

Programming Assignment #4

Learning Objectives:

Programming practice in the following areas:

- Writing functions that have parameters
- Call functions with parameters
- C++ strings
- C++ formatted output
- Repetition of instructions
- Moderately complex algorithms

Description:

Your main objective is to write a `printmonth` function that prints out an individual calendar month. Your program will have a `main` function that will call the function several times to test it. This is not a yearly calendar, i.e., it just prints out a month, regardless of the year.

Programming Specifications:

- The following is the prototype for function `printmonth`:

```
void printmonth( const int month, const int startday, const bool leapyear);
```

Description of Parameters:

const int month : Range (1..12) The numeric value for the month to be printed out. For example if the number 1 is passed in then January will be printed out, if 2 is passed in then February will be printed out. Etc.

const int startday: Range (0..6) This value will tell the routine which day of the week to start the 1st of the month on. The calendar will be laid out so that Sunday is the first day and Saturday the last. If the number 0 is passed in, the 1st will be on a Sunday, if 2 then the 1st of the month will be Monday, etc.

const bool leapyear: Range (true or false) This value will tell the routine to print out the Month of February with 29 days instead of 28.

Design Considerations:

- You must print out the alphabetic representation for the month (January, February, March, etc) instead of the numeric value passed in. One way to easily map the numeric to the alphabetic month is to have an array of strings. Indexing into the array with the numeric month can then be used to obtain the string representing the month of the year. Let's say that you called the string `MonthArray`. Then, `MonthArray[1]` is "January", `MonthArray[2]` is "February", etc.
- You may also store the number of days in the month in the same manner using an array of integers. The index of [1] would contain the number 31 to represent 31 days in the month of

CGS3406 Introduction to Programming using C++

January, the index of [2] would have the value of 28, etc. The February value would be changed to 29 if it was a leap year.

- **You must have a printmonth function with parameters as described previously.** The function will likely need the following steps:
 - Check if Leap Year is set and modify the number of days in February to 29 instead of 28.
 - Print out the Alphabetic Month
 - Print out the days of the week using the three letter abbreviation
 - Print out the first week starting the 1st on the appropriate day of the week.
 - Print out the rest of the month
 - Print out a new line (**endl**)
- **main** function. This will be your main routine, expected output corresponding to the execution of the below code is at the end of this assignment description.

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

void printmonth(const int month, const int startday, const bool leap)
{
    //*****
    // * Put your code here.
    //*****
}

int main(void)
{
    // *****
    // * Print a few arbitrary months
    // *****

    printmonth(1,0,false); // Print January 1st, on a Sunday
    printmonth(2,1,true);  // Print February 1st leap year, on Monday
    printmonth(1,2,false); // Print January 1st, on a Tuesday
    printmonth(2,3,false); // Print February 1st not leap, on Wednesday
    printmonth(1,4,false); // Print January 1st, on a Thursday
    printmonth(2,5,false); // Print February 1st, on a Friday
    printmonth(1,6,false); // Print January 1st, on a Saturday
    printmonth(6,1,false); // Print June 1st, on Monday
    printmonth(12,4,false); // Print December 12th, on a Thursday

    return 0;
}
```

Required Algorithms:

- You must figure out how to start printing with the 1st of the month on the appropriate day. As an example, if the 1st starts on a Wednesday you must leave Sunday, Monday, and Tuesday blank on the first week of the month.

CGS3406 Introduction to Programming using C++

- You can use `setw()` in formatting your output. Be sure to `#include <iomanip>` in your program.
- You must figure out how to proceed with printing the remainder of the month and be sure to start printing on the next line at the end of the week (i.e. when you print the day of the week for Saturday you start the next number on Sunday).

Grading Criteria:

- The program compiles. If the program does not compile no further grading can be accomplished. Programs that do not compile will receive a zero.
- (25 Points) The program executes without exception and produces output. The grading of the output cannot be accomplished unless the program executes.
- (25 Points) The program produces the correct output. Note that values other than those listed in the above sample code will be tested.
- (25 Points) The program specifications are followed.
- (10 Points) The program is documented (e.g., commented) properly. The `printmonth` function should be commented describing the inputs (i.e., parameters) and operation (e.g., strings written to standard output).
- (5 Points) Use constants when values are not to be changed.
- (5 Points) Use indentation to improve code readability.
- (5 Points) Use descriptive identifier names.

Sample Input:

- There is no input (i.e., from the keyboard) to this program. The program should execute and upon completion, exit.

Sample Output:

The sample output corresponds to the main routine provided in this assignment.

```
          January
Sun Mon Tue Wed Thu Fri Sat
  1   2   3   4   5   6   7
  8   9  10  11  12  13  14
 15  16  17  18  19  20  21
 22  23  24  25  26  27  28
 29  30  31
```

```
          February
Sun Mon Tue Wed Thu Fri Sat
      1   2   3   4   5   6
  7   8   9  10  11  12  13
 14  15  16  17  18  19  20
 21  22  23  24  25  26  27
 28  29
```

```
          January
Sun Mon Tue Wed Thu Fri Sat
      1   2   3   4   5
  6   7   8   9  10  11  12
 13  14  15  16  17  18  19
```

CGS3406 Introduction to Programming using C++

20 21 22 23 24 25 26
27 28 29 30 31

February

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

January

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

February

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

January

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

June

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

December

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31