CSE 110 - Lab 3

Lab Topics

- Using the Scanner Class with input validation
- Getting familiar with the basic flow control in Java.
- Input validation

Problem Description: Final Weighted Total Calculator

Your task is to design a program which takes 3 user inputs as homework grade, midterm exam grade, and final exam grade. Then use the following formula to calculate the weighted total:

$$Total_{weighted} = \left(Exam_{final} \ / \ 200 \ \times 50\right) + \left(Exam_{midterm} \times 0.25\right) + \left(Homework \times 0.25\right)$$

If the weighted total is greater or equal to 50 (>=50), show *"Student PASSED the class"*. Otherwise, show *"Student FAILED the class"*.

Your program has to validate the input values. A homework grade and midterm exam score should be in the range [0, 100]. The final exam grade should be in the range [0, 200]. If an input is invalid, you have to let the user retry immediately until it is a valid input.

Note: You must use only one loop

Sample Output

Sample Run 1 (The user passed the class):

```
Enter your HOMEWORK grade: 100

Enter your MIDTERM EXAM grade: 76

Enter your FINAL EXAM grade: 80

[INFO] Student's Weighted Total is 64.00

[INFO] Student PASSED the class
```

Sample Run 2 (The user failed the class):

```
Enter your HOMEWORK grade: 30

Enter your MIDTERM EXAM grade: 40
Enter your FINAL EXAM grade: 35

[INFO] Student's Weighted Total is 26.25

[INFO] Student FAILED the class
```

Sample Run 3 (Input validation, minimum requirement)

```
Enter your HOMEWORK grade: 100

Enter your MIDTERM EXAM grade: 76

Enter your FINAL EXAM grade: 8000
[ERR] Invalid input. A final grade should be in [0, 200].

Enter your FINAL EXAM grade: 40

[INFO] Student's Weighted Total is 54.00

[INFO] Student PASSED the class
```

Practice while-loops:

1. Below is the code that repeatedly prompts until the user types a non-negative number, then computes its square root. Sample of execution:

```
Type a non-negative integer: -5
Invalid number, try again: -1
Invalid number, try again: -235
Invalid number, try again: -87
Invalid number, try again: 121
The square root of 121 is 11.0
```

```
System.out.print("Type a non-negative integer: ");
int number = scan.nextInt();

while (number < 0)
{   System.out.print("Invalid number, try again: ");
     number = scan.nextInt();
}
System.out.println("The square root of " + number + " is " +
Math.sqrt(number))
```

Notice that number has to be declared outside the loop.

2. **Sentinel Loop**: A value that signals the end of user input.
   sentinel loop: Repeats until a sentinel value is seen.

Example: A program that repeatedly prompts the user for numbers until the user types -1, then outputs their sum. (In this case, -1 is the sentinel value.)

Enter a number (-1 to quit): **5**
Enter a number (-1 to quit): **15**
Enter a number (-1 to quit): **30**
Enter a number (-1 to quit): **-1**
The sum is 50
We need to use a pattern like this:

```
Declare the loop control variable num
prompt for input;
read the first input, and store it in num
while (num is not equal to -1) {
   add num to sum
   prompt for input;
   read the next num (input)
}
```

A Challenge (not required) do it for fun

Limit the number of chances that the user can retry to 3. It means when one input is invalid, the user will have 3 extra times to retry. If a user has retried three times consecutively and fails, the program loop should terminate immediately and show the error message: *"You have retried 3 times, Please restart your program."* In total, a user can input four times in a sequence with the first three inputs invalid. The counter will be reset to 0 when an input is correct. Your program's output messages should change according to the user input states.

