Lab 5 : Arrays and Methods

Topics covered arrays, iteration, methods, prototypes

Code Style Code style is very important to programmers. Programs with good code style are easier to debug, easier to maintain, and have fewer errors. Your tutor will discuss the following parts of code style, which should be applied in your code. Be aware of that 40% of Assignment 1 socre are given based on your code style.

- Naming conventions
- Commenting
- Indentation
- White space

Exercise 1: There are two ways to fill an array with content in Java. The first is to hard-code the values for the array at initialise-time like this:

```java
  int[] values = new int[] {0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20};
```

The second (and much more flexible) way is to assign values to the array programatically. Modify the code above to use a loop to initialise an array of integers, such that the array will have the same values as the array above.

Exercise 2: Write a program that will output the command line arguments in the following format, including the argument index and the quotes around the argument values.

```
$ java PrintArguments first second 45.9 "last argument"
args[0] = "first"
args[1] = "second"
args[2] = "45.9"
args[3] = "last argument"
```

Exercise 3: Write a program that will input a series of integers as command-line arguments, and convert those into an integer array, then print out all elements in the integer array. Remember, the *command-line arguments* are stored into the `args` variable which is an array of `Strings` (not array of integers).

What happens when you try to print out an array? Why would this happen?

```java
double[] values = new double[] {3.2, -1.6, 9.0};
System.out.println("My values: " + values);
// outputs "My values: [D@41a80e5a"
```

If you want to print out the contents of the array, you should iterate over each item in the array and print that out. Alternatively, you can print out an array in a readable way by using the `Arrays.toString(...)` method from the `Arrays` API. For example:

```java
double[] values = new double[] {3.2, -1.6, 9.0};
String readable = Arrays.toString(values);
System.out.println(readable);
```

To use the `toString()` method of `Arrays`, you first need to reference the `Arrays` library with `import` statement before your class declaration.

```java
import java.util.Arrays;
```

Exercise 4: Modify the program you created in Exercise 3 so that once the program has started, it reads integers from keyboard input and outputs whether or not the number that has been input is contained in the original command-line numbers. Make sure your program has an appropriate stopping condition, for example when the user enters ”exit” that the program stops.

```
$ java NumberSearch 7 3 15 9 6
Enter a number:
5
That number is not in the array.
9
That number is in the array.
exit
Exiting program.
```

Exercise 5: Write a class called SortArray to `sort` a given array, and then print the values out. You will need to use the `Arrays.sort()` method to sort an array. Again, to use this method, you have to reference the `Arrays` library. You can check this method from the `Arrays` API for details.

Exercise 6: What do you know about the method that has the following prototype?

```java
public static double dostuff(int a, int b)
```

Is there anything better or worse about this prototype for the same method?

```java
public static double calculatePercentage(int score, int total)
```

Exercise 7: Implement the `calculatePercentage` method in a class called `Scores`. Then write a program takes two command-line arguments and makes use of the `calculatePercentage` method to calculate the percentage score of the input numbers, where the first number is the score and the second number is the maximum possible score.

Exercise 8: Write a method prototype for each of the following methods:

a) A method that accepts two integers and returns the sum of the numbers

b) A method that accepts three words and returns all words together, each separated by a space

c) A method that accepts any decimal number and returns the next highest integer

d) A method that accepts an array of real numbers and returns the minimum of all the numbers

e) A method that accepts a word and will return the same word with all the vowels removed

f) A method that *prints* the sum of three numbers

g) A method that prints the sum of three randomly generated numbers

Exercise 9: Write a full method (including implementation) for the following methods. Try using your methods to see if they work as expected. Do you see any problems with any of the methods? Can you suggest improvements to them?

a) A method that will accept one number. If the number is odd, the number will be set to zero, otherwise the number is left alone.

b) A method that will calculate the square root of an integer, print the square root and then return the square root. If the number is negative, an error message is printed and the number is returned.

c) A method that accepts an array of numbers and sets the odd numbers to zero. Note: this method is **destructive**; i.e. it will modify the original array.

d) A method that accepts an array of numbers and returns only the even numbers. Note: this method is **non-destructive**; i.e. it will not modify the original array.

Extensions

Extension 1: Write a `CaesarCypher`: a program that can rotate the letters in a string around the alphabet by n places, where n is provided. For example, given the string “hello” and $n = 1$, the cypher should rotate ‘h’ to ‘i’, ‘e’ to ‘f’, ‘l’ to ‘m’ and ‘o’ to ‘p’. The letters should wrap around, so rotating ‘z’ should move it back to the beginning of the alphabet again.

Preserve the case! The letters should stay in the same case.

Useful to know to answer this one:

```
char ch = 'a';
Character.isLowerCase(ch); // true iff ch is lower case
```

Also it’s useful to know that the characters are stored in order, so the “value” of ‘a’ is one less than the “value” of ‘b’, etc.