

Doomsday Calculator

1 Introduction

This assignment will cover basic console I/O, conditional operation, and simple algorithms.

2 Problem Description

More likely than not, you have probably seen a movie or watched a show where a character (usually a “math genius”) asks someone for their birthday, then instantly rattles off the day of the week they were born. This isn’t some magic trick—it’s a simple algorithm that you can learn to do in your head. But here in CS 1331, we don’t do math in our head! That’s what computers are for.

Your homework will be to implement a Doomsday Calculator using this algorithm for any date in the 1900s.

3 Solution Description

The basis behind the [Doomsday Algorithm](#) is simple. As everyone knows, every seven days the sequence {Sunday, Monday, ...} recurs. Which day a year starts on is similarly cyclical; if the 1st of a given year is on a Monday, then the 1st of the next year must be 365 days later, a Tuesday. To see this, consider how many weeks there are in those 365 days: $\frac{365}{7} = 52\frac{1}{7}$, i.e. 52 weeks and one additional day. It’s because of the remainder that successive years begin on successive days of the week.

A convenient way of computing these sorts of remainders is the modulo operator: $365 \% 7 = 1$, for example. $1 \% 7 == 8 \% 7$, and so on and so forth.

Using this information, the following algorithm finds a year’s doomsday:

1. Divide the last two digits of the year by 12.
2. Modulo the last two digits of the year by 12.
3. Divide the answer to step 2 by 4.

4. Sum steps 1-3 and add the century's "anchor day". In the case of the 1900's for this assignment, this is 3.
5. Modulo the previous sum by 7 to get the "doomsday" for that year. A 0 corresponds to Sunday, 1 to Monday, etc.

The doomsday for a year is a day of the week on which each of the following dates occur:

- January 3rd (or 4th, for a leap year)
- February 28th (or 29th, for a leap year)
- March 7th
- April 4th
- May 9th
- June 6th
- July 11th
- August 8th
- September 5th
- October 10th
- November 7th
- December 12th

Finally, you take the difference between the day you are searching for and the doomsday of the month, then add the day you calculated in steps 1-5. Modulo that final answer by 7 to get the day of the week for the date you are looking for.

Some examples:

- February 5th, 1994 = Saturday
- July 1st, 1992 = Wednesday
- October 9th, 1920 = Saturday

4 Example Output

The program should prompt the user for the year, month, and day before returning with the week-day that day was on.

```
$ java Doomsday
Welcome to the Doomsday Calculator!
What year are you looking for? 1993
What month (1-12)? 3
What day? 29

3/29/1993 was on a Monday.
$
```

5 Checkstyle

Review the [Style Guide](#) and download the [Checkstyle](#) jar and associated XML file. Run Checkstyle on your code like so:

```
$ java -jar checkstyle-5.6-all.jar -c cs1331-checkstyle.xml *.java
Starting audit...
Audit done.
```

The message above means there were no Checkstyle errors. You can easily count Checkstyle errors by piping the output of Checkstyle through `wc -l` and subtracting 2 for the two non-error lines printed above (which is how we will deduct points). For example:

```
$ java -jar checkstyle-5.6-all.jar -c cs1331-checkstyle.xml *.java | wc -l
2
```

Alternatively, if you are on Windows, you can use the following instead:

```
C:\> java -jar checkstyle-5.6-all.jar -c cs1331-checkstyle.xml *.java | findstr /v "Starting
audit..." | findstr /v "Audit done" | find /c /v "hascode()"
0
```

Food for thought: is there a one-liner like above that shows you only the number of errors? Hint: `man grep`.

In future homework projects we will run Checkstyle on all the Java source files you submit and deduct one point from your score for each style error found by Checkstyle.

6 Tips

- Think about how you can use modulo to make your life easier on this homework.

- You'll have to compare the month that the user inputs to the doomsday for that month. Rather than comparing them one by one, you could use an array instead!
- The days for each month are only a suggestion; if you feel you have a better solution which fulfills the same requirement, feel free to use it!
- Download the Checkstyle jar and `cs1331-checkstyle.xml` files into a directory you create for this project to make it easy to run Checkstyle. (Is there a way to avoid having to copy the Checkstyle jar file and configuratoin file into each homework subdirectory?)

7 Turn-in Procedure

Submit your `Doomsday.java` file on T-Square. Do not submit any compiled bytecode (`.class` files), the Checkstyle jar file, or the `cs1331-checkstyle.xml` file. When you're ready, double-check that you have submitted and not just saved a draft.

8 Verify the Success of Your Submission to T-Square

Practice safe submission! Verify that your HW files were truly submitted correctly, the upload was successful, and that the files compile and run. It is solely your responsibility to turn in your homework and practice this safe submission safeguard.

1. After uploading the files to T-Square you should receive an email from T-Square listing the names of the files that were uploaded and received. If you do not get the confirmation email almost immediately, something is wrong with your HW submission and/or your email. Even receiving the email does not guarantee that you turned in exactly what you intended.
2. After submitting the files to T-Square, return to the Assignment menu option and this homework. It should show the submitted files.
3. Download copies of your submitted files from the T-Square Assignment page placing them in a new folder.
4. Recompile and test those exact files.
5. This helps guard against a few things.
 - (a) It helps insure that you turn in the correct files.
 - (b) It helps you realize if you omit a file or files.¹ (If you do discover that you omitted a file, submit all of your files again, not just the missing one.)
 - (c) Helps find last minute causes of files not compiling and/or running.

¹Missing files will not be given any credit, and non-compiling homework solutions will receive few to zero points. Also recall that late homework will not be accepted regardless of excuse. Treat the due date with respect. The real due date is midnight. Do not wait until the last minute!