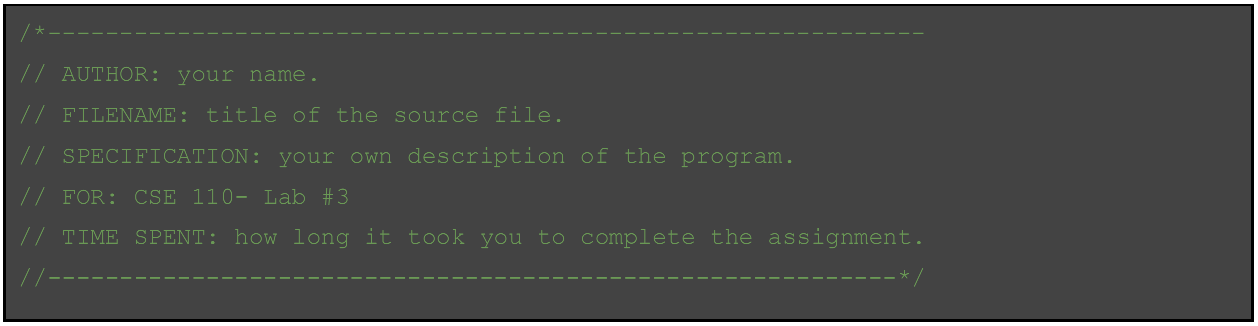
Sample Run for Challengers

```
Enter your HOMEWORK grade: 100

Enter your MIDTERM EXAM grade: 76

Enter your FINAL EXAM grade: 1000
[ERR] Invalid input. A final grade should be in [0, 200].

Enter your FINAL EXAM grade (2 chances left): -1293
[ERR] Invalid input. A final grade should be in [0, 200].

Enter your FINAL EXAM grade (1 chances left): 8023
[ERR] Invalid input. A final grade should be in [0, 200].

Enter your FINAL EXAM grade (0 chances left): 12342314
[ERR] Invalid input. A final grade should be in [0, 200].


[ERR] You have retried 3 times. Please restart your program.
```

Step 1: Get started with the header

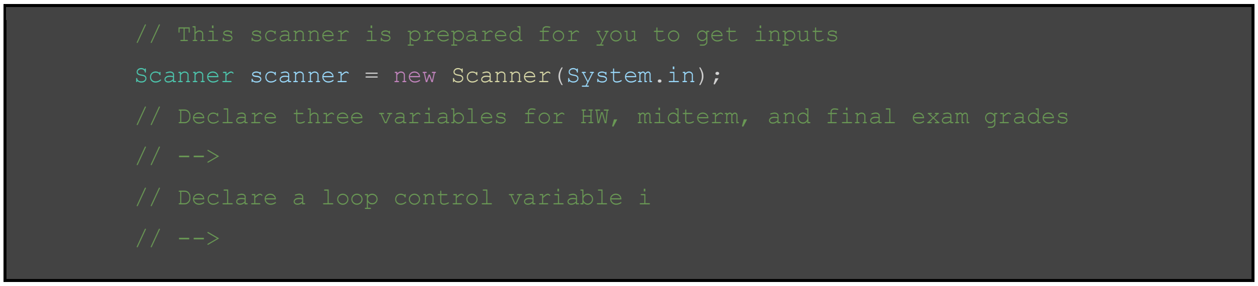
Create a class called Lab3. Use the same setup for setting up your class and main method as you did in previous labs. Be sure to name your file Lab3.java.

At the beginning of each programming assignment you must have a comment block with the following information:



```
/*------------------------------------------------------------
// AUTHOR: your name.
// FILENAME: title of the source file.
// SPECIFICATION: your own description of the program.
// FOR: CSE 110- Lab #3
// TIME SPENT: how long it took you to complete the assignment.
//----------------------------------------------------------*/
```

Step 2: Declare variables you need



```
        // This scanner is prepared for you to get inputs
        Scanner scanner = new Scanner(System.in);
        // Declare three variables for HW, midterm, and final exam grades
        // -->
        // Declare a loop control variable i
        // -->
```

Since your program will ask the user for the homework score, midterm exam score, and final exam score. Please declare three double variables to save these input values.

In addition, you might need a variable to control your while loop. Please declare an integer i for this purpose.

You can declare extra variables if you think they are needed.

Step 3: Use a while loop with the right condition

```
        while (/* Put in the condition involving i */) {
        }
```

We want the program to prompt the user and restart when the user input is invalid. Intuitively, we should use a loop structure to make it restart if a certain condition is not correct. Please first make a while loop statement and one loop control variable i. Then design the loop condition. Refer the template file if you are not sure what to do.

Hint: To help you to write the condition, think about the steps you need to implement:

1. Inside this while loop, the program asks three inputs from the user.
2. If any of the three inputs is not valid, the loop will repeat and asks the user to input the again (for example, if the midterm grade is invalid, let the user type the midterm grade again. Do NOT ask the user start from the homework grade.)
3. The while loop will stop when all inputs are done and valid.

