ICSI201/ICEN141 Fall 2022 -- Project 1

`RangeOverlap`

~~Due Friday, October 7th, 2022, at 11:59PM via Blackboard~~

~~**Note**, this is a firm deadline, and no extensions will be given (see syllabus for deadline policies).~~

~~(100 points)~~

ASSIGNMENT OVERVIEW

The goal of this project is to practice the programming knowledge and skills you have acquired so far to implement a program called `RangeOverlap`. The program takes as user input the start and end values for two closed integer ranges. The program then produces the list of numbers contained in the overlap of the two ranges, and the percent overlap of the two ranges. Note that we will only consider integers (as opposed to any real number). Also note that the ranges are closed, that is the start and end limits are included in the ranges.

Objectives: There are several objectives to this assignment.

- Begin to develop skills in independent Java application programming.
- Practice and demonstrate your understanding of important knowledge and skills we acquired so far including:
  - variable declaration, initialization, and use
  - arithmetic operations
  - getting user input
  - employing decision structures to accommodate various input conditions
  - employing loops to traverse a set of values
  - making independent choices about which decision and loop structure to use depending on the task
  - practicing good programming style (i.e. indentation) and documentation (though appropriate comments)
- Independently select and combine programming concepts to implement user functionality

Reading: Chapters 1-4 are the absolute minimum of reading you need to complete to confidently tackle this assignment. You should also be comfortable with the coding skills you acquired in Labs 0-4 and zyLabs #1-#3.

GRADING GUIDELINES:

This assignment is to be completed individually. Any assignments flagged for group work will be considered for plagiarism. See my policy on cheating for more details.

Important NOTE on Formatting and Code Compliance: Your code should use and expand the provided Java class skeleton with this assignment (`RangeOverlap.java`). You should make sure your code compiles in your own IDE before submitting. As a reminder, please submit your .java file. If you submit a .class file, we will not be able to grade your program and you will receive 0 points.

The breakdown of points for this assignment is in the table below.

Grading item	Points
Appropriately declares and initializes variables • Including proper choice of variable types	10
Correctly takes and error-checks user input	10
Correctly computes the values in the overlap of the two ranges ***using a loop structure***	30
Correctly computes the percentage overlap of the two ranges	30
Follows the assignment specification in displaying input and output to the user • Does not modify the provided prompts in the program skeleton • Does not modify the provided .java filename and the class name • Displays the values in the overlap to the user as per specification • Displays the percentage overlap to the user as per specification	10
Follows proper coding style (good indentation and self-documenting variable names)	5
Includes good documentation (appropriate comments)	5

~~**Procrastination Warning:**~~
~~This is your first independent assignment, and you want to start on it early. No matter how confident you feel now, when you get in the details you will have questions and the more time you have to approach your instructors and TAs the more successful you will be. Instead of assuming things should be done a particular way, **ask questions!** Use Piazza to ask clarifying questions (never post your code publicly), discuss your approach with peers and use the instructors' and TAs' office hours to get help if you are stuck.~~

File naming conventions and instructions compliance:
You are provided a code skeleton, which you must use in developing your submission. The skeleton file for this project is available on Blackboard with this assignment and is called `RangeOverlap.java`. We will be auto running your code and if your file and class naming conventions deviate from the skeleton, you will lose points. Additionally, the skeleton provides guidance on what structures should be used to implement different parts of the program. Where indicated, you must use the required structure.

~~**Assignment Turn-in:**~~
~~You must submit your source code, so we can assess your implementation. In other words, you must submit your `.java` file. If you submit a `.class` file, you will receive 0 points. You must submit a single `.java` file named `RangeOverlap.java` using the course Blackboard.~~

~~**Cheating policy:**~~
~~**Cheating is not tolerated.** We will be comparing your code against that of other students in the class and against similar assignments from online student support platforms. All students involved in a cheating accident (i.e. whether sharing or receiving code) will be penalized. Please, read the syllabus for our policies on cheating and refer to slides from Lecture 1 for examples of what we consider cheating, as those are not only limited to what you submit. Students caught cheating will receive 0 points for the assignment and will be reported.~~

ASSIGNMENT DETAILS:

This part of the assignment provides details on how your program should behave. Such a description is often referred to as *program specification* or simply *specification*. Specifications detail both the expected input and output of your program, as well as the target behavior of your program in manipulating user input. Please, follow the program steps as detailed below.

