Project #1: Time Interpreter

Overview

Write a program that accepts a time from the keyboard representing and prints the times in simplified form.

Input

The program must accept times in the following form `<hours> [space] <minutes> [space] <seconds>` where each `<hours>`, `<minutes>`, and `<seconds>` are integers and `[space]` represents the space key being hit.

Prompt the user with the exact phrasing of the sample input / output shown below; note that the input from the keyboard is depicted in red:

```
Enter the time in the form <hours minutes seconds>:

1 2 3

The time consists of 3723 seconds.

Simplified time: 1:2:3
```

Requirements

- The name of the class that contains the main must be `TimeInterpreter`.
- While input uses spaces between the input numbers, the output format with days, hours, minutes, and seconds should be delimited by colons; see sample output for examples.
- All times will be output without spaces (or other whitespace).
- *Negative Times.* If a specified time is negative, it should be printed with a single leading negative. For example, `0:-2:-34` is output as `-2:34`.
- *Simplification.* Times must be simplified before printed. For example, `12:2:-34` is simplified and output as `12:1:26`.
- *Output Brevity.* For input time `0:2:34`, the corresponding output should not list the number of hours (since there are none): `2:34`.
- A single output print statement will be allowed in the final solution code. That is, a proper solution will construct a `String` object and output it at the end of the program.
- You must define and use constants representing the number of seconds per minute, hour, and day.

Sample Input / Output

Please use all of the following examples for test cases (your instructor will), but do not limit your tests to these input times. There is no need to represent time beyond days in your output.

Input Time	Expected Output Time
1 2 3	1:2:3
1 2 90	1:3:30
1 2 -90	1:0:30
48 0 0	2:0:0:0
0 0 0	0
0 2 -120	0
1 -300 2	-3:59:58

Recommendations

- Compute the number of seconds represented by the input and use the number of seconds for the basis of your output; use a `long` for this quantity.
- Before coding, work out on paper the logic and computations for a given time.
- Be thorough in your testing because it is an instructor's job to try and break your code.

Submitting

Header Comments

Your program must use the following standard comment at the top of *each source code file*. Copy and paste this comment and modify the parenthesized values accordingly.

```
/*
 *  @author (Student Name)
 * <p>      (File Name)
 * <p>      (Assignment)
 * <p>      (Describe, in general, the code contained.)
 */
```

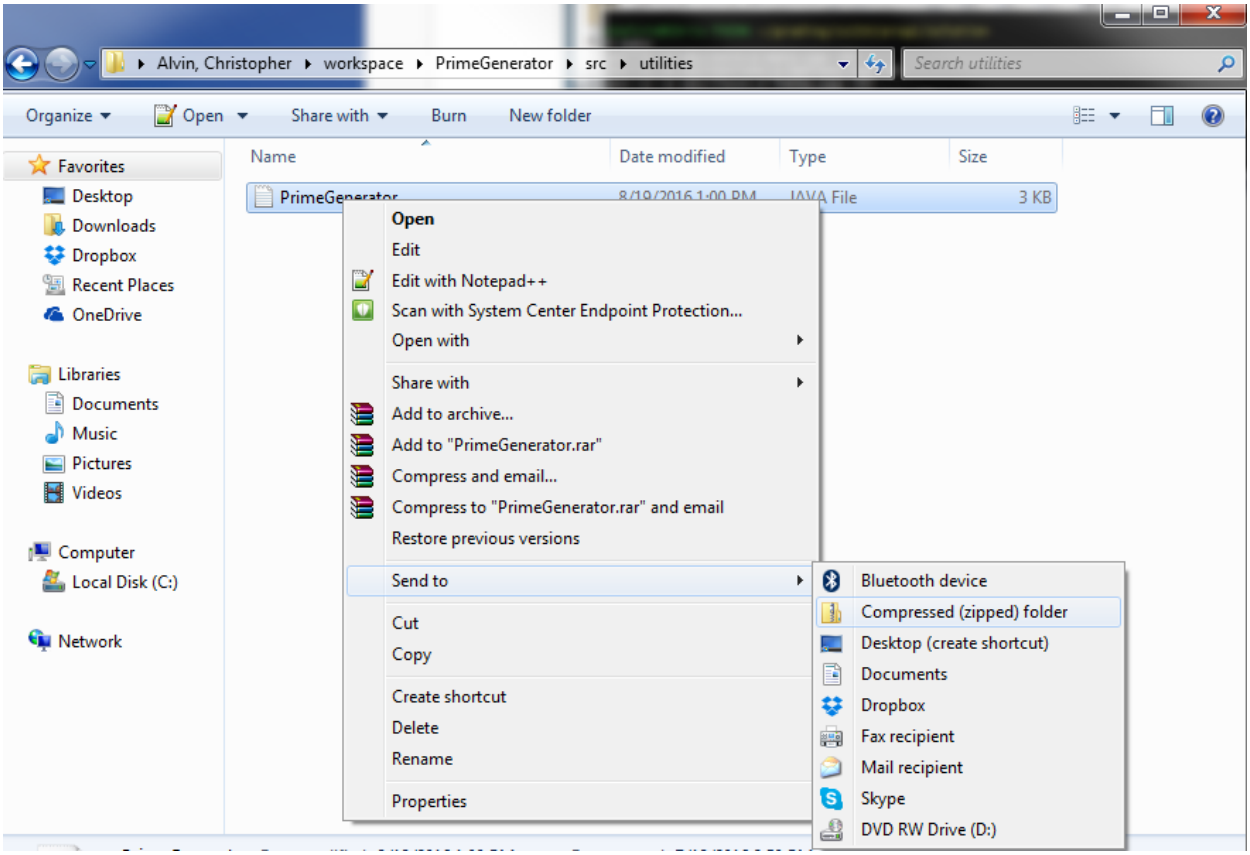
Inline Comments

Please comment your code with a *reasonable amount of comments* throughout the program. Each *block* of code (3-4 or more lines in sequence) in a function should be commented.

Although it is an issue of style and preference, please avoid *long* comments to the right of lines of source code. Long, ubiquitous comments to the right of code will result in a deduction.

Final Submission File

Create a zip file (`proj1.zip`) containing *only* the source code files (TimeInterpreter.java). Please note that the zip must not contain any subfolders or other extraneous files. In Windows, (1) select all the source files in a folder, (2) right-click, and (3) Send to > Compressed (zipped) folder:



Submit your zip file via Canvas under `Assignments > Project 1`. Be sure to review the university policy on academic dishonesty: this is an individual project.