CS 250 - Fall 2021 - Lab 03: Exceptions

Available: Sep 8, 2021.
Due date: **Sep 13, 2021, 11:59 PM**

Objectives:

The objectives of this lab are:

- To implement try/catch statements.
- To develop and test appropriate design strategies for handling Exceptions.
- To demonstrate the inefficiency of using exception handling as a means of normal program control.

Prerequisite:

- Read Chapter 10 of *Java, Java, Java, 3E* and the supplemented slide of lecture 03
- Read online tutorials of Exception Handling on the javatpoint website.

Part 1: Handle division by zero exception

Step 1. Run the Eclipse IDE program

NOTE: Remember how to find this program because you're going to need it every week.

Step 2. Create the Lab03 project.

- Click on the "Create a java project" link in the Package Explorer window or right-click on that window then select NEW then Java project. Enter the name of your project, i.e. "Lab03". The "Location" text box should indicate that your project will be stored on the desktop. In the project JRE setting, make sure at least one Java SE version is selected, default to JavaSE-1.8 if you followed instructions on the Prerequisites section. Click on Finish to create the project.

- In the Package Explorer window, right-click on the "src" item, then select "New" and then Class. Next, create a class named “P3a”.

Step 3. Implement the following activities:

- Declare a Scanner instance

- Prompt a text “Enter the numerator (a): “ and use the Scanner object to read an integer into an integer variable named numerator

- Prompt a text “Enter the numerator (a): “ and use the Scanner object to read an integer into an integer variable named denominator

- Calculate the ratio between the numerator and the denominator: ratio = numerator / denominator

- Print to the console a text with the format: "numerator / denominator = ratio” (replace numerator, denominator, and ratio with their values)

Test the above program and notice the output if the user enters 0 as the denominator. Now, we practice exception handler by putting a try-catch statement to wrap the code of calculating the ratio. Rerun the application to make sure the exception situation is handled.

Part 2: Read a sequence number from user’s input line and calculate the sum

In this part, we will develop a simple Java program that reads a sequence of positive integers from Console and calculate the sum of all numbers in the sequence. Create a new class named P3b and complete the following steps:

- Declare a Scanner object to read the user’s input from Console
- Prompt a text “Enter the length of the sequence: “
- Use the Scanner object to read an integer and assign it to an integer variable named len
- Declare an array of integers named arr to store the sequence; initialize the length of arr with the variable len
- Use a for loop with an index i from 0 to less than len:
    - Using the next() method of the Scanner object to read the next token from Console --- assign to a string variable str
    - Use Integer.parseInt to parse str into integer and assign it to arr[i]
- Define a sum variable with an initial value of 0 (This variable will store the total value of all numbers in the sequence)
- Use another for loop to add each number in the array arr to sum
- Print the sum variable to the Console

Now, execute the program with two sequences:

- 10 20 30 40 50
- 1.0 2.0 3.0 4.0 5.0

For the first sequence, we expect the program to run smoothly, and the final sum of 150 is printed. However, for the second sequence, there will be an exception as Java could not parse floating-point numbers to integers. Similarly, if the users enter any non-number input, the second program also causes exceptions to happen.

To fix the exception, place a try-catch statement surrounding the Integer.parseInt statement in the above code to catch the error. Also, display an error message inside the catch statement to warn the user about the incorrect format of inputs.

Lab Assignment

Examine the following code:

```
public class L3A {
	public static void main(String[] args) {
		int num1 = 22, num2 = 0;
		int divResult=0;
		char[] chArr = { 'a', 'b', 'c', 'd' };
		for (int i = 0; i < 2; i++) {
			if (i == 0)
				divResult = num1 / num2;
			else
				chArr[4] = 'X';
		}
		System.out.println("divResult=" + divResult);
		System.out.println("chArr=" + String.valueOf(chArr));
	}
}
```

Complete the following tasks:

a. (In Word or PDF file) Identify all lines of code of the L3A class above that have errors (either syntax or semantic errors) and can generate exceptions when running. Explain the reasons for such errors.

b. (Source code) Modify the program with exception handling (add try-catch block) to fix the code of L3A.

Lab Result Submission:

You need to submit the results of the following sections to the D2L assignment item of Lab#3:

- Complete the required Lab Assignment above

- Compress the whole Eclipse source code of your Lab#3 project in zip format using the Export function