

LAB #4: CLASSES

1. Task: Create a `Time` class. The class should contain 3 fields (integers called `hrs`, `mins`, `secs`, for hours, minutes and seconds) and the following methods:

1. Default and alternate constructors.
2. Three getters (accessors) to return hours, minutes and seconds
3. A method named `set`; invalid values set to 0
4. Two methods named `printMilitary` and `printStandard` to print each `Time` object in military and standard format
5. A method `toString()`
6. A method named `equals` to compare 2 `Time` objects for equality
7. 2 methods named `copy` and `getCopy` to make a copy of a `Time` object into another `Time` object
8. A method named `advanceSecs` to advance time by 1 second
9. A method named `lessThan` to decide if a time comes before another time

2. Task: Create a client for the `Time` class. Be very thorough with your testing (including invalid input) and have output similar to the sample output below:

```
Initial time t1 (alternate constructor invoked) - military format: 08:15:30
Initial time t1 (alternate constructor invoked) - standard format: 8:15:30 AM
Initial time t2 (default constructor invoked) - military format: 00:00:00
Initial time t2 (default constructor invoked) - standard format: 12:00:00 AM
t2 after call to setTime - military format: 09:45:35
t2 after call to setTime - standard format: 9:45:35 AM
After call to equals: times are NOT equal.
After call to lessThan: t1 is less than t2.
Enter hours, minutes, and seconds: 10 11 12
New time t1 after call to setTime - standard format: 10:11:12 AM
New time t1 after call to advanceSecs - standard format: 10:11:13 AM
New t2 after call to copy - standard format: 10:11:13 AM
Test toString for t2: 10:11:13
```

3. Task: Modify the `Time` class and add more methods.

3.1. The class `Time` (discussed in class) has 2 member methods `equals` and `lessThan`. It would be easier for the user to have access to new member methods `notEquals`, `lessOrEquals`, `greaterThan`, and `greaterOrEquals`. **Hint:** Don't start from scratch; make use of the already existing methods.

```
public boolean notEquals(Time otherTime) {... }
public boolean lessOrEquals(Time otherTime) {... }
public boolean greaterThan(Time otherTime) {... }
public boolean greaterOrEquals(Time otherTime) {... }
```

3.2. Have 3 methods to advance time, not only `advanceSecs`. Add to class `Time` the methods `advanceHrs` and `advanceMins` (you may have to change the method `advanceSecs`)

3.3. Change the method `getCopy()` – try to have a single return statement

3.4. Add 2 more methods named `toMilitary` and `toStandard`; they should return a `String` with the military and standard format for a `Time` object.

Modify the client for complete testing, including the new methods.

Notes:

A. The lab will NOT be graded but you have to submit good quality work in order to get credit.

B. The lab should be completed by the start of the next scheduled lab class. E-mail the **.java** files (attachments) to Rohan Patel (rpatel27@students.towson.edu)

Very important: Make sure that you have COSC 237.section, your name, and Lab#4 in the *Subject* box of your e-mail.

C. In case you have any problems, contact the instructor or the TA for assistance.