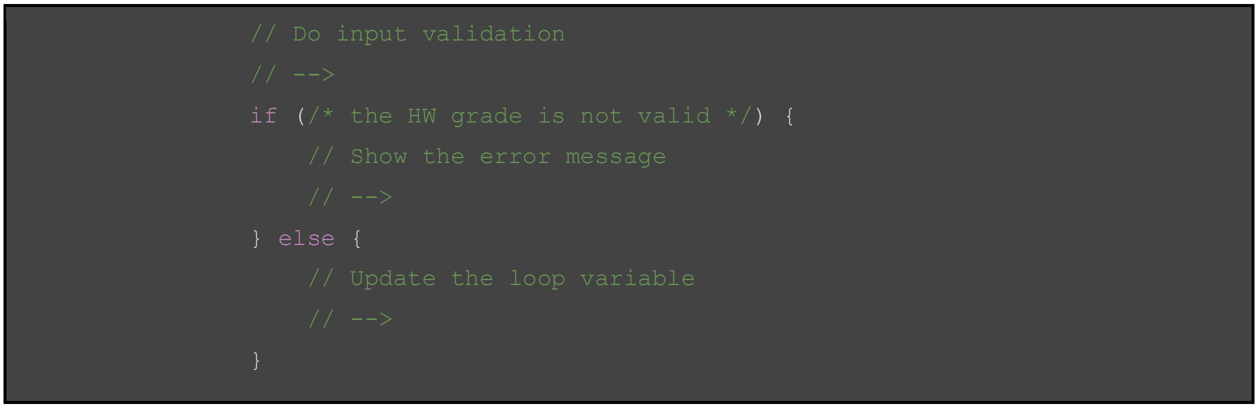
Step 4: Use if-else statements to check for invalid input

```
            if (i == 0) {
                // Ask the user for homework grade
                // -->
                // ...
            }
```

Inside the while loop, the program asks the user for the three grades. Since we are using one while loop to get the three inputs, we need a proper flow control in the loop for the valid inputs. To do so, construct three if-else statements involving your loop variable i, specifically:

1. if i is 0, asks for the homework grade
2. if i is 1, asks for the midterm exam grade
3. if i is 2, asks for the final exam grade

Step 5: Use if-else statements to validate the input values



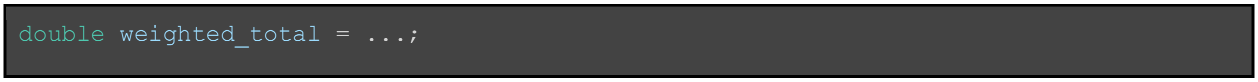
```
                // Do input validation
                // -->
                if (/* the HW grade is not valid */) {
                    // Show the error message
                    // -->
                } else {
                    // Update the loop variable
                    // -->
                }
```

Following Step 4, inside each if-else statement the program asks for the user input. To ensure the user gives you a valid number, you have to design another if-else statement to validate the input. For your reference, the valid ranges are

1. [0, 100] for a homework grade
2. [0, 100] for a midterm exam grade
3. [0, 200] for a final exam grade

Hint: You should update the loop control variable i when any of these inputs is valid, so as to make your program approach the terminal condition of the loop.

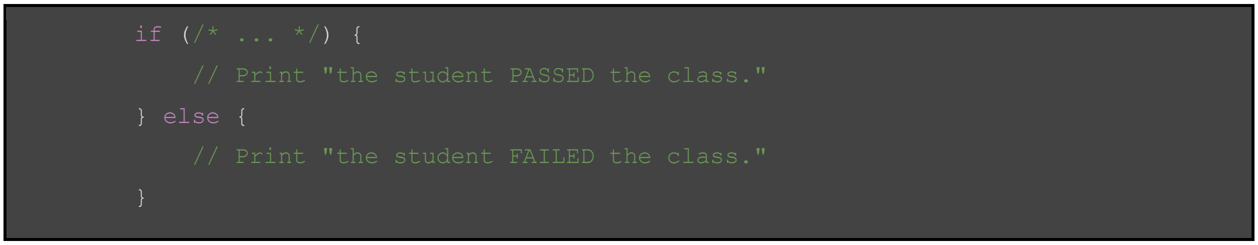
Step 6: Calculate the weighted total and show the value



```
double weighted_total = ...;
```

After the while loop, calculate the weighted total by the formula provided above. The result should be saved into another variable and showed in your console.

Step 7: Show the student passed or failed



```
        if (/* ... */) {
            // Print "the student PASSED the class."
        } else {
            // Print "the student FAILED the class."
        }
```

Finally, according to the weighted total, the program determines whether the user passed the class or not. Design another if-else statement to do so. The conditions are

1. if the weighted total is greater than or equal to 50, show "[INFO] The student PASSED the class"
2. Otherwise, show "[INFO] The student FAILED the class".

Step 8: Make sure to upload your Lab3.java

Please submit your Lab3.java ONLY to "Lab 3 Upload Link" in Weekly Labs section. Do not include any Eclipse project files.