# Hands-on Experiment # 11: Worksheet

Section\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No more than 3 students per one submission of this worksheet.

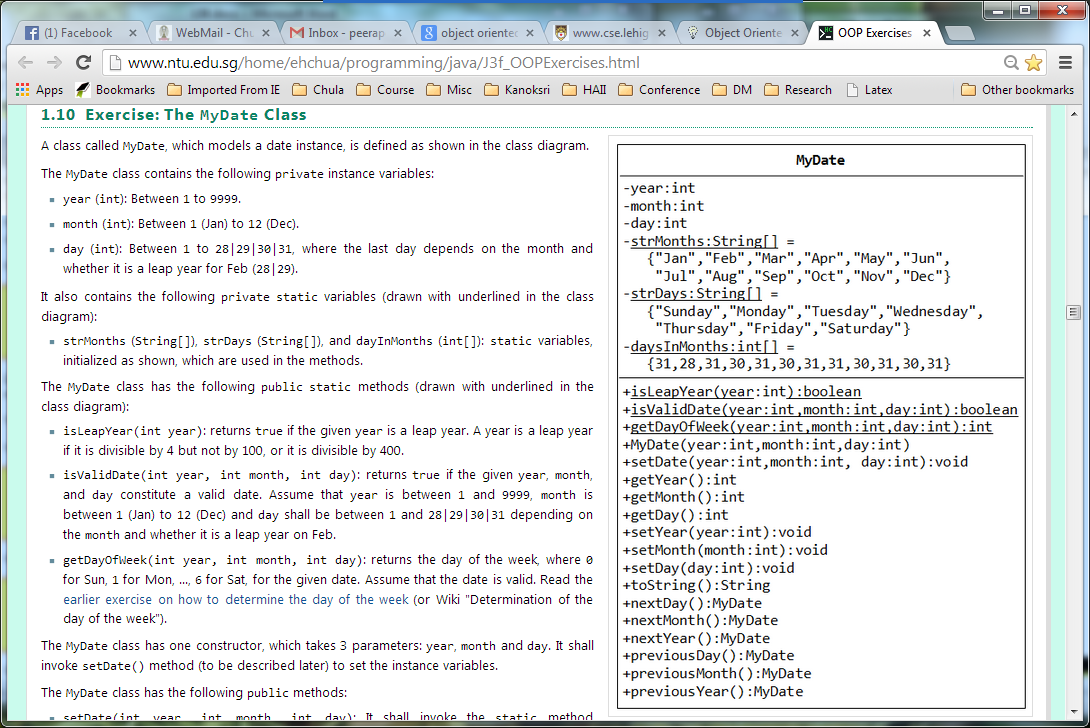
Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Part A: Getting Familiar with The MyDate Class[[1]](#footnote-1)

A class called MyDate, which models a date instance, is defined as shown in the class diagram.



The MyDate class contains the following private instance variables:

* year (int): Between 1 to 9999.
* month (int): Between 1 (Jan) to 12 (Dec).
* day (int): Between 1 to 28|29|30|31, where the last day depends on the month and whether it is a leap year for Feb (28|29).

It also contains the following **private static variables** (drawn with underlined in the class diagram):

* strMonths (String[]), strDays (String[]), and dayInMonths (int[]): static variables, initialized as shown, which are used in the methods.

The MyDate class has the following **public static methods** (drawn with underlined in the class diagram):

* isLeapYear(int year): returns true if the given year is a leap year. A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.
* isValidDate(int year, int month, int day): returns true if the given year, month, and day constitute a valid date. Assume that year is between 1 and 9999, month is between 1 (Jan) to 12 (Dec) and day shall be between 1 and 28|29|30|31 depending on the month and whether it is a leap year on Feb.
* getDayOfWeek(int year, int month, int day): returns the day of the week, where 0 for Sun, 1 for Mon, ..., 6 for Sat, for the given date. This method is provided in “DayOfWeek.java”

The followings are descriptions of **some public methods**:

* toString(): returns a date string in the format "xxxday d mmm yyyy", e.g., "Tuesday 14 Feb 2012".
* next/previousMonth(): must start from the 1st day of that month!
* next/previousYear(): must start from Jan 1 of that year!

## Part B: Questions about The MyDate Class

How many attributes and methods in the class?

Please give the code to create an object of “today” date

Please specify which of the followings are leap years?

* 2000, 2007, 2013, 2004, 2001, 2012

Are “1/15/2013” and “1/12/10000” a valid date? If not, why?

Use “DayOfWeek.java” to find out the day of week of “1/1/2014”?

What should the toString() method return if the input date is “1/1/2014”?

## Part C: Coding

Write the code for the MyDate class.

Use the following test statements to test the MyDate class:

MyDate d1 = new MyDate(2012, 2, 28);

System.out.println(d1); // Tuesday 28 Feb 2012

System.out.println(d1.nextDay()); // Wednesday 29 Feb 2012

System.out.println(d1.nextMonth()); // Thursday 1 Mar 2012 – must be “Day 1st”

System.out.println(d1.nextYear()); // Tuesday 1 Jan 2013 – must be “Jan 1”

MyDate d2 = new MyDate(2012, 1, 2);

System.out.println(d2); // Monday 2 Jan 2012

System.out.println(d2.previousDay()); // Sunday 1 Jan 2012

System.out.println(d2.previousMonth()); // Thursday 1 Dec 2011

System.out.println(d2.previousYear()); // Saturday 1 Jan 2011

MyDate d3 = new MyDate(2012, 2, 29);

System.out.println(d3.previousYear()); // Saturday 1 Jan 2011

// MyDate d4 = new MyDate(2099, 11, 31); // Invalid year, month, or day!

// MyDate d5 = new MyDate(2011, 2, 29); // Invalid year, month, or day!

Include the screenshots below.

List all your source code here.

Submit this worksheet (by only one member of the group) via <http://www.myCourseVille.com> (Assignments > Hands-on Experiment # 11) **within the day after your lecture**.

1. This exercise is updated from “Java Programming Tutorial OOP Exercises” at  
   <http://www.ntu.edu.sg/home/ehchua/programming/java/J3f_OOPExercises.html> [↑](#footnote-ref-1)