

## Assignment #6

### Goals:

- This lab gives students more experience in
  - Top-down design
  - Procedural abstraction using methods.

### Background Information

A year with 366 days is called a leap year. A year is a leap year if it is divisible by 4 (For example, 1996), except that it is not a leap year if it is divisible by 100 (For example, 1900); however, it is a leap year if it is divisible by 400 (for example, 2000); and there were no leap years before the introduction of the Gregorian Calendar on October 15, 1582.

### Requirement:

Design an algorithm, and then write the java program to ask the user to enter a year greater than 1582. Your program then displays the following:

- If the year is a leap year or not.
- If the given year is a leap year, your program should then **display**
  1. which day of the week was Feb 29<sup>th</sup>, and
  2. the next leap year where Feb. 29<sup>th</sup> happens on the same week day.

The following is a screen sample of the program:

Please enter a year greater than 1582: **2013**

Year 2013 is not a leap year.

Do you want to repeat the program: **yes**

Please enter a year greater than 1582: **2000**

Year 200 is a leap year

Feb 29, 2000 is on Tuesday

The next leap year where Feb 29 is on Tuesday is year 2028

Do you want to repeat the program: **no**

Program terminates

### NOTE:

- Please call your class **LeapYearInformation**
- Use the following algorithm, known as Zeller's Congruence, to compute a single digit, **0** to **6**, for Sunday through Saturday. For example, **22/10/1999** has occurred on Friday so the zellers(1999, 10, 22) returns **5** representing Friday.

## Zeller's Algorithm

### Define Terms

- 0.1 Given the YEAR, MONTH and DAY, representing a date.
- 0.2 Let DayNumber be the an integer number representing the day of the week( 0 represents Sunday, 6 represents Saturday)
- 0.3 Let StartMonth, StartYear, and LeapFactor be three integer variables used in the calculation

### Start

#### 1. IF MONTH < 3

- 1.1 StartMonth = 0
- 1.2 StartYear = YEAR - 1

#### OTHERWISE

- 1.3 StartMonth = INT (0.4 \* MONTH + 2.3)
- 1.4 StartYear = YEAR

#### 2. LeapFactor = (StartYear / 4) - (StartYear / 100) + (StartYear / 400)

#### 3. RETURN DayNumber = ((365 \* YEAR + 31 \* (MONTH - 1) + DAY + LeapFactor - StartMonth) - 1) MOD 7

## Marking Scheme

[10 marks] Presentation/Style: Organization, readability, descriptive identifier, indentation, bracket placement.

[70 marks] Correctness: Program works without bugs.  
Appropriate use of methods.

[20 marks] Documentation: Internal and External (pseudo code for your main method only)