

CS 159 Exception Exercises

Names: _____

1. This question refers to the `Divider` class provided on an attached page.

The `Divider` class represents an unfinished effort to write a command line integer division program. When completed, this program will read two integers from the terminal, divide the first by the second, then print the result to the terminal. The author has included several print statements throughout the program to help her debug the exception handling code.

What will be printed when each of the following are entered as command line arguments?

(a) 3 2

(b) 4 0

(c) -4 2

(d) 4

(e) a b

2. Create a new exception class named `InvalidStringException`. This should be a checked exception.

3. Complete the following method so that it conforms to the Javadocs.

```
/**
 * Count the total number of characters in all Strings in the provided array
 * of Strings. All strings must have length of at least one.
 *
 * @param strings - The array of strings.
 * @return The character count.
 * @throws InvalidStringException - Thrown if any string has length 0.
 * @throws NullPointerException - Thrown if there are any null strings in
 *         the array.
 */
public static int countCharacters(String[] strings)
```

```

public class NegativeResultException extends RuntimeException
{
    public NegativeResultException(String message)
    {
        super(message);
    }
}

```

```

public class Divider
{
    public static void main(String[] args)
    {
        try
        {
            int result = divideStrings(args[0], args[1]);
            System.out.println(result); // <----- PRINTLINE HERE!
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println("No good!"); // <----- PRINTLINE HERE!
        }
        catch (RuntimeException e)
        {
            System.out.println("Please try again."); // <----- PRINTLINE HERE!
        }
        finally
        {
            System.out.println("Thank you."); // <----- PRINTLINE HERE!
        }
    }

    public static int divideStrings(String numerator, String denominator)
    {
        System.out.println("A"); // <----- PRINTLINE HERE!
        int numInt, denomInt;
        int result = 0;

        try
        {
            numInt = Integer.parseInt(numerator);
            denomInt = Integer.parseInt(denominator);
            result = divideInts(numInt, denomInt);
        }
        catch (NumberFormatException e)
        {
            System.out.println("B"); // <----- PRINTLINE HERE!
        }
        catch (ArithmeticException e)
        {
            System.out.println("C"); // <----- PRINTLINE HERE!
        }
        System.out.println("D"); // <----- PRINTLINE HERE!

        return result;
    }

    public static int divideInts(int numerator, int denominator)
    {
        System.out.println("E"); // <----- PRINTLINE HERE!
        int result;
        result = numerator / denominator;

        if (result < 0)
        {
            throw new NegativeResultException("Division has negative result.");
        }

        System.out.println("F"); // <----- PRINTLINE HERE!
        return result;
    }
}

```