Lab11: Classes
OSL, Week of 31st Mar 2014

Objective:
1. Practice programs involving classes.

General instructions
1. This lab is to be done in groups of two. Work with the same partner as from previous labs. If your partner is absent, ask the TAs to include you in another group in your lab batch.
2. As before, you and your partner will get equal credit, irrespective of who did more work.

Lab instructions:
Exercise 1: Queue simulation
1. Download demo18-queue.cpp from the cs101 Moodle page and go through the class and function definitions.
2. Modify the function allinOne() to have a customerQ. If there is no taxi available when a customer arrives, then the customer ID is appended to this Q. As and when a taxi arrives, the customer at the front of this Q is removed. In this case, we are not using the queue class.
3. Now modify main() to add a customerQ as another instance of the queue class defined, and provide the same functionality as above. Which one did you find easier to do?
4. Set the length of both the Taxi queue and the Customer queue to be 5. Run your simulator for 30 arrivals; each arrival may be either a Taxi or a Customer, with equal probability. Note the number of customers who were asked to “Go Away” in your simulation.
5. Now modify the class queue to use a linked list instead of an array. The private data and the implementation of the public functions need to be modified, without any change in main().
6. Save your modified program as lab11-queue.cpp for uploading your submission.

Exercise 2: Stack implementation
1. From the above exercise, observe that a queue is a FIFO (First-In-First-Out) structure. On the other hand, a stack is a LIFO (Last-In-First-Out) structure. You may recall the use of a stack for implementing function calls. Two main functions provided by a stack are: push() to put a item at the top of the stack, and pop() to remove the item from the top of the stack.
2. Suppose the data items are integers (such as taxi numbers). Write a class stack that stores the data and provides the functions of push() and pop(). You may use either an array or a linked list for your implementation of your stack.
3. Test the working of your stack class. You could copy and modify the main program of your queue class for this purpose. Rename your program as lab11-stack.cpp and upload.

Exercise 3: Verify your in-sem marks
1. Take your quiz2 paper from your TA.
2. Verify that all your in-sem marks so far – quiz1, midsem, quiz2, labs, and project – have been entered correctly. If some labs are yet to be graded, they will be done by next week.
3. If there are any discrepancies in data entry, bring them to the notice of your TA.
4. You can verify the changes, if any, next week.

Upload your submission on Moodle – in the manner specified for previous labs.

<The End>