uOttawa General Engineering (GNG)

GNG1106 - Lab 6

Objectives

Practice the use of:

1. Conditional coding (if and if-else)
2. Loops (do-while) and sentinel variable
3. Symbolic constants (#define …)
4. Function definitions and function calls

Instructions

It is not allowed to use math.h in this lab to evaluate sin, cos, tan or exponential functions. Only use it for `fabs()` if needed.

Pre-Lab Submission (20%): Ensure you have submitted your pre-lab before attending the lab session.

Deliverable 1 (25%): Upgrade lab 5 Deliverable 3 as follows:

a. (10%) The calculation of sine x should be done in a function with the prototype: **`double sin(double x);`**

b. (10%) The calculation of cosine x should be done in a function with the prototype: **`double cos(double x);`**

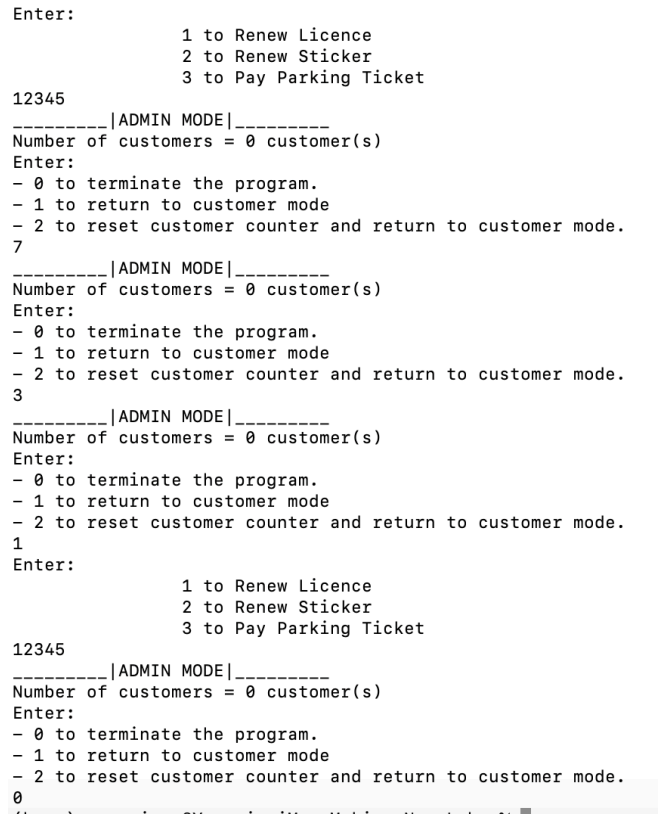
c. (5%) The main function: The remaining of the original lab 5 Deliverable 3 code should remain in the main function, and the functionalities of the program also should remain the same, the only change is that the main call both functions by passing the value of x and each function should returns the calculated cos(x) or sin(x) respectively, then the main uses them to calculate tan(x) and display the results.

Deliverable 2 (55%): Upgrade the basic version of the "City Hall" program seen in lectures so that the program contains two modes: the customer mode and the admin mode. Once started, the program runs in the customer mode, prompting a customer to make selection and give instructions. By the default, the program runs in the customer mode forever (unless interrupted by an administrator) and keeps a counter for the number of serviced customers. For the administrator to enter the admin mode, he needs to enter a secret option (i.e., an integer serving as the "password") in the customer mode (of course, this option is not displayed). When

entering the admin mode, the program prints the customer count so far, and give the administrator three options: 1) terminate the program, 2) return to the customer mode, 3) reset the customer counter and return to the customer mode. The screen shots of two executions of a sample program are given in the pictures below, where 12345 is the admin password.

```
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
1
Go to Counter 5
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
2
Go to Counter 8
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
4
Invalid selection!
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
12345
_________|ADMIN MODE|_________
Number of customers = 2 customer(s)
Enter:
- 0 to terminate the program.
- 1 to return to customer mode
- 2 to reset customer counter and return to customer mode.
2
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
3
Go to Counter 2
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
12345
_________|ADMIN MODE|_________
Number of customers = 1 customer(s)
Enter:
- 0 to terminate the program.
- 1 to return to customer mode
- 2 to reset customer counter and return to customer mode.
1
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
2
Go to Counter 8
Enter:
                1 to Renew Licence
                2 to Renew Sticker
                3 to Pay Parking Ticket
12345
_________|ADMIN MODE|_________
Number of customers = 2 customer(s)
Enter:
- 0 to terminate the program.
- 1 to return to customer mode
- 2 to reset customer counter and return to customer mode.
0
```

Sample execution 1



Sample execution 2

More detail instructions are given below.

1. (5%) Replace the switch-cases statements with if-else statements (as an exercise, not for upgrading the program)
2. (5%) Implement the loop using a do-while loop, so that the program by default repeats the customer mode.
3. (5%) Use symbolic constants for the service options and the instructions.
4. (5%) Display an error message if a wrong selection is entered.
5. (5%) Add a counter to count the number of customers visiting the city hall. The counter is initialized to zero when starting the program, then it will increment by 1 for each valid selection.
6. (5%) Add an administrative password as a symbolic constant
7. (25%) Implement the admin mode of the program. When entering the admin mode, a message ADMIN MODE should be displayed on top followed by the number of customers that used the system, then followed by 3 options. (Refer to the screenshots above,) If the admin enters other than 1, 2, or, 3 the system will loop forever till the

admin user enters a valid value between 0 and 2. Implement the admin mode as a separate function with the prototype `int admin(int count),` where input `count` is the value of the customer counter, and the returned value from the function should contain information that allows the main function to decide if the customer counter is to be reset and/or if the program should be terminated.

Check out and Submission

You must check out with your TA before submitting the deliverables. During the check out, your TA may inspect your work and to-be-submitted deliverables, and ask you questions to further check your understanding. At the end of the check out, your TA will give you an initial mark for the in-lab component of this lab and let you know. You must then submit the deliverables before the due time of this lab. While this initial mark is likely to be the final, your TA reserves the right to reduce this initial mark after checking more carefully the deliverables you submit

Grading Criterion

- **Correctness (60%):** Correct syntax, logic and execution.
- **Style (40 %) :** Descriptive variables names and appropriate indentation (especially those in the loops).