PA 8: Exception Handling in JAVA

Name: Humesh Reddy Venkatapuram

Date: 04/26/2023

**Problem Description:** The task is to convert the grade distribution program written in ADA language to JAVA language. In the converted program, we need to modify the second half of the first loop in such a way that all updates to the "Freq()" counting array should only occur in the Exception part of the code. There should be no updates to "Freq()" anywhere else in the loop.

**Code in JAVA:**

import java.util.\*;

class Assignment\_8 {

public static void main (String[] args) {

// Creating a scanner object

Scanner s = new Scanner (System.in);

// Declared the required variable and array here.

int[] freq = new int[11];

int limit\_1, limit\_2, index;

int new\_grade;

/\* Getting into the loop.

\* The loop will run unlimited times unless there is a negative number in the

input.

\*/

for ( ;; ) {

// Scanning input

new\_grade = s.nextInt();

try {

if (new\_grade < 0) {

throw new ArithmeticException();

}

} catch (ArithmeticException e) {

  break;

}

// Scaling index

index = new\_grade/10 + 1;

// Taking every input grade as an exception.

try {

if (new\_grade >= 0) {

throw new ArithmeticException();

}

}

// As part of the logic the freq[] updating is done in the catch section.

catch (ArithmeticException e) {

if (new\_grade < 100)

freq[index] = freq[index] + 1;

if (new\_grade == 100)

freq[10] = freq[10] + 1;

if (new\_grade > 100)

System.out.println("Error -- new grade: " + new\_grade + " is out of range");

}

}

System.out.println("Limits Frequency");

System.out.println("");

// Printing out the values.

// We can reach 90-100 index.

for ( index=0; index<10; index++ ) {

limit\_1 = 10 \* index;

limit\_2 = limit\_1 + 9;

if (index == 9)

limit\_2 = 100;

System.out.print(limit\_1 + " ");

System.out.print(limit\_2 + "  ");

System.out.print(freq[index+1]);

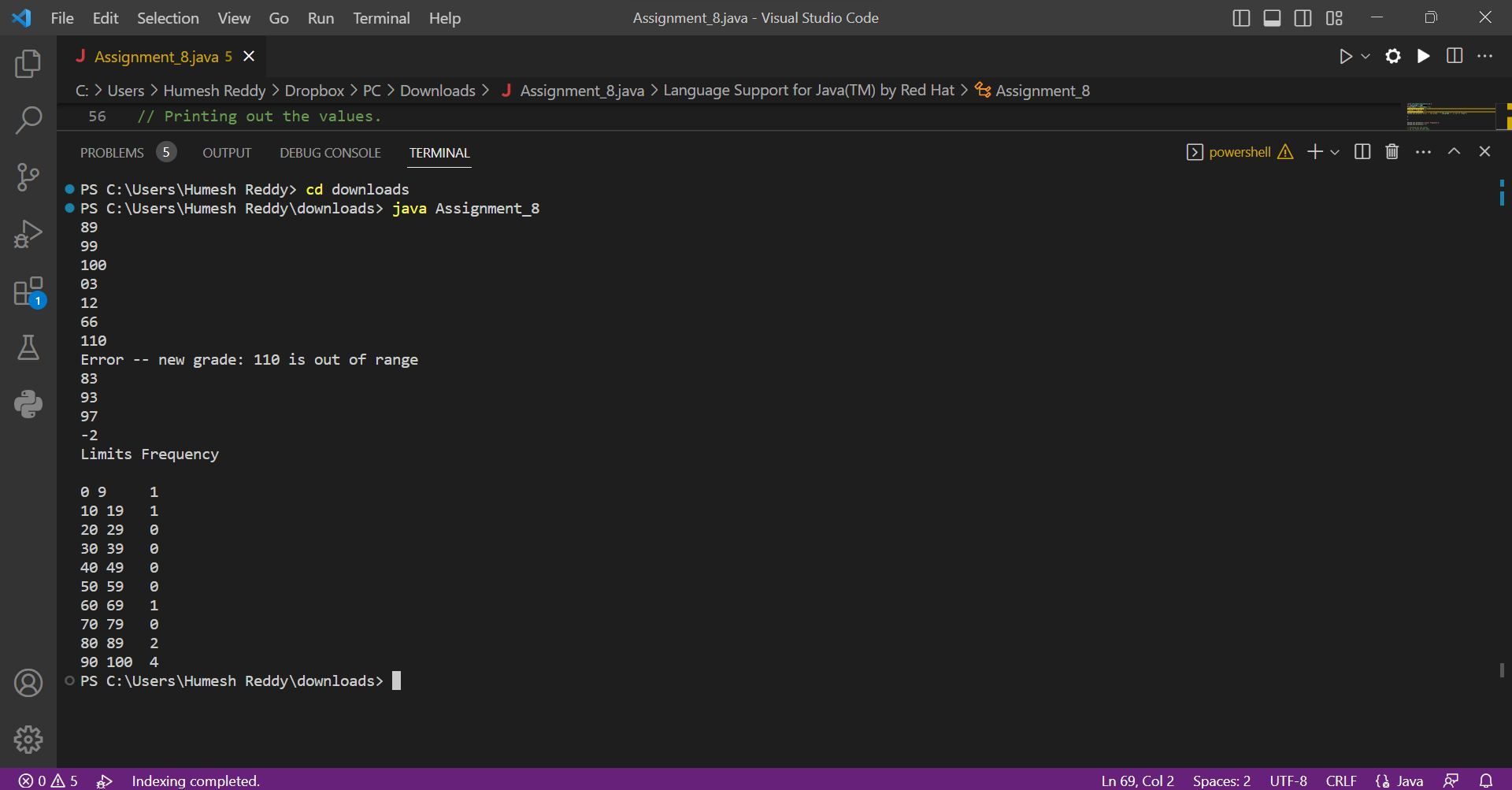
System.out.println("");

}

}

}

**Output:**

****

The program is designed to accept any number of positive numbers as input and it will continue to do so until a negative number is entered, which signals the end of the input sequence.

If a number above 100 is entered, the program will generate an error message indicating that the input is out of range. Therefore, it is recommended to input numbers within the specified range.