G51OOP Coursework 1.
Darts - 10% of final grade

Question

You are to write a java program which will keep track of the score for a darts game. When the game is over, your program will print out statistics from the game. For simplicity, the game will only be one player.

Your application needs to read integers from user input, such as the target score (e.g 301 or 501), and the scores of each dart thrown. To achieve this, you should make use of the G51OOPInput class provided on Moodle. You also need to download the skeleton java file, named DartsGame.java to get started. If you intend to work from home, you should also install Eclipse by following the guide on Moodle. Once you have Eclipse set up, you should put both the G51OOPInput and the DartsGame.java files in your project. You should only write or change any code in the DartsGame.java file.

When your program starts, it should first prompt for a target score (**Be careful to follow the exact format, including spacing for the output)**. Then, your program should print out a message with the round number, followed by a prompt for user input of an integer representing the score of the first dart. This prompt includes the remaining score. A normal round of darts has 3 throws, so the prompt for a throw should be repeated three times e.g.

Output example (Green represents user input):

```
Please enter a target score: 501
Next Round is round 1
Please enter a throw (remaining score is 501): 60
Please enter a throw (remaining score is 441): 11
Please enter a throw (remaining score is 430): 60
```

Once the first round is complete, the second round will begin, and so on until the remaining score is 0. If a throw exceeds the remaining score, a message with the text "BUST!" should be printed, and the next round begins with the score from the start of the previous round (See expected output example). If the user enters an invalid throw (Less than zero or more than 60), a message should be printed with the text "Invalid Input! Try again", followed by the same prompt, and the invalid throw **should not be counted.** For

this assignment, we do not require that the first throw or last throw is a double, and any throw which is between 0 and 6 (inclusive) is valid, even if it may not actually be possible on a real darts board, e.g 29. **Note that a round which results in "BUST" should be counted as using all three darts, even if only one was thrown.**

When the remaining score is 0, the following statistics should be printed.

- Total Rounds
- Total Darts
- Highest Scoring Round
- Min Scoring Round
- Best Throw
- Worst Throw
- Average Per Dart (2 decimal places)

Therefore, after every throw you will need to check if it is the highest/lowest, and after every round if it was the highest/lowest round. Any round which results in "BUST", counts as 3 darts, even if 3 darts are not thrown. A round which results in BUST **cannot** be the highest or lowest scoring round. For the final round ending the game, only the darts thrown should be counted (See expected output).

Make sure to follow the exact format shown in the expected output, paying particular attention to spacing and spelling (It is recommended where possible, to copy/paste from the expected output).

Error Conditions

A throw must be between 0 and 60 inclusive. If not, the message "Invalid input! Try again" should be printed.
If a negative target score is entered, the program should exit with the message "Invalid target score".

Notes

For this exercise, **arrays should not be used**.

You have one (1) submission. Requests for re-submission with reason stated need to be addressed to Colin Higgins.

Any constant values should be declared as `final` and `static` in your program.

The values in the output other than the average should all be whole numbers with no decimal places shown. The output for the average should always show two decimal places, as shown in the example output. You may want to use printf to format your output correctly, and make sure to use the correct variable types.

You can be up to two days late with standard University penalties applied (ie 5% per day or part thereof). Later than 2 days will attract a 100% penalty (ie no marks will be awarded).

Example Output (Green represents user input)

Example 1 (Normal usage):

```
Please enter target score:  501
Next round is round 1
Please enter a throw (remaining score is 501):  60
Please enter a throw (remaining score is 441):  60
Please enter a throw (remaining score is 381):  60
Next round is round 2
Please enter a throw (remaining score is 321):  1
Please enter a throw (remaining score is 320):  2
Please enter a throw (remaining score is 318):  3
Next round is round 3
Please enter a throw (remaining score is 315):  60
Please enter a throw (remaining score is 255):  50
Please enter a throw (remaining score is 205):  40
Next round is round 4
Please enter a throw (remaining score is 165):  60
Please enter a throw (remaining score is 105):  0
Please enter a throw (remaining score is 105):  15
Next round is round 5
Please enter a throw (remaining score is 90):  60
Please enter a throw (remaining score is 30):  30
=== Statistics===
Total Rounds: 5
Total Darts: 14
Highest Scoring Round: 180
Min Scoring Round: 6
Best Throw: 60
Worst Throw: 0
Average Per Dart: 35.79
```

Example 2 (BUST!):

```
Please enter target score:  301
Next round is round 1
Please enter a throw (remaining score is 301):  50
Please enter a throw (remaining score is 251):  50
Please enter a throw (remaining score is 201):  50
Next round is round 2
```

```
Please enter a throw (remaining score is 151):  60
Please enter a throw (remaining score is 91):  60
Please enter a throw (remaining score is 31):  60
BUST!
Next round is round 3
Please enter a throw (remaining score is 151):  60
Please enter a throw (remaining score is 91):  30
Please enter a throw (remaining score is 61):  30
Next round is round 4
Please enter a throw (remaining score is 31):  10
Please enter a throw (remaining score is 21):  22
BUST!
Next round is round 5
Please enter a throw (remaining score is 31):  10
Please enter a throw (remaining score is 21):  20
Please enter a throw (remaining score is 1):  1
=== Statistics===
Total Rounds: 5
Total Darts: 15
Highest Scoring Round: 150
Min Scoring Round: 31
Best Throw: 60
Worst Throw: 1
Average Per Dart: 20.07
```

Notes:

- The highest round resulted in “BUST” so does not count towards the highest scoring round statistic.
- The number of darts is 15, rounds which went BUST count as all 3 darts thrown.
- The best/worst throws **do** still count, even if the round was bust.

Example 3 (Invalid Input):

```
Please enter target score:  301
Next round is round 1
Please enter a throw (remaining score is 301):  61
Invalid input! Try again
Please enter a throw (remaining score is 301):  -1
Invalid input! Try again
Please enter a throw (remaining score is 301):  60
Please enter a throw (remaining score is 241):  60
Please enter a throw (remaining score is 181):  60
Next round is round 2
Please enter a throw (remaining score is 121):  30
Please enter a throw (remaining score is 91):  50
Please enter a throw (remaining score is 41):  61
```

```
Invalid input! Try again
Please enter a throw (remaining score is 41):  1
Next round is round 3
Please enter a throw (remaining score is 40):  40
=== Statistics===
Total Rounds: 3
Total Darts: 7
Highest Scoring Round: 180
Min Scoring Round: 40
Best Throw: 60
Worst Throw: 1
Average Per Dart: 43.00
```

Note: The best throw was not 61, and the worst throw was not -1, these throws do not count.

Example 4 (Invalid Target):

```
Please enter target score:  -1
Invalid target score
```

Mark Scheme

You will be marked on correctness (if your output is correct for the given input). There are also marks for variable naming, layout, efficiency and output format. Marks will be given for correct use of final variables, selection statements, correct output streams and a sensible choice of loop. As with any programming, you should use comments where necessary in your code to make it more readable. You may be marked down for failure to submit the correct files, or follow the specification (See submission).

Submission

Submission is via Moodle. You should submit a .zip file containing only DartsGame.java and G51OOPInput.java. You should not submit any other format other than .zip (no .iso, .rar, .7z or other alternatives). Make sure to submit the .java files, and not the .class files.