THE UNIVERSITY OF BRITISH COLUMBIA

Department of Electrical & Computer Engineering

CPEN 333 – Software Systems Engineering

Assignment 1 – Sept 2018

Introduction:

The 'Chevron' chain of gas stations is planning to build a new gas station on the University campus. Your task is to provide a real-time simulation of this, the specification of which is given below. The architecture for the software is given in a separate document on the course web site. You should read this document in conjunction with the architecture document otherwise it won't make much sense.

1. There are 4 pumps on the forecourt enabling 4 vehicles to be filled up simultaneously.
2. A display present inside the gas station office shows the real-time status of all pumps.
3. Drivers have to pay in advance via a credit card before gas is dispensed.
4. Drivers get to choose the grade of fuel they wish to purchase: There are 4 types based on the Octane rating of the fuel (e.g. 87, 91 etc), the higher the rating, the more the fuel costs.
5. For each customer, you will have to **simulate**:
    a. Arrival at the pump (you may have to wait in line)
    b. Swiping a credit card.
    c. Removing the gas hose from the pump.
    d. Selecting a grade of fuel.
    e. Dispensing fuel.
    f. Returning the hose to the pump.
    g. Driving away.
6. Once the customer has swiped their credit card, removed the hose from the pump and selected the grade of fuel. The gas station attendant signals to that pump that it can begin dispensing fuel at a fixed rate of 0.5 liter per second.
7. As fuel is dispensed by each pump, the display inside the gas station office is updated in real-time so that the operator can see exactly what is happening at all four pumps. The pump itself also displays the cost per liter of each grade of fuel, the grade of fuel selected by the customer and also how much fuel has been dispensed and the running cost of the purchase in real time.
8. A facility to change the cost of each grade of fuel should be provided to the gas station attendant computer.
9. The gas station has a relatively small tank which only holds 2000 litres of fuel (500 litres for each of the 4 grades), so a facility to restock the gas station should be provided. The stock level of fuel remaining in the tank should be displayed on the attendants display in the office and should flash in RED when it gets below 200 litres. At this point no new transactions can be started until the gas station has been refilled by delivery truck, but customers in the process of refuelling are

allowed to complete their transaction (obviously they can only consume, at most the remainder of the fuel in the gas station tank).

10. Customer cars are to be simulated by 'active class' objects which are created randomly during the simulation and are 'programmed' to try to consume a random amount (70 litres max) and grade of gas when they arrive at their pre-programmed pump (*they may have to wait in line*). Each customer will also be programmed with a random name and credit card number to complete transactions.

11. Each pump is simulated via an "Active Class object". These will not write anything to the DOS screen, instead they write to data pools in a producer/consumer arrangement. The Attendants computer (see below) is responsible for displaying all output to the screen.

    The customer will, as stated previously, be an active object (i.e. a thread running inside a class) and the communication between customer and pump will be via a pipeline rather than any kind of keyboard. In addition, each pump communicates with the gas station attendant's computer using a data pool in a producer/consumer arrangement. Finally each pump has access to a data pool which stores the quantity of fuel remaining in the tank; this allows the pump to deduct fuel from the tank in real-time as it is being dispensed.

The attendant's computer in the office is also simulated as a 'Process' which has access to the data pool representing the fuel in the tank (so that it can be refilled) and of course the four pumps discussed earlier (via the pipelines) so that the gas station attendant can control the price of fuel and enable/disable the pumps. The attendant's computer should also record each transaction performed at each of the four pumps, which should include

1. Time of purchase
2. Credit card number and name of purchaser
3. Grade of fuel selected
4. Quantity of fuel dispensed.

The history of all transactions for that day can be displayed on demand

The gas station attendant is 'you' controlling the office computer using commands entered at your PCs keyboard.

Any protocol or commands required by your simulation can be chosen by you.

