

Lab 04 - Selection Statements & Iterations

Instructions:

- The lab requires writing a complete cpp file within an hour. It requires completing 5 tasks.
- Accompanying this file is a template cpp file that you must modify. You cannot include additional libraries to or remove any libraries from the file. All other modifications are allowed.
- Your submission must be submitted to the Labs directory of your github repository and/or as an attachment on Google classroom under the Lab04 assessment. The file must remain a cpp file.
- Cheating of any kind is prohibited and will not be tolerated.
- Violating and/or failing to follow any of the rules will result in an automatic zero (0) for the lab.

TO ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS ABOVE, AT THE BEGINNING OF YOUR SUBMISSION, ADD A COMMENT THAT CONSISTS OF YOUR NAME AND THE DATE

Your objective is to write a complete program that reads in a month and day for the year 2003, and then, displays the date in the format

WeekDay, *Month Day*, 2003

where *WeekDay* and *Month* are represented as capitalized words. For instance, if the provided values for month and day are 2 and 19 respectively, the program will display

Wednesday, February 19, 2003

To accomplish the objective, complete the following tasks:

- ☐ define a string function named **MonthName()** that takes an int parameter. If the value of the parameter is between 1 and 12 inclusively, the function returns the capitalized name of the month whose position in the calendar corresponds to the value of the parameter. Otherwise, it returns an empty string.
- ☐ define a bool function named **ValidDate()** that takes two int parameters. It returns true if the value of the first parameter is a valid numerical representation for a month [1 - 12] and the value of the second parameter is a valid day in the month represented by the first parameter. Otherwise, it returns false.
- ☐ define a void function named **RetrieveDate()** that takes two int reference parameters. It continually prompts the user to enter a month (numerical representation) and day separately, and then, it stores the inputs in the parameters until the month and day are both valid [they represent an actually date in the year 2003].
- ☐ define a string function named **WeekDayName()** that takes two int parameters. Given that the first parameter and second parameter represents the month and day of a date in the year 2003 respectively, it returns the week day that correspond to the date as a capitalized word. To determine the week day of any date in a year, use the formula

$$\text{WeekDay} = (\text{FirstDay} + \text{DaysBeforeDate}) \% 7$$

where *WeekDay* is represented numerically with the following correspondence

Week Day	"Sunday"	"Monday"	"Tuesday"	"Wednesday"	"Thursday"	"Friday"	"Saturday"
Value	0	1	2	3	4	5	6

and *FirstDay* is the numerically representation of January 1st of the year and *DaysBeforeDate* is the number of day before the specified date. For instance, in 2003, January 1st falls on Wednesday (1) and there are 49 days before February 19th; hence, the formula determines that the week day for the 19th of February is

$$\begin{aligned}\text{WeekDay} &= (1 + 49) \% 7 \\ &= 50 \% 7 \\ &= 1 \\ &= \text{"Wednesday"}\end{aligned}$$

- ☐ define a void function named **DisplayDate()** that takes no parameters. It retrieves the month and day of a date in the year 2003 from the user and displays the date in the format mentioned above.