Vraj Desai

JH1 worksheet

I think We just opted out Class Participation So may be I don't need to fill Class Participation Part?

********** Class Participation **********

Your Class Participation Entry for JH1:

Discussion board containing your entry: _____

Date of your Entry: _1/19/2018_____

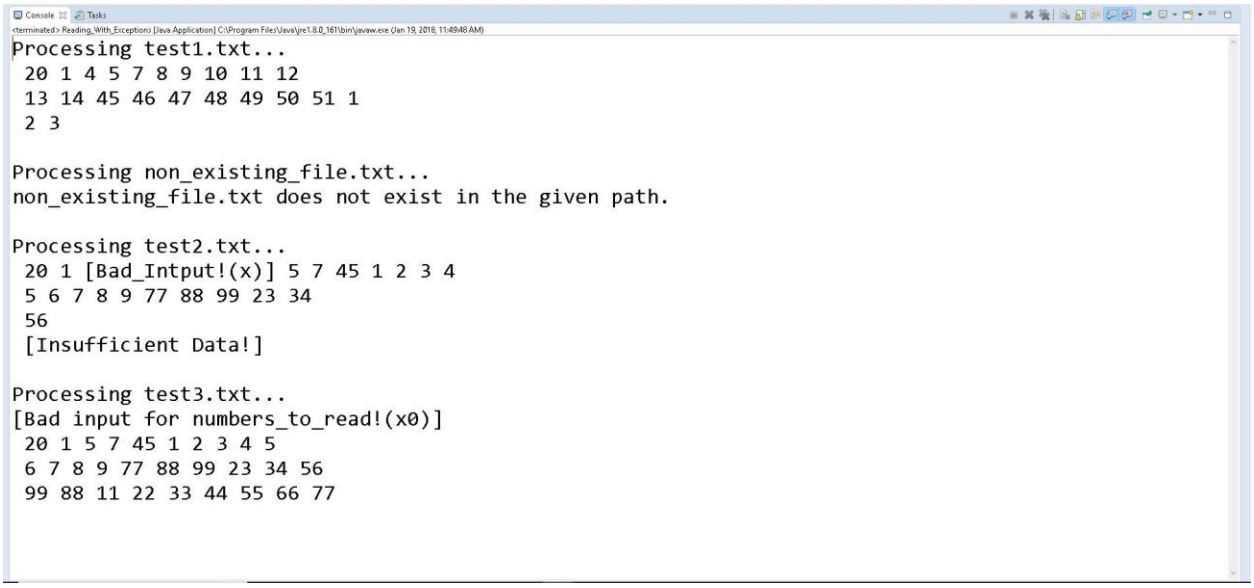
Subject Line of your entry: _Reading_With_Exceptions_____

I am going to add my console output screen and my code after that.

********** reading_with_exceptions **********

reading_with_exceptions output from the screen when the command line parameters are:

file1.txt<U+FFFD>non-existent-file file2.txt file3.txt



Reading_With_Exceptions.java:

```
import java.io.FileNotFoundException;
import java.io.*;
import java.util.*;

public class Reading_With_Exceptions {

	static void process(String inputFilename) throws FileNotFoundException {
		FileInputStream in_File = null;
		Scanner reader = null;
		PrintStream out = null;
		PrintStream err = new PrintStream(new File("Errors.txt"));

		try{
			//variables
			String outputFilename = null;
			int numbers_to_read = -1;

			//Process
			in_File = new FileInputStream(inputFilename);
			reader = new Scanner(in_File);
			if(reader.hasNext())
				outputFilename = reader.next();
			else
				System.out.println("Invalid File Format!");
			out = new PrintStream(new File(outputFilename));
			if(reader.hasNextInt()) {
				numbers_to_read = reader.nextInt();
```

```
                    if(numbers_to_read < 0) {
                        out.println("Bad int : numbers_to_read < 0");
                    }
                }
                else{
                    out.println("[Bad input for numbers_to_read!("+ reader.next() +")]");
                }
                copyNumbers(reader, out, numbers_to_read, err);
                printToScreen(outputFilename);
            }
            catch(FileNotFoundException e) {
                System.out.println( inputFilename + " does not exist in the given path.\n");
            }
            catch(InputMismatchException e) {
                err.println(reader.next());
            }
            finally {
                if(reader != null)
                    reader.close();
                if(out != null)
                    out.close();
                if(err != null)
                    err.close();
            }
    }

    static void copyNumbers(Scanner reader, PrintStream out, int int_num, PrintStream err) {
            int count = 0;
            while(reader.hasNext() && count != int_num) {
```

```
                if(count != 0 && count%10 == 0)
                        out.print(" \n");
                if(reader.hasNextInt())
                        out.print(" " + reader.nextInt());
                else {
                        out.print(" [Bad_Intput!(" + reader.next() + ")]");
                }
                count++;
        }
        out.print("\n");
        if(int_num != -1 && count < int_num) {
                out.println(" [Insufficient Data!]\n");
        }
        else
                out.println();
}

static void printToScreen(String filename){
Scanner scan = null;
        try {
                FileInputStream fis = new FileInputStream(filename);
                scan = new Scanner(fis);
                while (scan.hasNextLine()){
                        System.out.println(scan.nextLine());
                }
        }
        catch (FileNotFoundException e)
        {
                System.out.println("printToScreen: can't open: " + filename);
```

```
            }
            finally
            {
                if (scan != null)
                    scan.close();
            }
        }// end of printToScreen

        public static void main(String[] args) throws FileNotFoundException{
            for(int i = 0; i < args.length; i++) {
                System.out.println("Processing " + args[i] + "...");
                process(args[i]);
            }
        }
}
```