Lab Session	**#11**

Learning Outcome:

1. **Students could create java program using exceptions**

To Do List (during the lab):

1. **Completing online course:** https://runestone.academy/ns/books/published/javajavajava/handling-exceptional-conditions.html?mode=browsing
2. **Completing the tasks below:**

All Cohorts:

1.Guided
You are tasked to copy a file named "input.txt" inside a folder named "exceptions". Inside this folder there are another folder named "output" which stores all the output from your program. Create a program to copy the file and make another file inside the output folder.

You have been given the task of copying a file named "input.txt" located in a folder called "exceptions". Inside this "exceptions" folder, there is another folder named "output" where you need to store the output from your program. Your goal is to create a program that copies the "input.txt" file and creates another file inside the "output" folder.

To accomplish this task, you can follow these steps:

1. Identify the location of the "input.txt" file in the "exceptions" folder.
2. Create a new file in the "output" folder to store the copied contents.
3. Read the contents of the "input.txt" file.
4. Write the contents to the newly created file in the "output" folder.
5. Close the files properly to ensure the copied contents are saved.
6. Print a success message to indicate that the copying process has been completed.

In the program that you will make there are exceptions you will need to be aware of.
By following these steps, you will be able to copy the "input.txt" file and create another file in the "output" folder as per the given task.

2. Un-Guided

You are tasked with writing a Java program that performs a division operation and handles arithmetic exceptions.

- The program starts by prompting the user to enter the numerator and denominator values.
- The user is then expected to provide integer inputs for both values.
- Inside the program, a division operation is performed by dividing the numerator by the denominator.
- To handle potential arithmetic exceptions, the division operation is placed inside a try block.
- If a division by zero occurs (an ArithmeticException), the program catches the exception.
- Upon catching the exception, the program outputs the message "Tidak dapat membagi dengan angka 0!".
- If no exception occurs during the division, the program calculates the result and outputs it to the console, preceded by the message "The result is: ".
- The program terminates.

This program ensures that division by zero is not allowed and provides appropriate feedback to the user in case of an arithmetic exception.

Submit your work through codeboard:

1. **Cohort A:**
2. **Cohort B:** https://codeboard.io/projects/390940
3. **Cohort C:**
4. **Cohort D:**
5. **Cohort E:**
6. **Cohort F:**