

Things You Need To Know for Project 1 and Final Project 2

(in addition to the separately supplied project specifications supplied by Olive S. Cargot)

Project 1 – Due 4/19/19

Project 1 should strive to complete the Nutrition Label portion of the software – Project 1 does not require any of the added Recipe software.

Shoot for only nutrition labels, multiple labels (you might want to reduce the number of fields (items listed) for this version – add a label, list label names and display /print to file the one selected, delete a label and Edit label. Suggest that you save the design of delete a label and edit a label for last.

Project 1 - What to submit:

- ***your .cpp code file***
- ***any associated files (input) you find necessary to have someone test your code.***
- ***a single document in which you include sections for:***
 - ***what work/ what doesn't work so the grader knows what to expect***
 - ***a user's manual***
 - ***a brief reflection on the process you went through to complete this project***
 - ***any things you would enhance, change, redo, etc... given time to do so***
 -

It is highly recommended to that you schedule a 1-1 meeting with your instructor to review your Project 1 code. This meeting will give you personal feedback and potential insights that may aid you in moving on to a solid Final Project.

Final Project – Due 5/11/19

Final Project should complete/correct Project 1 Nutrition Label code and integrate in the Recipe portion of the software and make it a cohesive product along with thorough testing of the code.

Final Project - What to submit:

- ***your .cpp code file***
- ***any associated files (input) you find necessary to have someone test your code.***
- ***a single document in which you include sections for:***
 - ***what work/ what doesn't work so the grader knows what to expect***
 - ***directions for proper use***
 - ***a brief reflection on the process you went through to complete this project***
 - ***any things you would enhance, change, redo, etc... given time to do so***
 - ***a thorough user's manual***

- *a thorough set of test cases – showing the full functionality of your software with screen snips illustrating the functioning of your code along with descriptive text.*
- *a listing of your source code*
- *any other related items you deem relevant (pseudocode, etc...)*

Early Completion Option for Final Project:

Should you fully complete all the requirements for your Final Project prior to 5/8/19 you may demo your fully functional code to the class and instructor in person (prior to 5/8/19) and reduce your submission to:

- *your .cpp code file*
- *any associated files (input) you find necessary to have someone test your code.*
- *a single document in which you include sections for:*
 - *a brief reflection on the process you went through to complete this project*
 - *any things you would enhance, change, redo, etc... given time to do so*

Information relevant for both Project 1 and the Final Project

Submit on Canvas by the due date/time indicated on Canvas calendar. No late projects will be accepted. It is recommended that after you submit your code that you download it from Canvas, build a project and make sure it compiles and runs as you expect it to – just a double check that you have submitted the file you meant to.

1. Your work must **compile and run under MSVS 2017 Community**. This is the IDE under which it will be tested and the one used in class.
2. This is an **individual project – not a group project**. You may consult with peers as to concepts and thoughts but all work submitted must be your own work. Do not share your code with others.
3. **Nothing will be accepted after the due dates/times** listed on the calendar.
4. In preparation for CS 202 work you **will not use the string data type** for the remainder of the semester. You will use c-strings instead (a char array).
5. **No global variables** are to be used. A grade of 0 will be given if used. You may use **global constants** for your array sizes. Ask if you want an example.

6. Use **only concepts/constructs found in the first 7 chapters (omitting vectors from CH 7)** of our text book and **no use global variables, exit, continue, goto and break** (other than as standard use in switch statements) statements.
7. only have **one return statement per function**.
8. The program must include a **minimum of 8 substantive functions**. The main() function is to be a driver function. A grade of 0 will be given for less than 8 functions. A substantive function is one that serves a useful function – quite often they require parameters/returned value(s).
9. The submitted program **must compile** regardless of its state of completion. A 0 will be given for a program that does not compile.
10. The code **must include comments**. Comments shall consist of
 - a. a **comment header at the start of each source code file**, (template supplied in Canvas)
 - b. a **comment header at the start of each function** definition, template supplied in Canvas)
 - c. brief, meaningful **comments throughout the code** that allow a human to follow it w/out reading it line-for-line. There will be a grade of 0 for no, or 50% loss of grade for meaningless, comments.

Remember: solve the problem before you try to start coding. Think about how you can achieve the different parts of the project and write out your pseudocode to check your logic. If you need practice with a concept/construct write some small programs to understand that prior to incorporating into a larger program. We'll spend time in class looking at how we can think about the program in a generalized fashion and how you might modularize your problem into smaller tasks. You want to develop your code incrementally – one part at a time – and test and debug with each addition – this way you'll have a functional program at each step of the way – use function stubs (in Ch 6 and talked about in class).