

|  |
| --- |
| **COMP 2120 – Programming in Java**  Seminar #8 |

## Due Date: Beginning of your lab on November 13, 2012

**Problem:**

**The purpose of this assignment is to gain experience solving user input problems by using exception handling in your Java code.**

**(Read this carefully and make certain that your program implements all parts *exactly* as described. Marks will be lost otherwise.)**

Write a program that reads a string from the keyboard and tests whether it contains a valid date. Display the date as described below if it is valid, otherwise display a message as described below. **The input date should have the format mm/dd/yyyy, where mm and dd can be one or two digits.**

The program solution should follow good programming development principles and be coded as a number of methods.

Here are the input errors (exceptions) that your program should detect and deal with:

Receive the input from the user (give an example of the format you are expecting the user to use when entering the date) and check for the following:

* 1. missing one of parts (that is the month or the day or the year is missing. Check this by checking the number of slashes (/)), eg mm/dd is invalid
  2. mm or dd or yyyy is not numeric
  3. month not between 1 and 12
  4. day is not in the right range (this depends on the month)
  5. year doesn’t have 4 digits

**Be specific when reporting an error. For example, “The input 13/4/2000 contains an invalid month of 13” or “February doesn’t have 30 days”, etc.**

**If the user gives you invalid input, start the whole process again by asking the user for a valid date. Keep prompting the user until you get a valid date. A while (!valid) loop would work well here.**

**If the date is valid, then output a message in this format, “December 25, 1940 is a valid date”.**

One of the checks should be for leap years. February has 28 days except for leap years when it has 29. A leap year is any year divisible by 4 but not divisible by 100 unless it is also divisible by 400.

**Notes:**

1. String class methods (indexOf method) can be used to find the slashes (/). Strings can be converted to integers by using the Integer.parseInt( ) method.
2. **Be sure that the output that you submit shows that your program has tested ALL the different types of errors that can occur.**
3. **Check for 6 types of exceptions**:
   * SlashException – checks to see if the input string is missing one of the slashes (/), which separates the month and day, and the day and year
   * NumberFormatException – if the input string has the correct format, ie mm/dd/yyyy, then determine if the input consists of numbers only (ie, no non-numeric characters other than /). The easiest way to do this is to use the parseInt method from the Integer class, ie Integer.parseInt(mm) or parseInt(dd). parseInt( ) will throw a NumberFormatException if mm or dd can’t be converted to an integer
   * MonthException: month not between 1 and 12
   * YearException: year is not between 1900 and 2012
   * MonthDayException: month-day combinations is not ok, for example check that February doesn’t have 30 days, etc.
   * You may structure your solution however you wish but at least one method must throw at least 2 exceptions

* The messages output by the catch blocks should be very specific. For example, output “123 is an invalid month”, or “abc is an invalid month”, rather than simply “invalid month”. The first 2 messages allow the user to better understand what input caused the problem; the last message leaves the user guessing about what the source of the problem is.
* **You do not need to write code for the NumberFormatException class; this already exists as part of java.**
* **You’ll need to write class definition files for the other exceptions described above.**

**Assignment Submission:**

Hand-in a print-out of each class definition’s source code, the driver program source code and the output.