1. Your program should begin by displaying an information message to the user. At this stage, the program should also display the first input prompt. Note that while the code skeleton provides you with the input prompts, you must implement the information message yourselves.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
```

2. Your application should take as an input four integers that comprise the start and end limits of the two number ranges. Below is a snapshot of a program taking user input.

```
Please specify the start limit of your first range
10
Please specify the end limit of your first range
20
Please specify the start limit of your second range
15
Please specify the end limit of your second range
20
```

3. The program should then display a summary of the user input.

```
--------------------------------
You have specified the following ranges:
        Range 1: 10 - 20
        Range 2: 15 - 20
--------------------------------
```

4. The program should then perform an input error check. Specifically, the program should check if the start limit is smaller or equal to the end limit for both ranges. If the user has made a mistake, by specifying a start limit that is larger than the end limit, the program should keep asking the user for input until the user specifies proper ranges. Below is a snapshot of what the error message should look like.

```
--------------------------------
ERROR: Your input is incorrect.
Ensure that the start limit is smaller or equal than the end limit for both ranges.
--------------------------------
```

5. Once correct input is obtained by the user, the program should calculate and display the values in the overlap between the two ranges and the percentage overlap between the two ranges as floating-point numbers with precision to the first digit after the decimal point. Here is an example output to be displayed to the user:

```
The overlap between the two ranges is: 15 16 17 18 19 20.
The percentage overlap between the two ranges is: 54.5%
```

In what follows, we provide more detail on how to compute the value overlap and the percentage overlap.

   a. *Value overlap.* This is the set of values that are shared among the two user-specified ranges. Note that we assume that the ranges are closed (that is, both the start and end limit values are included in the range). In the example above, where the first range is `[10 20]` and the second is `[15 20]`, the value overlap comprises the numbers `15 16 17 18 19 20`. Note that your program should support all possible range overlaps including full overlap, various partial overlaps and no overlap. As pointed in the skeleton, you must use a loop structure to find the value overlap.

   b. *Percentage overlap.* We calculate the percentage overlap as the size of the overlap of the two ranges divided by the size of the union of the two ranges.

      We define *the union of two ranges* as the combination of all values that span the two ranges. Your program should account for disjoint ranges. I.e., if the first range is `[5 8]` and the second range is `[10 12]`, the size of the union is `7` ~~and not `8`.~~

      We already defined the overlap of two ranges above. We need to calculate the *size of the overlap* of the two ranges. Pay, attention, if the ranges are non-overlapping, the overlap size will be negative. To handle such cases, you should re-set the overlap size to 0, to get the expected 0% overlap.

   HINT: You can perform arithmetic operations on the range limits to find the sizes of the overlap and union. Make sure to account for the limit values in closed ranges.

6. Your code should follow a proper coding style and include good documentation.

   a. ~~*Coding style.* Well-written code uses proper indentation that follows the scope of your statements and employs intuitive and self-documenting variable names. Make sure your code uses correct indentation. Additionally, ensure that you are using descriptive variable names. For example, if you are creating a variable to store the annual salary of a worker, a self-documenting way to name the variable might be `annualSalary` or `salary`.~~ A poor naming choice, in turn, would be to use `x` or `zz` as a variable name.

   b. *Documentation.* To gain points for good documentation, your code should include comments that detail the purpose of blocks of code or individual statements. Good

code/documentation does not imply that more is better. The goal is to be efficient, elegant, and succinct! Here are some examples of bad comments:

```
double annualSalary; // declares annualSalary
public static void main (String args[]) // Method header
```

Here are some examples of good/useful comments:

