Programming Exercises - PRO1 - Session 07

Exercise 7.01

Write a program that asks a user to enter a number within the range 1-10. The program should then display the Roman numeral version of the entered number. If the number is outside the range 1-10, the program should display an error message of your choice. The first ten Roman numerals are {I, II, III, IV, V, VI, VII, VIII, IX, X}.

Exercise 7.02

Further modify your MyDate class from the last session by adding a method `daysInMonth()` that returns the number of days in the month (e.g. for January it should return 31).

Remember: If year is a leap year February has 29 days (unlike the usual 28 days), so you can use the `isLeapYear()` method you made previously to calculate the number of days properly.

Exercise 7.03

Add yet another method `getAstroSign()` to the class MyDate that will return a String with the name of the astrological sign corresponding to the date.

The 12 astrological signs are:

Aries – March 21 - April 19 (both days included)
Taurus – for April 20 - May 20 (both days included)
Gemini – May 21 - June 20 (both days included)
Cancer – June 21 - July 22 (both days included)
Leo – July 23 - August 22 (both days included)
Virgo – August 23 - September 22 (both days included)
Libra – September 23 - October 22 (both days included)
Scorpio – October 23 - November 21 (both days included)
Sagittarius – November 22 - December 21 (both days included)
Capricorn – December 22 - January 19 (both days included)
Aquarius – January 20 - February 18 (both days included)
Pisces – February 19 - March 20 (both days included)

Exercise 7.04

Write a program that asks the user to enter a username, a password and a confirmation password. Then print out the username, and a message about whether or not the entered password and confirmation password are identical.

Exercise 7.05

Given a day, month and year, it is possible to calculate which day of the week (Monday, Tuesday, Wednesday, etc.) that date is. The equation for doing that is slightly complicated, but if you read the details carefully, and perhaps break it up into smaller parts, then it's not impossible to calculate. So add one more method to the MyDate class called `dayOfWeek()` that returns a String with the name of the day.

The equation looks like this:

$$h = \left(q + \frac{13(m+1)}{5} + k + \frac{k}{4} + \frac{j}{4} + 5j \right) \% 7$$

- `q` is the day of the month (1 to 31)
- `m` is the month, but in a special format:
  - March=3, April=4, ..., November=11, December=12, January=13, February=14
- `k` is the year of the century (you can find that as `year%100`)
- `j` is the century (you can find that as integer division of `year/100`)
- `h` is the day of the week that we are looking for, but the result will be in a slightly special format:
  - 0=Saturday, 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday

Note: January and February are actually counted as months 13 and 14 of the previous year. E.g. February 23, 2024, should be counted as the 23rd day of the 14th month of 2023. So, when the month is January or February, you will also have to subtract 1 from the year used in the calculation.

Now you can try out this new method in a test class, to find the day of the week for any date in any year. So, you could e.g. find out which day of the week you were born.