Tutorial 4: Arrays & Debugging

Aim:

- Get familiar with Java arrays and debugging.
- Consolidate learning from week 3, 4.
- Get feedback.

Note: Use the material from the lecture if you need. If you find any problem or have a question, ask your tutor. If you do not have enough time during the session, it is recommended that you finish the exercises at home. Challenges are optional, although recommended.

Exercise 1: Arrays.

Write a program that creates an array of integers with dimension 10 (10 positions). Ask the user to introduce 10 marks and store them in the array. Then, display the 10 values in the array and calculate the number of students who have failed (mark lower than 40) and the average mark. Print the number of students who have failed and the average.

Exercise 2: Arrays: swap values.

Write a program where an array of length 10 is filled with doubles. The user then inputs 2 integer values i and j and the values in the array position i and j are swapped around. The program should then print out the contents of the array. What happens if i or j are larger than 10?

Exercise 3: Set.

Given the set (array) A = {2, 3, 4, 5, 6, 7, 8, 9} find all elements of set A that are:

a) Even numbers
b) Those numbers that being multiplied by 2 give a number that is also an element of A.

Write a Java program that calculates and prints the solution of the previous two tasks.

Note: To find if a number is even or odd, you can calculate the remainder of a division by using the operator %. An even number is a number that is divisible by 2 and generates a reminder of 0.

Acknowledgement: This exercise was taken from Mathematics in Computing (4COSC002W).

Exercise 4: Debugging: Find the bugs/errors.

We have the following Java program:

```
public class Tutorial3_ex4 {      public
static void main(String[] args) {
for(int i = 9; i > 0; i--)         {
for(int j = 9; j > 0; j--)             {
```

```
                    System.out.print(j);
                }
                System.out.println();
            }
        }
    }
```

That prints the following output:

```
987654321
987654321
987654321
987654321
987654321
987654321
987654321
987654321
987654321
```

However, this output is not correct and instead, should print:

```
999999999
888888888
777777777
666666666
555555555
444444444
333333333
222222222
111111111
```

The code has a bug and therefore the output is not the expected. Use the rubber duck debugging method to find the bug. (Refer lecture notes)

Exercise 5: Multi-dimensional arrays.
Write a Java program that asks 3 marks (integers) of 5 students (each student will have 3 marks). Use a multi-dimensional array to store the marks and find the average of each student.

Exercise 06: Probability.
A fair die is rolled twice and we get two numbers X = result of the first roll and Y = result of the second roll.

a) What is the probability that X = 4?

b) What is the probability that Y = 4?
c) What is the probability that both X = 4 and Y = 4?

Write a program in Java that simulates this scenario 1,000 times (you roll two dice 1,000 times) and estimates the probabilities. How good is your simulation? Is it any better if you increase the number of simulations to 1,000,000?

Note: You can use the Java class Random to generate random numbers (you will need to import java.util.Random. You can then create a generator: Random generator = new Random();

Acknowledgement: This exercise was taken from Mathematics in Computing (4COSC002W).

Self-study Questions (Attempt on your own and raise questions with tutor)

1. Write a program that reads in and stores 10 student marks and then reduces each mark by 10% (because it was 24 hours late). Use a loop to reduce each mark. Each reduced mark is then stored in the same element of the array. You should then output the contents of the array.

2. Write a program that reads in 100 *doubles* and stores them in an array called nums. The program should then print out the largest number.

3. Write a program that reads in a sequence of ints into an array. The program should then shift each value on one position in the array. For example, whatever is in array position 4 should be moved into position 5. Whatever is in the last should be moved to the start. The program should then print out the contents of the array.

4. Write a program where an array of length 20 say is filled with doubles. The user then inputs any two integer values i and j and the values in the array positions i and j are swapped around. The program should then print out the contents of the array.

5. Extend your answer above. You need to write a method (function) called swap(). swap() accepts two integer parameters and swaps the contents of an array and those locations. This method does *not* return a value. Implement this method into a suitable java program by demonstrating how the method changes the contents of the array.

6. Write a program to get 10 Integers from the user and store them in an array called myNamesArray. then clone the Integers in the myNames into an ArrayList called myNamesArrayList. Sort the myNamesArrayList using the sort method.

7. Are arrays passed by reference or passed by value? Justify your answer with an example code snippet

8. How do you find all pairs whose sum is equal to a given number from an integer array in Java? You have given an array of int primitives and a number, you need to find all pairs in an array whose sum is equal to given number e.g. if an array is {1, 2, 3, 4, 5} and given sum is 6 then your program should return {2, 4} and {1, 5}.

9. Write a method removeBadPairs that accepts an ArrayList of integers and removes any adjacent pair of integers in the list if the left element of the pair is larger than the right element of the pair. Every pair's left element is an even-numbered index in the list, and every pair's right element is an odd index in the list.

   For example, suppose a variable called list stores the following element values: [3, 7, 9, 2, 5, 5, 8,
   5, 6, 3, 4, 7, 3, 1] We can think of this list as a sequence of pairs: (3, 7), (9, 2), (5, 5), (8, 5), (6, 3), (4, 7), (3, 1). The pairs (9, 2), (8, 5), (6, 3), and (3, 1) are "bad" because the left element is larger than the right one, so these pairs should be removed. So the call of removeBadPairs(list); would change the list to store the following element values: [3, 7, 5, 5, 4, 7] If the list has an odd length, the last element is not part of a pair and is also considered "bad;" it should therefore be removed by your method. If an empty list is passed in, the list should still be empty at the end of the call. You may assume that the list passed is not null. You may not use any other arrays, lists, or other data structures to help you solve this problem, though you can create as many simple variables as you like.

HackerRank: Interview questions. Solve the following tasks in HackerRank:

1. Java 1D Array
2. Java 2D Array
3. Java Arraylist
4. Java 1D Array (Part 2) [more challenging]