

**Programming Exam 2**

**Instructions.** This exam has one question. You have 50 minutes. The exam is *open course materials*, which includes the course textbook, the companion booksite, the course website, your course notes, and code you wrote for the course. Accessing other information or communicating with a non-staff member (such as via email, text, Facebook, Piazza, phone, or Snapchat) is prohibited.

**Submission.** Submit your solution via the link on the *Class Meetings* page. Click the *Check All Submitted Files* button to verify your submission. You may submit multiple times.

**Grading.** Your program will be graded for correctness, clarity (including comments), design, and efficiency. You will receive partial credit for a program that correctly implements some of the required functionality. You will receive a substantial penalty if your program does not compile.

**Discussing this exam.** Discussing or communicating the contents of this exam before solutions have been posted is a violation of the Honor Code.

**This exam.** You must turn in this exam. Print your name, NetID, precept, and the room in which you are taking the exam in the space below. Also, write and sign the Honor Code pledge. You may fill in this information now.

**Name:**

**NetID:**

**Exam Room:**

**Precept:**

*“I pledge my honor that I will not violate the Honor Code during this examination.”*

---

*Signature*

**Problem.** Write a mutable data type `Clock.java` that represents time on a 24-hour clock, such as 00:00, 13:30, or 23:59. Time is measured in *hours* (00–23) and *minutes* (00–59).

**Part I (27 points).** Implement the following API:

```
public class Clock


---


public Clock(int h, int m)           creates a clock whose initial time is
                                     h hours and m minutes
public String toString()             returns a string representation of
                                     this clock, using the format HH:MM
public boolean isEarlierThan(Clock that) is the time on this clock earlier than
                                     the time on that one?
public void tic()                    adds 1 minute to the time on this clock
public static void main(String[] args) tests all instance methods in this class
```

Here is some more information about the required behavior:

- *Two-argument constructor.* Throw an `IllegalArgumentException` if either integer argument is outside its prescribed bounds (hours between 0 and 23, minutes between 0 and 59).
- *String representation.* The format is the hours (2 digits), followed by a colon, followed by the minutes (2 digits). Two examples are 00:00 and 23:59.
- *Ordering.* Times are ordered from 00:00 (earliest) to 23:59 (latest).
- *Tic.* Add one minute to the current time. For example, one minute after 06:00 is 06:01; one minute after 23:59 is 00:00.
- *Test client.* The `main()` method must not only call each instance method directly, but it must also help verify that each method works as prescribed (e.g., by printing results to standard output).

**Part II (3 points).** Add the following one-argument constructor to your data type.

```
public Clock(String time)           creates a clock whose initial time is specified
                                     as a string, using the format HH:MM
```

The string argument is composed of two digits, followed by a colon, followed by two digits, such as "09:45". Throw an `IllegalArgumentException` if either the string argument is not in this format or if it does not correspond to a valid time between 00:00 and 23:59.

*Hint.* The function `Integer.parseInt()` ignores any leading zeros. For example, `Integer.parseInt("09")` returns the integer 9.