CMPSC 111
Introduction to Computer Science I
Fall 2014

Practical 6
30–31 October 2014
Due in Bitbucket by midnight of the day of your practical
"Checkmark" grade

Summary

In this practical you will create methods to determine if certain events occurred during the year given by the user. For this task you will use `if/else` statements and the boolean logic operators. Finally, using the "`git add`", "`git commit`", and "`git push`" commands you will upload your modified programs and the output you obtain from running your programs to your Git repository hosted by Bitbucket.

Review the Textbook

You may refer to sections 5.1–5.3 in your textbook to learn more about `if/else` statements and boolean expressions.

Save the Example Programs

Get the files "`Practical6.java`" and "`Practical6Main.java`" from the shared course repository. Copy these files into your own Bitbucket repository into a directory called `/practical06`. Study these programs first and make sure you understand them. Please note that these programs are not complete, and will not run correctly.

"Through All the Years"

The quotation is from the College's Alma Mater. In the second class (`Practical6.java`), you will write methods that for the user's input (from `Practical6Main.java`), determine which of the following events occurs that year:

- leap year
- emergence of the 17-year cicadas (more specifically, Brood II)
- peak year of sunspot activity

A year is a *leap year* if it is divisible by 4, *unless* it is a century year. If it is a century year, it is a leap year if it is divisible by 400. For instance, 1968 and 1972 are leap years since they are divisible by 4; 1967 and 1970 are not. The year 2000 is a leap year because it is divisible by 400; however, 1900 is not (even though 1900 is divisible by 4—century years are treated differently).

The 17-year cicadas emerge from underground every 17 years. There are several "broods;" the one we are interested in emerged in 2013. (So any year that differs from 2013 by a multiple of 17 is also an emergence year, e.g., 2040, 1996, 1928, 3713.)

Sunspot activity usually peaks every 11 years. The year 2013 was supposed to be such a "solar max" year. (So any year that differs from 2013 by a multiple of 11 should also be a solar max year, e.g., 2002, 2024, 1793.)

The sample output is shown below:

```
ewire23-29:practical6 jjumadinova$ java Practical6Main

Enter a year between 1000 and 3000: 1452
1452 is a leap year
It's an emergence year for Brood II of the 17-year cicadas
It's a peak sunspot year.
Thank you for using this program!
```

Thoroughly test your program! Try the above date and some of your own as well.

Modify the Programs

- Edit the file `Practical6.java` and find the constructor in this class. Make sure you understand what the constructor contains, when it gets called, and what happens after it is executed. Please ask the instructors or the teaching assistance if you are not clear about the purpose of the constructors.
- In `Practical6.java` and find three methods `setLeapYear(), setCicadaYear(),` and `setSunspotYear()`. You task for this practical is to create the actions that these methods need to execute, as described in the previous section.
- Edit the file `Practical6Main.java` by including method calls to `setLeapYear(), setCicadaYear(),` and `setSunspotYear()`, or in other words, by writing Java statements that will invoke these three methods. You do not need to change anything else in this class.

Completing the Practical Assignment

To finish this assignment and earn a "checkmark", you should submit the `Practical6.java` and `Practical6Main.java` files you just edited in your Bitbucket repository by using appropriate `git` commands. You also need to submit an output file with several runs of your program.

General Guidelines for Practical Sessions

- **Submit *Something*.** Your grade for this assignment is a "checkmark" indicating whether you did or did not complete the work and submit something to the Bitbucket repository using the "`git add`", "`git commit`", and "`git push`" commands.
- **Update Your Repository Often!** You should `add`, `commit`, and `push` your updated files each time you work on them, always including descriptive messages about each code change.
- **Review the Honor Code Policy on the Syllabus.** Remember that while you may discuss your writing with other students in the course, text that is nearly identical to, or merely variations on, the work of others will be taken as evidence of violating the Honor Code.