

Spring 2017 CMSC140 Programming Project 1: Weekly Schedule

Concepts tested in this project

- To work with cout Object
- To work with Variables and Literals
- To work with constants
- To learn and use different Data Type
- To learn and use Programming Style
- To learn and use Arithmetic Operators

Project Description

Create a program that displays your weekly schedule. The schedule should be displayed in a grid format, with the days of the week being the columns and the different times during the day being the rows. Display your classes on the correct days and the correct times during those days. Also indicate at what times you work during the week, if you do.

The program should also display, at the bottom, the total number of hours and minutes you spend at school, at work, and the estimated number of hours and minutes you feel you will have time to study for your classes. The total number of hours and minutes should be calculated in the program using the sum of the number of minutes you are at college, work, or studying. Each class time for each day should have its own variable to hold the amount of time at that class for that day. Have one variable for the total number of minutes spent working, and one for the total number of minutes spent studying.

For example, if you spend three hours in total for one class, one hour each on Monday, Wednesday, and Friday, each day should have its own variable to hold the number of minutes the class lasts (which is 60). The sum of the time should also be minutes (e.g. 180). Then, at the end of the program, you will calculate the number of hours you spend at class in total and the leftover number of minutes. (Assuming the example is the only class, the total time would be 3 hours and 0 minutes).

Write, compile and run a C++ program that displays your schedule per week in grid style and that calculates the total number of hours and minutes you spend at class, working, and studying. **Refer to the screen shot of the sample output for more details.**

- Create appropriate variables to hold the following data:

- Name (yours)
- The number of minutes for each class (separate variables for each day as well)
- A constant variable for the number of minutes in an hour (60)
- The sum of all the minutes of your classes
- The number of minutes you spend at work per week
- The number of minutes you spend studying per week (doesn't have to be exact)
- The total number of hours spent in class, and the total number of minutes
- The total number of hours spent studying and working, and the total number of minutes for each.

You may need to create more variables to hold data in order to operate on them later.

Spring 2017 CMSC140 Programming Project 1: Weekly Schedule

Following is a sample run of the program:

```
C:\WINDOWS\system32\cmd.exe

-----
| John Smith | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday |
-----
| 9:00 - 10:00 | MATH 181 | CMSC 140 | MATH 181 | CMSC 140 | MATH 181 |
-----
| 10:00 - 11:00 | ENGL 101 |          | ENGL 101 |          |          |
-----
| 11:00 - 12:00 | ENGL 101 | Work    | ENGL 101 | Work    |          |
-----
| 12:00 - 1:00  | BIOL 101 | Work    | BIOL 101 | Work    | BIOL 101 | Work    |
-----
| 1:00 - 2:00   | BIOL 101 | Work    | BIOL 101 | Work    | BIOL 101 | Work    |
-----
| 2:00 - 3:00   |          |          |          |          |          | Work    |
-----
| 3:00 - 4:00   |          |          |          |          |          | Work    |
-----
| 4:00 - 5:00   |          |          |          |          |          |
-----
| 5:00 - 6:00   |          |          |          |          |          |
-----
| 6:00 - 7:00   |          |          |          |          | Work    |
-----
| 7:00 - 8:00   |          |          |          |          | Work    |
-----

Total Weekly School Time: 15 hour(s) and 5 minute(s).
Total Weekly Work Time: 11 hour(s) and 30 minute(s).
Total Weekly Study Time: 5 hour(s) and 30 minute(s).
Press any key to continue . . .
```

Note: Sunday should be included as well if you work or take classes on that day.

