

CS 115
Assignment 3
Dr Malek Mouhoub

Project 1

This exercise is taken from Project 10 page 485 (4th and 5th Editions).

Write an interactive program that plays the game of Hangman. Read the word to be guessed into `word`. The player must guess the letters belonging to `word`. The program should terminate when either all letters have been guessed correctly (player wins) or a specified number of incorrect guesses have been made (computer wins). Hint: Use `solution` to keep track of the solution so far. Initialize `solution` to a string of symbols '*'. Each time a letter in `word` is guessed, replace the corresponding '*' in `solution` with that letter. A sample run follows.

```
% assign3p1
Enter the word to be guessed: alpha
Enter the max number of tries: 10
Guess a letter (you have 10 tries left): a
Right! Word so far: a****
Guess a letter (you have 9 tries left): +
      ERROR: '+' is not a letter.
Guess a letter (you have 9 tries left): x
Wrong! Try again. Word so far is: a****
Guess a letter (you have 8 tries left): l
Right! Word so far: al***
Guess a letter (you have 7 tries left): a
      'a' has already been used. Try again.
Guess a letter (you have 7 tries left): u
Wrong! Try again. Word so far is: al***
Guess a letter (you have 6 tries left): p
Right! Word so far: alp*a
Guess a letter (you have 5 tries left): h
You Win! Good guesser!
```

```
% assign3p1

Enter the word to be guessed: dog
Enter the max number of tries: 4
Guess a letter (you have 4 tries left): o
Right! Word so far: *o*
Guess a letter (you have 3 tries left): a
Wrong! Try again. Word so far is: *o*
Guess a letter (you have 2 tries left): b
Wrong! Try again. Word so far is: *o*
Guess a letter (you have 1 try left): x
You loose !
```

Project 2

Develop a class `Date` for representing a calendar. The class should provide a default constructor that initializes the date to January 1, 1900. Another constructor should initialize a `Date` object to a specific value using three integer parameters corresponding to the desired month, day and year. Define the function `nextDate` that returns the successive date, the new value of the `Date` object. You should take into account if a year is a leap year or not. For example, applying `nextDate` on the date 12-31-2000 provides a new date 01-01-2001. Also define the function `ToString` which returns a string version of a `Date` object. For example, applying `ToString` on the date 12-01-2000 returns *December 1st, 2000*. Write a driver program to test your class.

A leap year is:

divisible by 400

or

divisible by 4 and not divisible by 100.

Examples :

- 1600 is a leap year because 1600 is divisible by 400
- 1988 is a leap year because 1988 is divisible by 4 and not by 100.

A simple run of the driver program follows.

```
Enter a new date using the format mm-dd-yyyy: 12-31-2000
The string version of the date is: December 31st, 2000
The next date in string version is: January 1st, 2001
Do you want to continue [Y/N]: Y
Enter a new date using the format mm-dd-yyyy: 13-13-2000
Error: Month must be greater than zero and not larger than 12
Do you want to continue [Y/N]: y
Enter a new date using the format mm-dd-yyyy: Jan 12 2006
Error: Wrong format
Do you want to continue [Y/N]: Y
Enter a new date using the format mm-dd-yyyy: 02-28-2004
The string version of the date is: February 28th, 2004
The next date in string version is: February 29th, 2004
Do you want to continue [Y/N]: y
Enter a new date using the format mm-dd-yyyy: 02-28-2005
The string version of the date is: February 28th, 2005
The next date in string version is: March 1st, 2005
Do you want to continue [Y/N]: n
Good bye !
```

Hand In

1. Name the file containing the program of the project 1 “`assign3p1.cpp`”. The header, implementation and driver program of project 2 should be respectively named: `MyCalendar.h`, `MyCalendar.cpp` and `TestMyCalendar.cpp`. Your C++ program **SHOULD** compile using CC (Sun compiler) under Hercules.
2. Submit all the above files using WebCT: www.uregina.ca/webct. You will then receive an acknowledgement email confirming your submission. You should save this email as a proof of submission. If you do not receive an email acknowledging your submission then you should promptly email the marker (mark115@cs.uregina.ca) with your submission in attachment.

Marking scheme for each project : total = 50% + 5% (Bonus)

1. Readability (program style) : 10%
 - Program easy to read,
 - well commented,
 - good structured (layout, indentation, whitespace, ...) and designed (following the top-down approach)
2. Compiling and execution process : 5%
 - program compiles (with CC under hercules) w/o errors and warnings
 - robustness : execution w/o run time errors
3. Correctness : 35%
 - code produces correct results (output).
 - **output meets the initial requirements (see above for the output format).**
4. Bonus : 5%
 - Features that increase functionality and/or presentation.