**CIS 2275 C++ Programming Part II**

# Program 1 Calendar

# Review arrays, vectors, file I/O, functions, passing variables using pointers/references

**Turn In Requirements:**

1. **5 pts Name your Visual C++ 2019 project LastnameP1, such as NelsonP1.**
2. **5 pts Upload to Blackboard, remove the intermediate files/ folders from your project before uploading it to Blackboard.**

# 

**Program Requirements:**

1. **3 pts Write your name, email address and file name at the top of your source code in a comment.**
2. **5 pts Your main function should have cout statements that write “header” information to the screen. The header info includes your name, course and program information, as well as a 1-2 line description of the program.**
3. **5 pts. Use good C++ programming style and formatting for your program. Use appropriate comments to explain what you are doing.**

The C++ Corporation has expanded to printing monthly calendars. As you know, monthly calendars are usually a grid of 6 rows by 7 columns, filled with numbers representing the month’s days. The day of the week is above each column, beginning with Sun, ending with Sat. Across the top of the grid is the name of the month and the year.

You will write a C++ program that creates, fills and writes a file showing a monthly calendar with appropriate dates and days of the week. In main, declare a monthGrid integer array sized 6 rows by 7 columns and a string for the month name. Fill a string array with the names of the week, beginning with “Sun” and ending with “Sat.” Fill a string array with the names of the months, along with an integer array holding the total number of days for each of the 12 months, That is, monthName[0] is January and monthDays[0] is 31.

You will have three files: the Driver, which will contain your main function, and Calendar.h, containing your function prototypes, and Calendar.cpp, containing function bodies. Be sure to use include guards around your prototypes.

The functions needed for this program are listed below. Pay careful attention to the specifications for each function. All functions are called from main.

|  |  |  |
| --- | --- | --- |
| Function Name | Return Type | Description |
| Header | string | Returns a string to write your name, program title, and program objective to the screen so that it is the first thing displayed when your program runs. It may be re-used elsewhere. |
| AskForMonth | void | The user selects the month to be displayed. This should be the conventional integer, i.e., January is 1 and February is 2, etc. Then ask for the year. Both variables will be references. |
| FillMonthGrid | int | Fills the dates into their appropriate grid position. Contains the code to determine if February is in a leap year. Also contains the algorithm that calculates the day of the week, so it will know which day to start the month’s calendar. Returns the number of days in the month. |
| CreateString | string | Constructs a nicely formatted string for the Month to be displayed. |
| WriteMonth | bool | Writes the results to a file. Returns true/ false saying if the file is opened. The filename is created in this function, named MonthNameYear.txt, such as “September2001.txt”. This file should be a neatly laid out month as well. |

After calling the Header function, main will open a do-while or while loop. Call AskForMonth to get the month and year. Make the earliest year be 1600. Nothing before that.

Then call FillMonthGrid to fill the dates into their appropriate grid position. This function contains the code to determine if February is in a leap year. **You will need to find an algorithm to calculate the day of the week so you can start your monthly calendar on the appropriate day for that year.**

Call CreateString to display the nicely formatted month in a cout to the user. This display should be a neatly laid out month including the name centered across the top of the page, the day names across the top of the columns and the individual date spaced along the rows.

Then call WriteMonth. WriteMonth constructs the filename from the MonthName and the year and “.txt”. For example, the filename will then be “September 1997.txt”.

In main, check if the file was opened and report the filename if the write was successful. If unsuccessful, report that also.

Ask the user if he/she wants to estimate any other operations. If yes, loop back up to AskForMonth. Otherwise, drop out of the loop.

As the program exits, write a good-bye message to the user.