 | 10 | 8 | 19 | 55 | 11 | 13 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

and the integer is 12 , the resulting array should be:

| 17 | 20 | 34 | 19 | 55 | 13 |
| :--- | :--- | :--- | :--- | :--- | :--- |

7) Write a static method that takes an array of integers and returns a boolean, true if the integers in the array are stored in ascending numerical order, false otherwise. So for all the arrays given above, the method would return false. Below is an example of an array where it would return true:

| 3 | 8 | 13 | 16 | 16 | 19 | 27 | 30 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

8) Write a static method that takes two arrays of integers and returns true if and only if all the integers in the second array also occur in the first array. For example, true would be returned if the first array is:

| 17 | 11 | 20 | 34 | 5 | 10 | 8 | 19 | 55 | 11 | 13 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

and the second array is:

| 19 | 20 | 13 | 19 | 55 | 8 |
| :--- | :--- | :--- | :--- | :--- | :--- |

9) Write a static method that takes two arrays of integers and returns the number of integers that occur in both. So, for example, if the arguments are

| 7 | 11 | 40 | 8 | 5 | 10 | 8 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | and | 35 | 10 | 13 | 40 | 7 |
| :--- | :--- | :--- | :--- | :--- |

it should return 3, as three numbers: 7,10 and 40 , occur in both.

