

Program; Assignment 2 (20 points)

Due Date: 22 October 2020

Write a program in C++ from page 905, question 14 in the textbook on composite/aggregate classes. The program should be tested with data that has no parking violation, parking violation by less than 60 minutes, parking violation by 60 minutes and parking violation greater than 60 minutes.

Procedure for handing in Program Assignments

A softcopy of the entire project should be placed in the *cpscstorage02* drive under a folder named **Assignments**. The program should be fully documented.

The program should have the following information at the top of the program as documentation:

- Your first name and last name at the top of the program.
- Your userID
- Name of the project
- Name of the file
- A header titled **Status**: indicate if the program executes **Correctly-Works Completely** (completely) or **Not Correctly-Does Not Work Correctly** (no long winded explanations, thank you)
- Starting date of the project.
- Ending date of the project.
- A description of the program. The description should be a summary of the problem solved by the program.
- Type in the following text after that: “This program was done entirely by me and no part of this program was plagiarized, intentionally or unintentionally, from anybody. I would be held accountable and penalized if any part of this program was plagiarized.

- Follow the language style and format for declaring variable, constants, classes and instances.
- All constants should be declared using `const`.
- There should be no magic numbers in the program.
- Program should be fully documented
- Place a copy of the **entire project** in the *cpscstorage02* drive in a folder named **Assignments**. I should be able to run the program from the *cpscstorage02* drive without having to make any changes. That is, you need to ensure that all the libraries and links to files are set up the directory. If I cannot run it from the *cpscstorage02* drive, the assignment will be considered as incomplete.
- **Late Assignments will receive 0 points.**

Programming Rubric

Faculty: Dr. Sam R. Thangiah

Student: _____ **(Fill in your name)**

Trait					Score
Algorithm Implementation Following Directions (10 points)	An incomplete solution is implemented on the required platform, and uses the compiler, specified. It does not compile and/or run.	A completed solution is implemented and tested on the required platform, and uses the compiler, specified. It compiles and run, but has logical errors.	A completed solution is implemented and tested on the required platform, and uses the compiler, specified. It compiles and runs without errors, but does not meet the specifications and/or work for all test data.	A completed solution is implemented and tested on the required platform, and uses the compiler, specified. It compiles and run without errors. It meets the requirements and works for all test data.	
Program Design (5 points)	The program design does not meet the specification and does not use appropriate control structures and data structures and/or objects. Methods are not well-designed.	The program minimally meets the specification. Not all of the selected control structures, data structures and/or objects are appropriate. Methods are modular.	The program design generally meets the specification, is efficient, and uses appropriate control structures, data structures and/or objects. Methods exhibit good design and are themselves modular.	The program design fully meets the specification, is efficient, reliable and uses appropriate control structures, data structures and/or objects. Methods exhibit good design and are themselves modular.	
User Interface	User interaction is incomplete and does not meet specifications.	User interaction minimally meets the specification. The organization and/or content of the output and prompts do not increase the usability of the program.	User interaction generally meets the specification. The organization and content of the output and prompts is acceptable to the user. User messages for data checking and error are generally acceptable.	User interaction is as specified. The organization and content of the output and prompts is natural to the program. Data checking and error handling is explicit in the user dialog.	
Program Documentation (5 points)	No program documentation	Program is minimally documented.	Some required documentation is missing.	All required documentation is present (description of methods, objects and code segments as necessary)	

--	--	--	--	--	--