

## TODO List

*Professor Caleb Fowler*

*January 17, 2021*

### *Problem.*

Create a command line TODO list program. Prompt your client. The first character/symbol they enter will be the command. Follow that with a space. Finally, enter the todo list item. Example: "+ Study for Final" (don't enter the quotes). This causes Study for Final to be entered into the TODO list with today's date. You will need to have the following fields at a minimum: TODO itself, date added, and TODO Identification number, but you can add as many other fields as you wish.

### *Requirements.*

---

- No C libraries (no .h).
- No global variables or analogues of global variables.
- No standard template library anything.
- **Create a alphabetic menu to handle program options.**
- Don't forget to put a header on your source file.
- No compiler warnings.
- Only allow valid letters or symbols as input for this menu. Upper and lower case if appropriate. Re-prompt in event of error. Can you do this without displaying an actual menu?
- Style guide elements apply: comments, layout, Program Greeting, Source File Header, and variables etc. etc. In your program greeting function, **be sure to display the current date.** Get this from the system hardware.
- **Create at least 1 object.**
- **All objects must have a component testing method** which runs at least **2 diagnostic tests on that object.** Include this in main() right after the program greeting.

### *Specifications.*

---

- ☐ **// Specification C1 - Overload «**  
Overload the stream insertion operator to output a TODO.
- ☐ **//Specification C2 - Overload »**  
Overload the stream extraction operator to input a TODO.
- ☐ **// Specification C3 - Test TODO's**  
Generate at least 5 TODO's in your component testing method.

☐ // Specification C4 - TODO array

Put your TODO's in a dynamic array of TODO's

---

☐ // Specification B1 - + Symbol

Allow the user to enter tasks with a "+" symbol.

☐ // Specification B2 - ? Symbol

Allow the user to display all tasks with a ? symbol.

☐ // Specification B3 - - symbol

Allow the user to remove a task with a "-" symbol (use an ID number to remove the TODO). This doesn't necessarily mean you need to delete it immediately.

☐ Specification B4 - Persistence

Have your TODO list survive program termination by dumping the TODO's to disk when program ends. Load the data when the program first runs - if the file exists.

---

☐ // Specification A1 - Overload Copy Constructor

Overload the default copy constructor to handle your TODO's. Do this even if you don't have any pointers in your TODO object/struct. This is a great method to put in your component testing method.

☐ // Specification A2 - Overload Assignment Operator

To handle TODO assignment.

☐ // Specification A3 - System Date.

Pull the date for your TODO record directly from the system date method.

☐ // Specification A4 - Overload Constructor

Allow empty input for an add from the menu. However, empty adds trigger the regular constructor which creates a test record filled with obviously dummy data. When the client actually enters data during an add, trigger an overloaded constructor filling it with the data supplied by the client.

*Rubric.*

---

Element	Points
Meets all elements in requirements section above	25
Each of the 4 'C' Specifications	5 (20)
Bonus for all 'C' Specifications	5
Each of the 4 'B' Specifications	5 (20)
Bonus for all 'B' Specifications	5
Each of the 4 'A' Specifications	5 (20)
Bonus for all 'A' Specifications	5
Total Points	100

*Due Date.* [due\\_sunday.tex](#)

---

*Late Work.* [late\\_til\\_monday.tex](#)

---

*How to Turn in your Homework.* [submit\\_Sp21.tex](#)

---

*Style Guide* [flc\\_style\(advanced\).tex](#)

---

*Grep Trick.* [grep\\_trick.tex](#)

---

*Client System.* [client\\_sys.tex](#)

---

*Using the Work of Others.* [plagiarism.tex](#)

---

### *Educational Objectives.*

This is about operator overloading (Chapter 11).

### *Change log.*

1/17/21

Updated to new standards. Corrected a few typo's.

1/5/18

Eliminated the ONL anf GRD versions, just too much to maintain.  
Collapsing them into 1 assignment I can use for both classes. Moved  
more common text out to Common/