COMP2026 Problem Solving Using Object Oriented Programming

Laboratory 5

Part A Discovery Exercises

Task 1: More about Arrays

Given the following arrays.

```
int[] array1 = {1, 2, 3};
int[] array2 = {4, 5, 6};
int[] array3 = {7, 8, 9};
```

a) What are the values inside the arrays after running the following code fragment?

```
array1[0] = 10;
array2[1] = 20;
array3[2] = 30;
```

Answer:

b) What are the values inside the arrays after running the following code fragment?

```
array3 = array1;
array1[0] = 40;
array2[1] = 50;
array3[2] = 60;
```

Answer:

c) Given the following arrays.

```
int[] array1 = {1, 2, 3};
int[] array2 = {4, 5, 6};
int[] array3 = {7, 8, 9};
```

What are the values array3[array1[0]] and array3[array1[1]]?

Answer:

Task 2: Two-Dimensional Arrays

Given a two-dimensional integer array **y**.

a) Write statements to print the size of **y**. You have to print the number of rows and the number of columns in each row.

b) Write statements to print all the elements of a 2D array **y** in table format.
For example, if **y** is {{1, 2, 3}, {4,5,6}}, then the result will be printed as follows:

```
1 2 3
4 5 6
```

c) Write statements to compute and print the average value of all the elements in an integer 2D array **a**.

Part B Programming Exercises

Task 1: Merging Arrays

Given ***sorted*** arrays, **`a`** and **`b`**, create a sorted array that contains the values sorted in both **`a`** and **`b`**. In other words, your program should merge arrays **`a`** and **`b`** into a third array. You can merge the two sorted arrays as follows:

Declare a new array **`c`** that is large enough to hold the contents of both **`a`** and **`b`**; also declare two integer variables, **`i`** and **`j`**. Initialize **`i`** and **`j`** to 0.

i. Compare **`a[i]`** and **`b[j]`** and copy the smaller value into **`c`**.
ii. Increment **`i`** if **`a`** contains the smaller value, otherwise increment **`j`**.
iii. Repeat this procedure until either **`i`** or **`j`** exceeds, the highest index of **`a`** or **`b`**, respectively.
iv. Copy the remainder of either **`a`** or **`b`** to **`c`**.
v. Print **`c`**

Complete the given **`Merging.java`**.

Task 2: Flood Map

You are given a two-dimensional array of integer values that give the height of a terrain at different points.

a) Complete the given **`MapFlooding.java`** to print out a flood map, showing which of the points in the terrain would be flooded if the water level is given. In the flood map, print a * for each flooded point and a space for each point that is not flooded.

Sample outputs:

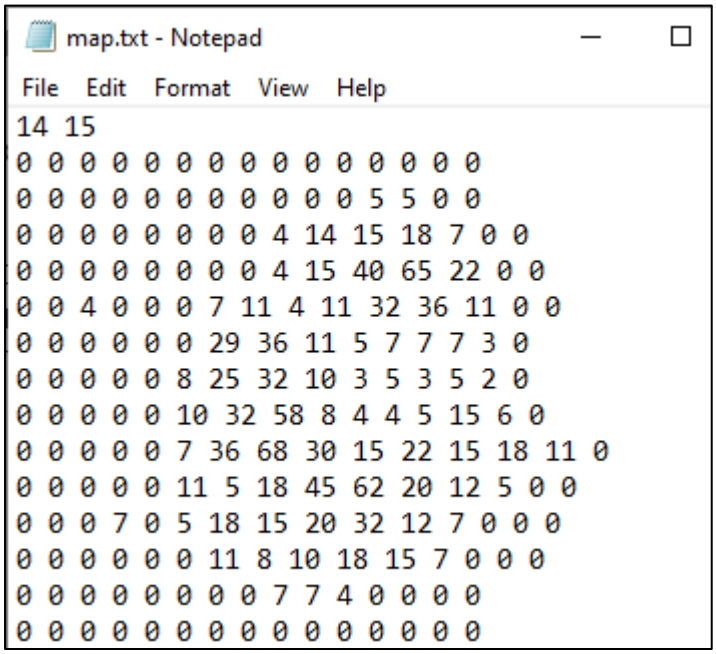
```
Enter the water level: 5
***************
***************
*********    **
*********    **
******  *    **
******   *   **
*****    ******
*****    ***  *
*****         *
***** *     ***
*** **      ***
******      ***
********  *****
***************
```

```
Enter the water level: 30
***************
***************
***************
**********  ***
**********  ***
******* *******
******* *******
******  *******
******  *******
********  *****
********* *****
***************
***************
***************
```

b) Modify the program in (a) to read the heights of a terrain from a text file.

