

Lab 3 (September 18 or September 19)

Instructions: Complete the steps below. Be sure to show your code to one of the lab TAs before you leave, so that you can receive credit for this lab. You must also upload a copy of all your source code (.java) files to the link on Blackboard by 11:59 PM on Tuesday, September 19.

1. Write a program that prompts the user to enter the month and year, and displays the number of days in the month. For example, if the user enters 2 for the month and 2000 for the year, the program should display that February 2000 has 29 days (note that February only has 29 days during a leap year). If the user enters 3 for the month and 2005 for the year, the program should display that March 2005 has 31 days.

NOTE: Your first step (after collecting the user input) should be to determine whether the year entered was a leap year. Leap years follow a specific set of rules; they aren't always every fourth year. Use the following series of tests to determine whether or not a year is a leap year (by "divisible", we mean "divides with a remainder of 0"):

- a. If `year` is not divisible by 4, then it is a common year.
- b. Otherwise, if `year` is not divisible by 100, then it is a leap year.
- c. Otherwise, if `year` is not divisible by 400, then it is a common year.
- d. Otherwise, `year` is a leap year.

This means that 1992, 2000, and 2032 are all leap years, but 1900, 2002, and 2017 are not.

Sample execution: (program output is shown here in *italics*, while user input is shown here in **boldface**)

Please enter the month (1-12): **8**

Please enter the year: **2014**

August 2014 has 31 days.

Do not worry about grammatical variations in the output message like "had", "has", and "will have" based on whether the date entered is past or future.

2. Write a program that reads the following information and prints a neatly-formatted payroll statement (don't worry about rounding off any of the output to two decimal places):
 - a. Employee's name (a `String` like "Smith")
 - b. Number of hours worked in a week (a `double`)

- c. Hourly pay rate (a double like 9.75)
- d. Federal tax withholding rate (a double like 0.20 for 20%)
- e. State tax withholding rate (a double like 0.09 for 9%)

Sample execution: (program output is shown here in *italics*, while user input is shown here in **boldface**)

*Enter employee's name: **Smith***
*Enter the number of hours worked this week: **10***
*Enter the hourly pay rate: **9.75***
*Enter the federal tax withholding rate: **0.20***
*Enter the state tax withholding rate: **0.09***

Employee Name: Smith
Hours Worked: 10.0
Pay Rate: \$9.75
Gross Pay: \$97.5

Deductions:

Federal Withholding (20.0%): \$19.5
State Withholding (9.0%): \$8.77

Total Deduction: \$28.27

Net Pay: \$69.22

Grading Guidelines: This lab is graded on a scale of 0-3 points, assigned as follows:

0 points: Student is absent or does not appear to have completed any work for the lab

1 point: Student has written only one program, but it does not compile or run at all due to errors.

2 points: Student has written (or attempted to write) both programs, but only one compiles and runs without error.

3 points: Student has written both programs, and they both compile and run correctly, without any apparent errors.