```
/*
 * This method takes input from the user about
 * the number of miles they can currently run
 */
private static int inputMiles()
{
    ....
```

An important part of your documentation is the identity of the developer who wrote the code. Please, add a comment at the very beginning of your program. Here is an example from your instructor:

Appendix

Below we provide a few snapshots of full runs that explore possible cases of range overlap and erroneous input. Please, compare your program's output to the examples below to ensure that it produces both the same prompts and results.

Example 1: Sample output with partially overlapping ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
10
Please specify the end limit of your first range
20
Please specify the start limit of your second range
15
Please specify the end limit of your second range
20
--------------------------------
You have specified the following ranges:
        Range 1: 10 - 20
        Range 2: 15 - 20
--------------------------------
The overlap between the two ranges is: 15 16 17 18 19 20.
The percentage overlap between the two ranges is: 54.5%
```

Example 2: Another sample output with partially overlapping ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
11
Please specify the end limit of your first range
14
Please specify the start limit of your second range
10
Please specify the end limit of your second range
29
--------------------------------
You have specified the following ranges:
        Range 1: 11 - 14
        Range 2: 10 - 29
--------------------------------
The overlap between the two ranges is: 11 12 13 14.
The percentage overlap between the two ranges is: 20.0%
```

Example 3: Another sample output with partially overlapping ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
10
Please specify the end limit of your first range
15
Please specify the start limit of your second range
7
Please specify the end limit of your second range
12
--------------------------------
You have specified the following ranges:
        Range 1: 10 - 15
        Range 2: 7 - 12
--------------------------------
The overlap between the two ranges is: 10 11 12.
The percentage overlap between the two ranges is: 33.0%
```

Example 4: Another sample output with partially overlapping ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
10
Please specify the end limit of your first range
20
Please specify the start limit of your second range
13
Please specify the end limit of your second range
17
--------------------------------
You have specified the following ranges:
        Range 1: 10 - 20
        Range 2: 13 - 17
--------------------------------
The overlap between the two ranges is: 13 14 15 16 17.
The percentage overlap between the two ranges is: 45.5%
```

Example 5: Another sample output with partially overlapping ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
5
Please specify the end limit of your first range
9
Please specify the start limit of your second range
7
Please specify the end limit of your second range
13
--------------------------------
You have specified the following ranges:
        Range 1: 5 - 9
        Range 2: 7 - 13
--------------------------------
The overlap between the two ranges is: 7 8 9.
The percentage overlap between the two ranges is: 33.3%
```

Example 6: Sample output with fully overlapping ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
10
Please specify the end limit of your first range
20
Please specify the start limit of your second range
10
Please specify the end limit of your second range
20
--------------------------------
You have specified the following ranges:
        Range 1: 10 - 20
        Range 2: 10 - 20
--------------------------------
The overlap between the two ranges is: 10 11 12 13 14 15 16 17 18 19 20.
The percentage overlap between the two ranges is: 100.0%
```

Example 7: Sample output with non-overlapping ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
5
Please specify the end limit of your first range
10
Please specify the start limit of your second range
11
Please specify the end limit of your second range
15
--------------------------------
You have specified the following ranges:
        Range 1: 5 - 10
        Range 2: 11 - 15
--------------------------------
The two ranges do not overlap.
The percentage overlap between the two ranges is: 0.0%
```

Example 8: Sample output with negative values.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
-10
Please specify the end limit of your first range
-8
Please specify the start limit of your second range
-9
Please specify the end limit of your second range
-5
--------------------------------
You have specified the following ranges:
        Range 1: -10 - -8
        Range 2: -9 - -5
--------------------------------
The overlap between the two ranges is: -9 -8.
The percentage overlap between the two ranges is: 33.3%
```

Example 9: Sample output with incorrect ranges.

```
--------------------------------
This program takes as an input the start and end limits of two integer ranges.
It then prints out the value overlap and the percentage overlap between the ranges.
--------------------------------

Please specify the start limit of your first range
10
Please specify the end limit of your first range
5
Please specify the start limit of your second range
10
Please specify the end limit of your second range
15
--------------------------------
You have specified the following ranges:
        Range 1: 10 - 5
        Range 2: 10 - 15
--------------------------------
--------------------------------
ERROR: Your input is incorrect.
Ensure that the start limit is smaller or equal than the end limit for both ranges.
--------------------------------

Please specify the start limit of your first range
```