

cs171: Introduction to Programming
Programming Assignment 10

Due Date: Monday, April 10

Points: 70

Here we design a class that will allow us to keep track of and manipulate times in hh:mm formats. You are to design a class that implements the API description (attached).

Too many students are starting their work too late and thus are submitting work below acceptable quality. We will be trying a new grading strategy.

- If you send me a solution within the first 55 hours (7pm on Wednesday), I will grade it overnight and send the results back to you over email. You may then either keep the grade you receive or make corrections and resubmit for a second grade. Your grade for the assignment will be the higher of the two.
- I will not answer email questions about the program after 7pm on Saturday. It will be up to you to finish the program if you still have major parts to accomplish in the last day.

Once again, there will be multiple parts. The additional parts are more difficult but you will be eligible for a higher grade.

- Go to the class webpage and download `Program10.java`. This is my main program for this assignment. I will use this same program to test your code. At a minimum, you should be sure your program works using this main method. Notice Part II is commented out here. If you implement Part II be sure to switch the comment lines in the main program.
- You are to create a `TimeStamp` class. The full API description is attached to this handout; thus you have very little design work to complete. Most of your effort is to implement the API in Java code. All of your work will be in the `TimeStamp.java` file; you will email me this at the end.
- Implement the instance variable section of the `TimeStamp` class.
- Implement the methods section of the `TimeStamp` class.
- Be sure to comment methods appropriately and use `private` and `public` correctly. Test your code on other input besides those indicated in the main program. It is **your** responsibility to see that the code you submit works on all inputs. I am seeing too many errors that students did not check for during their testing phase.

- There are three parts to the lab. Here is a description of each part and how many points you will earn (a total of 70 points).
 1. (55 points) Implement all methods in the API except for the second constructor which accepts a `String` as input.
 2. (10 points) Implement the second (`String`) constructor.
 3. (5 points) Implement JavaDoc style comments. Try the JavaDoc preview button in your "Tools" menu bar option. See if you can generate a JavaDoc as similar to the attached API as possible. There is a line on our class webpage with instructions how to write JavaDoc style comments.
- Here is the output of my program running.

```
14:20 - 12:10 = 2:10
12:10 - 14:20 = 0:0
```

Name your program `TimeStamp.java` and email me the source code **before** the 11:30am on Monday, April 10. I will not accept any solutions sent after 11:30am exactly on that Monday – don't wait till the last minute.

Hints for Part 2

When you implement the second constructor, you will have to parse a `String` such as "8:47" to separate the hours and minutes. Go back through the `String` class API to see what methods will be useful. We have seen them all before, there is nothing new needed from this class that we didn't at least mention in class. Think about dividing the input string into two new strings, one containing only the hours ("8") and a second one containing only the minutes "47").

You will also need the ability to parse a string and retrieve a number. Check out the `Integer` class API. There is a method here which does exactly this task.

This Part 2 is not difficult. I have hinted at or told you about the tools you need to complete this method. The method should only be a few lines of code, it is not lengthy. Still you will need to set aside some time to test various methods in the `String` and `Integer` classes to be sure you understand how they work and how they can be used to complete this second part of the assignment.