An additional 20% will be awarded for what the assessor considers to be bells and whistles features, such as the clever use of say colour graphics (see online notes on cursor control under Win32 and colour in rt.h/rt.cpp files, e.g. MOVE_CURSOR(X,Y) function), or any other innovative features that you can come up with that are relevant to the gas station simulation, such as more than 4 pumps and/or showing in real-time the list of pending customers at a pump. Some bells and whistles features will obviously attract more marks than others; colour for example is not too difficult (see rt files for examples), while code to deal with more than 4 pumps is more interesting and worth more marks.

Make sure you save any previous working versions of your code. You would not believe how many students change something critical 10 mins before the demo and end up with something that does not work and no fall back situation!!!!

The same goes for backing up your work. **YOU** are responsible for this. Please do not come to the lab on the day of the demo and tell the TA that your laptop has crashed and you have lost all your work. You have an obligation to safeguard your work and keep backups/previous working versions. I'll say no more.

Deadline – You should arrange to present your demonstration in your designated lab period of teaching week 10.

Please do not ask to demonstrate in any other lab period (because for example, it gives you more time to work on it) as this is not feasible, the TA can only mark so many demos in a two hour period. Similarly, you should come to the lab prepared to demo **when asked**, do not continue to develop your solution on that day as this serves only to push back all the demos into the last 10 mins of the lab period which is likewise unacceptable. If you cannot demo when asked by the TA you may not get a mark!!

Group Work:

This assignment may be done in groups of NO MORE THAN 2, or, if you prefer, individually. All students who wish to be awarded marks for their work must attend the demonstration and expect to be asked questions about their design. If you cannot find a partner, then ask and I will announce it in the class and try to hook you up with someone.

All students within a group will be awarded the same mark i.e. that awarded for the assignment itself, unless there is clear disagreement about the amount of work each person did. If this cannot be resolved amongst the students themselves, then the TA will interview the students to determine their knowledge of the work they are submitting and award individual marks on that basis alone. The TAs decision is **final**. If you don't turn up, you don't get a mark regardless of whether your name is on a submission.

SUBMISSION OF WORK and AWARDS

YOU SHOULD UPLOAD YOUR VISUAL C++ PROJECT TO CANVAS AS A ZIPPED FILE BEFORE YOUR DEMONSTATION. ONLY A ZIPPED FILE WILL BE ACCEPTED. WITHOUT THE UPLOAD, YOU WILL NOT RECEIVE A GRADE FOR THIS ASSIGNMENT.

THE FINAL DECISION ABOUT WHAT IS A BETTER DESIGN RESTS WITH THE INSTRUCTOR OR TA. THEIR WORD AND DISCRETION ARE FINAL AND ABSOLUTELY NO DISCUSSION OR RENEGOTIATION OF ANY MARK FOR THE ASSESSMENT PROCESS ITSELF WILL BE ENTERED INTO.

IT SHOULD BE NOTED THAT DESIGN IS NOT AN EXACT SCIENCE AND EVERYONES SOLUTION WILL BE SOMEWHAT DIFFERENT. IT FOLLOWS FROM THIS THAT NEITHER IS MARKING SUCH A DESIGN AN EXACT SCIENCE.

THE MARKING PROCESS WILL THEREFORE NOT BE AN ABSOLUTE MEASURE OF ANYTHING, RATHER DEMONSTRATIONS WILL BE AWARDED MARKS RELATIVE TO THE PERFORMANCE OF OTHERS IN THE GROUP, IN SIMPLE TERMS, BETTER ASSIGNMENTS WILL ATTRACT BETTER MARKS, IF YOUR ASSIGNMENT IS INCOMPLETE, CONTAINS BUGS OR IS NOT AS EFFICIENT, THEN DO NOT EXPECT TO GET THE SAME MARK AS SOMEONE ELSE'S PROGRAM THAT DOES NOT SUFFER FROM THESE LIMITATIONS. THIS SHOULD BE OBVIOUS BUT IS SPELLED OUT HERE FOR THOSE STUDENTS THAT THINK SUCH THINGS NEED SPELLING OUT!!!