4COSC010C Software Development II

Files and testing

Aim:

- Get familiar with files handling
- Get feedback.

Exercise 1: File creation and writing.

In this task, you will create a file, write in the file created, and read from the file.

1.- Create a new project and add the following imports for file handling (before your public class declaration):

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
```

2.- Create a new file named Exercise1.txt using the following code in your `main`:

```
File file = new File("Exercise1.txt");
try {
     boolean file_created =file.createNewFile();
     }
     catch (IOException ex) {
     System.out.println("ERROR: File could not be created.");
}
```

Remember that file handling needs to be inside a try/catch statement.

4.- Check if the file has been created and write “This is my first file.” in the file, and close the file:

```
if (file_created) {
     FileWriter writer = new FileWriter("Exercise1.txt");
     writer.write("This is my first file.");
     writer.close();
}
```

Your code should look like this:

```
import java.io.File; import
java.io.FileWriter; import
java.io.IOException; public
class Week8_ex1 {
 public static void main(String[] args) {
File file = new File("Exercise1.txt");
try {
      boolean file_created = file.createNewFile();
      if (file_created) {
            FileWriter writer = new FileWriter("Exercise1.txt");
            writer.write("This is my first file.");
            writer.close();
      }
  } catch (IOException ex) {
    System.out.println("ERROR: File could not be created.");
  }
 }
}
```

5.- Run your code and open the file Exercise1.txt (use the tab Files or the Explorer), and **make sure the file contains the text "This is my first file")**.

Exercise 2: File reading.
In this task, you will read from the file created in exercise 1.

1.- Create a new project and add the following imports:

```
import java.io.File; import
java.io.IOException; import
java.util.Scanner;
```

2.- Create a File and a Scanner that reads from the file (remember to import `java.util.Scanner` first):

```
File file = new File("../week8_ex1/Exercise1.txt");
try {
   Scanner file_reader = new Scanner(file);
}
catch (IOException ex) {
  System.out.println("ERROR: File could not be created.");
}
```

Make sure that the file path is correct when creating the object File. Change `week8_ex1` in the path name for your project's name (exercise 1).

2.- Read the file and print its contents:

```
while (file_reader.hasNextLine()) {
     String text = file_reader.nextLine();
     System.out.println(text);
}
```

When running the program, you should see the following text in the output console:

```
This is my first file.
```

Your code should look like this:

```
import java.io.File;
import java.io.IOException;
import java.util.Scanner;
 public class Week8_ex2 {
     public static void main(String[] args) {
         File file = new File("../week8_ex1/Exercise1.txt");
         try {
                  Scanner file_reader = new Scanner(file);
                  while (file_reader.hasNextLine()) {
                        String text = file_reader.nextLine();
                        System.out.println(text);
            }
        } catch (IOException ex) {
            System.out.println("ERROR: File could not be opened.");
      }
    }
}
```

Exercise 3: Save the array to a file.
Given the following array:

```
double[] array = {1.3, 5.6, 2.3, 9.0};
```

Write the contents of the array in a file using a loop. Ensure the values of the array have been stored correctly in the file (open the file to check the output is correct). What happens when you write in a file the array directly? Does it work? Or do you need to store each item individually?

Exercise 4: Read array from file.
In a new project, create a method that given a String with a filename and path, reads an array and returns it.

In your main, use the method to read the file from Exercise 3, and print the values in the array.

Note: You can use `Scanner` with `nextDouble` to read doubles from a file. The method `hasNextDouble()` from Scanner will return false when there are no more doubles in the file.

Exercise 5: Testing.

A software engineer has asked ChatGPT to create a Java program that finds the items in an array that are lower than 50 and ChatGPT has generated the following code:

Sure, here's a simple Java program that finds and prints the items in an array that are lower than 50:

```
public class Main {
   public static void main(String[] args) {
     int[] array = {10, 60, 30, 40, 80, 20, 90}; // Example array
      System.out.println("Items in the array lower than 50:");
     // Loop through the array
     for (int i = 0; i < array.length; i++) {
       // Check if the current element is lower than 50
if (array[i] < 50) {
         System.out.println(array[i]);
      }
    }
  }
}
```

Given the above Java code, design a test plan using black-box and white-box testing.

Test case / scenario	Input	Expected Output	Output	Pass/Fail

Coursework surgery
If you need, you can also work on your coursework during this tutorial. Ask your tutor any questions you may have regarding the coursework.

Section 02
Read a multi-dimensional array.
Create a .txt file with the following content (make sure the file is in the main project directory, or you will need to specify the whole path):

```
14 38 43 64 48 45 74
38 49 74 96 73 07 73 28
93 40 75 43 85 65
38 07 14 68 53 57 10
```

Read the file and put the numbers in a multi-dimensional array. Then, find the positive numbers that appear multiple times in the multi-dimensional array (highlighted in bold).

Section 03
HackerRank: Interview questions.
There are no exercises with Files and Testing, but you can solve the following tasks in HackerRank:

1. Java Exception Handling (Try-catch)
2. Java Exception Handling