Project 1 Submission requirements:

Notes:

- Proper naming conventions: All constants, except 0 and 1, should be named. Constant names should be all upper-case, variable names should use "camel case" (i.e. start with lower case, with subsequent words starting with upper case: *hoursWorked* for example) or underscores to separate words (i.e. *items_ordered*) (textbook, page 42)
- Variable and method names should be descriptive of the role of the variable or method. Single letter names should be avoided.
- Documentation: The documentation requirement for all programming projects is one block comment at the top of the program containing the course name and CRN, the project number, your name, project description, the due date and platform/compiler that you used to develop the project. If you use any code or specific algorithms that you did not create, a reference to its source should be made in the appropriate comment block. Additional comments should be provided as necessary to clarify the program.
- Indentation: It must be consistent throughout the program and must reflect the control structure.
- **Program Header:** You should include one block comment (header) at the top of each program containing the course name and CRN, Instructor's name, the

Spring 2017 CMSC140 Programming Project 1: Weekly Schedule

project number, your name, the date and a short description of the project as follows:

```
/*
 * Class: CMSC140 CRN
 * Instructor:
 * Project [number] 1
 * Description: (Give a brief description for Project1)
 * Due Date:
 * I pledge that I have completed the programming assignment independently.
  I have not copied the code from a student or any source.
  I have not given my code to any student.
  Print your Name here: _____
 */
```

Deliverables:

1. A Word document that includes:
 - Title Page with the following information
 - Project <#>, Due date (including year) , Your name, class, and section
 - Screenshots of the program
 - Pseudocode for the program
2. Your source code (.cpp file). Your source code file should include a block comment (header) listed below.
3. The C++ files zipped and saved as LastNameFirstName_Project1_Moss.zip

This .zip will not have any folders in it – only .cpp files.

Note: This format is required to check for duplicate submissions using "MOSS" Plagiarism Detection Software.



Submit your completed assignment to **Blackboard** no later than the due date.

Grading Criteria for Project 1

This project will be graded using the following are components. If program does not compile, project will get grade "0". Contact your instructor prior to the project submission due date, if you have compilation issues.

Attributes	Value (points)
Functionality (If project does not compile, project will get grade "0")	Total 100
Displays the student's schedule appropriately formatted	30
Calculates and displays the total amount of hours and minutes spent at classes, work, and spent studying	30
Program executes correctly (produce expected output)	15
Meets all requirements	15
Overall Look-and-Feel	10

Spring 2017 CMSC140 Programming Project 1: Weekly Schedule

Total	Total 100 points
-------	------------------

Project General Requirements (points will be deducted)

Attributes	Value(points)
Programming Style and proper naming convention: (see coding standards)	(-20 pts maximum)
Constants not all caps	-5
Curt or unclear variable names	-5
Long variable names should use camel case or underscores to separate words	-5
Comments and internal notes	
Sparse and inadequate comments.	-5
File header is not included	-5
Essentially no comments	-10
Indentation and white spaces should be a visual aid to understanding code structure	
Indenting is mostly okay, but sometimes inconsistent.	-5
No indenting, or very inconsistent indenting that is a barrier to understanding the code	-10
Lack of white space separating variables and operators. Lack of white spaces separating functions and major code blocks (later projects only)	-5
Test Plan (not applicable for project 1)	(-20 pts maximum)
Missing Entirely	-20
Cursory or inadequate testing	-10
Adequate overall, but missing a few crucial tests	-5
Missing Required Items (only if required for the project)	(-20 pts maximum)
Pseudocode, Flowcharts, or Hierarchy chart missing	-20
Screen shots cursory or incomplete	-5
Screen shots completely missing	-10
Project writeup and reflection	
List of assumptions made (not applicable for Project 1)	-5
Highlights of your learning experience	-5
Awkward Code Internal Structure (not applicable for Project1)	(-10 pts maximum)
Hard-coding input values (not applicable for Project 1)	-5 each
Poor structured programming: inappropriate loop choices, incorrect use of break statements to exit loops, and so on (not applicable for Project 1)	-5 each
Excessive reliance on global variables (e.g., using them to avoid pass by reference) (not applicable for Project 1)	-5
Processing array contents piecemeal rather than using loops (not applicable for Project 1)	-5
Other poor coding practices not mentioned	-5