1 Purpose

Write a program that prints the calendar for a month, given the month and a year.

For example, April, 2009 would display:

April, 2009

S M T W R F S
---------------------
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

Your program must include several methods. The number and type are up to you, but you should incorporate more than just one or two. Having this few will not suffice. And main() does not count as one of your methods. There is ample opportunity to utilize loops and conditionals.

2 Processing

2.1 Input

The user is prompted for the month and year they are interested in. Months are input as integers (1 - 12); years as integers (one or higher). Your code should make sure the input values are in the appropriate ranges and require the user to keep entering values until they get it right.

2.2 Computation: Compute what day of the week the month starts on

What follows is a magic formula for calculating what day of the week a month starts on for a given year. $m$ represents the month, $y$ represents the year. The value being computed is $day$, whose value is between one (Sunday) and seven (Saturday). (NOTE: this is all integer arithmetic).

$$y_0 = y - (14 - m)/12$$
$$x = y_0 + y_0/4 - y_0/100 + y_0/400$$
$$m_0 = m + 12 \times ((14 - m)/12) - 2$$
$$day = (1 + x + (31 \times m_0)/12) \% 7 + 1$$

2.3 Output: Display the calendar

Once your program knows the day of the week that the first is on, it needs to print the calendar. There are many ways in which you could approach this. I would suggest that you might want to use a loop to display the weeks.

Sample output is shown above.

3 Notes

Creating a HIPO chart before attempting to program should be useful. The chart should help in deciding what would be appropriate methods. By breaking your program into modules (methods), you can work on the individual pieces rather than trying to get the whole thing working at once. Here’s where jshell is helpful - you can copy one or more method definitions into jshell to test them individually.
4 Submission

Submit your source code, which is a .java file and should be found in the src folder of your project. See the document on the course web site under the Course Materials link for the proper naming of project files.

Also send a screen shot of a run of your program, and a HIPO chart that shows the structure of your program.

Your program should be well-tested. You should make sure it executes properly on sets of inputs other than the single set used in the above example.

Make sure your program performs the processing. Do not submit a program that simply prints the values you see in the sample output above. Such a program is worthless. Such a submission is worth 20 points at most.

Programs that do not compile are worth at most 50 points.

The program requires a header of the form

```
/*****************************/
 *                     *
 * Author: your name    *
 * Date: date of submission *
/*****************************/
//Description of what the program does
//This could be any number of lines
```

Methods require headers of the form

```
/*****************************/
 *                     *
 * method name          *
/*****************************/
//inputs:
//return value:
//description
```

The program should be well-structured and should have ample documentation (i.e., comments), which includes using variables and methods with meaningful names. The program should be formatted appropriately. These aspects constitute twenty per cent of your grade - see syllabus.

Submit your program via email, with a subject line of project 3.

All work is to be your own. Working together, sharing code, etc. is strictly not acceptable. Any indication that your work is not your own will result in a zero.

Grading will be based on the correctness of your solution, having appropriate and sufficient comments, and having the output formatted as indicated above.