ENGG1340 Computer Programming
Course Project

Instruction

Form a group of two students on Moodle and pick an idea to work on by Mar 15. Read carefully the instructions regarding project idea, coding and submission requirements. Each team will be assigned later on a fellow student TA for close discussions or consultation. You are always welcome to consult the TAs and instructors for help too.

In this document, we have provided five project ideas with statements and objectives for your reference. Please note that you have the flexibility in redefining the goals and make reasonable assumptions. Other than the five project ideas, you can also work on your own idea different from the five, as long as your final program comprises the elements as set out in the code requirement below. If you decide to work on your own idea, please write the project statements and objectives of your idea in the readme.md carefully. Your code will be marked against your statements and objectives.

Plagiarism is strictly prohibited. Using Github's global searching function, we will do plagiarism checking for every project. Copied or merely reformatted code can be easily detected. If you have used a piece of code from other Github repositories (including your own repo), please cite accordingly and responsibly.

Code Requirement

Your program should involve the following coding elements:

- Dynamic memory management
- File input/output
- Data manipulation such as sorting, searching, adding/editing/deleting data records
- Program codes in multiple files
- Proper indentation and naming styles
- In-code documentation

Submission

All projects should be committed to a public Github repo. Each student should contribute to at least 25% lines of the code of their project, gauged by GitHub's "Contributors Graph" of each project. Commit comment should not be empty and should be written sensibly. For each function, comments on "what it does", "what are the inputs" and "what are the outputs" are needed. Only C standard libraries and C++ STL (not including Boost) are allowed. No library that directly covers the functionality of a project is allowed. If you are not sure about whether a library could be used or not, please consult the instructors or TAs.

Stage 1 Submission - Deadline: Mar 29

Set up a public Github repo and create a readme.md file there. The readme.md file for stage 1 submission should contain the problem statement and problem setting. You will need to write your own problem statement and problem setting even if you make use of the five project ideas provided below. This basically will define the scope of your project. Write down also the program features that you would like to implement.

*** All you need to hand in a link to the repo.**

Reference:

Getting started with writing and formatting on GitHub

About READMEs

Stage 2 Submission - Deadline: Apr 27

Your Github repo should contain:

- A readme.md detailing the problem statement (i.e., what problem to solve), problem setting (e.g., reasonable assumptions), functionalities and features, input/output specifications, compilation and execution instructions. Simply put, this serves like a manual to your program
- Makefile
- Source files (.h / .cpp / .c)
- Sample input/output files (whenever appropriate)

*** Again, all you need to hand in a link to the repo.**

Grading

- Problem statement (including setting & assumptions) 10%
- Implementation (including the use of dynamic memory management, file I/O, data manipulations) 40%
- Program modularity (proper use of functions) 10%
- Program functionality and special features 20%
- Documentation (including readme.md) and coding style 20%

Project Ideas

You can pick any of the ideas below to start with. You are encouraged to further refine the problem statement and setting. In case you want to know more of the ideas below, please talk to the respective TA for each project.

* For questions regarding self-proposed ideas, please contact Tim.

1. Table management system (Tingxiang)

Good restaurants usually run a system to manage the occupancy, reservation, and billing of their tables. In this idea, we target to build a small table management system to manage the few tables of

different sizes in your family-owned deli. Minimally, the system shall include basic features including 1) check whether a table is available or not; 2) occupy a table; 3) release a table; 4) given the number of a group of customers, suggest a vacant table with minimal enough seats; and 5) notify the waiter for tables occupied for too long (say 2 hours). Since your deli has a customer base that changes seasonally, your system should allow changes to the number of tables and the size of the tables. Imagine putting the system to production and please think of a function not aforementioned, and add the function to your system.

2. Commodity inventory system (Marco)

Inventory management is important for renowned brands to control and manage the ordering and storage of inventory. In order to optimize the management process, an inventory system is usually implemented. In this idea, we aim to build a commodity inventory system that allows staff to manage and record commodity inventory in different retail shops for monitoring inventory status, providing necessary data for procurements and delivering alerts when exceptional events happened (e.g. some commodity becomes out-of-stock). Minimally, the system shall include basic features: 1) search commodity according to different filters (e.g. in-stock/out-of-stock); 2) insert new commodity with basic information (name, manufacturer, amount, price, availability of shop(s)) after each procurement, 3) delete obsolete commodity; 4) update commodity information; 5) automatic change of inventory status according to the amount of commodity that is currently available (e.g. when the amount reaches zero, the inventory status would become "out-of-stock"); 6) deliver alerts when the commodity is "out-of-stock". Imagine putting the system into reality and please suggests a function that is not mentioned above, to be a function of your system.

3. Cash Register system (Lavender)

Build a system that will shorten the grocery store waiting line for cashiers and maximize the profit. There is one line of customers with grocery waiting in line for the cashiers. The more the grocery one customer has, the time it takes to spend at the register will be longer. If the waiting time is longer than 15 minutes, the customers will leave without buying anything. Opening a new register costs an activation fee. Known variables about the grocery include grocery types, price and amount. Minimally, the system shall include basic features including 1) be able to read from grocery file including information such as grocery and its price; 2) be able to randomly generate a customer (at least 50 customers for testing) with random grocery; 3) be able to calculate the processing time of one customer spent in the cashier; 4) the system is able to sort customers into cashier; 5) (optional) express line for grocery amount under 5; 6) profit should be positive at the end of testing. Imagine how a real grocery store will work with a long line of customer and how to optimize the number of cashier available due to the number of customers.

4. Accounting system (Shumin)

Learn how to manage our financial status is an important lesson for all of us. More and more people choose to use digital tools to manage their wealth efficiently. In this idea, we target to build an accounting system to help people keep track of their income and expense. Minimally, the system shall include basic functions including 1) add records of income and expenses; 2) the records should

contain some basic information, like amount, date, types of income or expense (e.g. food, game, salary, etc.), account (e.g cash, bank card, credit card, etc.); 3) the records could be deleted and edited at any time; 4) users could view their records by date, type, and account; 5) the system should provide statistical report of users' financial state (e.g. monthly income and expenses, percentage of food expenses, etc.); and 6) the accounting system allows budget setting. When expenses reach the budget, there should be an alert from the system. Add a function you think is useful to the system but not previously mentioned.

5. Staff Management System (Qiuhui)

An employee management system uses a series of methods to gather and manage information about the employees of a business. In this idea, we target to develop a system to collect and manage the employee information of a small business. Minimally, the system shall include basic features including 1) create a new employee with employee ID, name, age, role, and salary; 2) search for and delete an employee via their ID, name, age or role; 3) fire an employee; 4) edit the details of an existing employee; 5) search for all employees with a salary higher than or lower than a user input; and 6) allow adding user-defined attributes, the value of a new attribute of the existing employee should be empty or undefined. A menu should be required to help the users realize function 2-4. Imagine building the management system for production. Please think of a useful function that is not mentioned before, add it to your system.