File format:

- Each text file contains one map.
- The first line of the text file contains two integers, and they are the number of rows and the number of columns in the map.
- The heights of the terrain start from the second line.

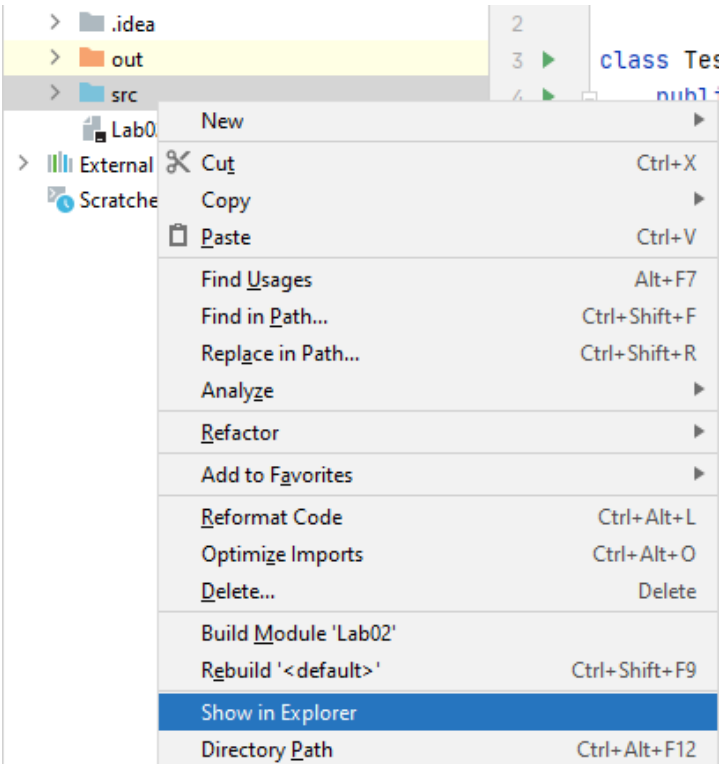


Sample output:

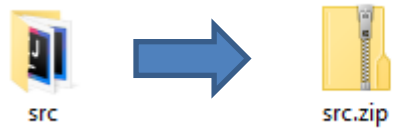
```
Enter the filename: map.txt
Enter the water level: 10
***************
***************
*********   ***
*********    **
******* *    **
******   ******
******  *******
******  **** **
******        *
***** *     ***
******     ****
****** **  ****
***************
***************
```

Part C Submitting Exercises

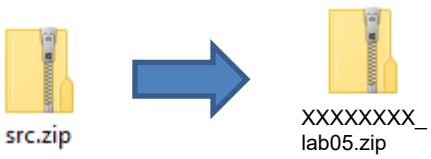
Step 1: Right-click the **src** folder and select **Show in Explorer**



Step 2: Zip the **src** folder into **src.zip**



Step 3: Rename the src.zip file to ***XXXXXXXX_lab05.zip*** where ***XXXXXXXX*** is your **student id**



Step 4: Submit ***XXXXXXXX_lab05*.zip** and **XXXXXXXX_lab05.docx** to Moodle.

References

[1] Bravaco, R., & Simonson, C. (2009). *Java programming: From the ground up*. Dubuque, IA: McGraw-Hill.

[2] Dean, J., & Dean, R. (2008). *Introduction to programming with Java: A problem solving approach*. Boston: McGraw-Hill.

[3] Farrell, J. (2012). *Java programming. Boston, MA: Course Technology Cengage Learning*

[4] Forouzan, B. A., & Gilberg, R. F. (2007). *Computer science: A structured programming approach using C (3rd ed.)*. Boston, MA: Thomson Course Technology.

[5] Gaddis, T. (2016). *Starting out with Java (6th ed.)*. Pearson.

[6] Liang, Y. D. (2013). *Introduction to Java programming: Comprehensive version*. (8th ed.). Pearson.

[7] Schildt, H. (2006). *Java a beginner's guide*. New York: McGraw Hill.

[8] Schildt, H., & Skrien, D. J. (2013). Java programming: A comprehensive introduction. New York: McGraw-Hill.

[9] Wu, C. T. (2010). *An introduction to object-oriented programming with Java*. Boston: McGraw Hill Higher Education

[10] Xavier, C. (2011). *Java programming: A practical approach*. New Delhi: Tata McGraw Hill.

[11] yet another insignificant Programming Notes. (n.d.). Retrieved from https://www3.ntu.edu.sg/home/ehchua/programming

[12] Zakhour, S., Kannan, S., & Gallardo, R. (2013). *The Java tutorial: A short course on the basics (5th ed.)*.