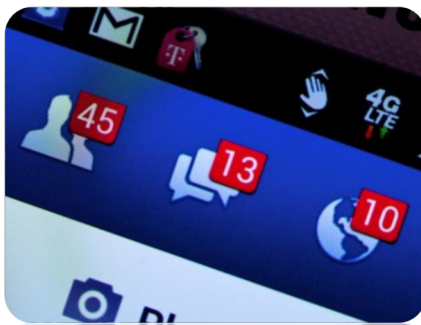


Introduction to **Programming** (CS200)

Shafay Shamail



Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. You put all your work onto the LMS folder designated for the lab (i.e. "Lab02") before the time the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.
5. Any cheating case will be reported to Disciplinary Committee without any delay.

Marks: Name: _____ Roll #: _____

Task1	Q1	Q2	Q3	Q4a	Q4b	Q4c	Q4d	Q5	Q6	Q7	Q8		Total
	3	3	3	3	3	3	3	3	3	3	0		30

Task 2	Q1a	Q1b	Q1c	Q2a	Q2b	Q2c	Q2d		Total
	6	3	1	5	5	5	5		30

Task3	Q1	Q2	Q3	Q4				Total
	5	5	5	5				20

Total Marks
Obtained

/100

Task 4	Q1a	Q1b	Q1c	Q2a	Q2b	Q2c	Q2d		Total
	2	3	5	2	3	2	3		20

Let's Begin.....

Task 1: (30)

1. Write a class called Point having three private data members x, y, z of type int. 3
2. Write setter functions. (Note: x, y, z are non-negative in nature.) 3
3. Write getter functions. 3
4. Write constructors. These should include:
 - a. Default constructor. Use default values of zero for x, y, and z. 3
 - b. Constructor to initialize x only 3
 - c. Constructor to initialize x and y only 3
 - d. Constructor to initialize x, y, and z. 3
5. Overload “==” to compare two Points. Return true if the corresponding members are equal otherwise return false. 3
6. Overload “<<”. The function shall print the point as “(x,y,z)”. Declare this overloaded function as friend of Point class. 3
7. Overload “>>”. The function shall read in x, y, z separated by spaces. Declare this overloaded function as friend of Point class. 3
8. Write main() function to test the class. 0

STOP AND SHOW YOUR WORK TO THE TA

Task 2:

(30)

1. Suppose a firm hires you and wants you to write a program for their special kind of Vending Machine. You have to write it by making a class "VendingMachine" with the following necessary private data members:
 - a. "available_items" 6
can be an array or vector and it should initially contain items given below, you need to store these items through constructor as these items are supposed to be in every new vending machine. (Note: you can work with fixed size array. It is up to you that how you format your array such that you can extract number and price of a particular item from your stored array element. Hint: Structures are helpful in such situations.)
 - i. 7 "coca_cola" cans (50 rupees per can)
 - ii. 3 "lays_chips" packs (20 rupees per pack)
 - iii. 3 "chocolates" (120 rupees per chocolate)
 - iv. 2 "biscuits" (17 rupees per biscuit)
 - v. 1 "candy" (5 rupees per candy)
 - b. "sold_items" 3
can be an array or vector (initially empty) (Hint: Think of a mechanism that will help in keeping track of each type of item sold)
 - c. "earned_money" 1
an integer (initially zero)
2. Following are the functions which your class should provide for proper working of machine. (Note: You can make additional helping functions for providing proper functioning.)
 - a. "print_available_items()" 5
It should print available items in a column (print them with or without prices, up to your creativity).
 - b. Overload "<<" so that it implements the same functionality as that of "print_available_items()". 5
 - c. "add_items()" 5
This function should ask for the name of item and then it should add that item in the list of available items. For simplicity, an item will be added if and only if at least one piece of that item exists already.
 - d. "sell_item()" 5
This function should ask for the name of an item and then remove one piece of that item from the list and update "sold_items" and "earned_money" accordingly.

STOP AND SHOW YOUR WORK TO THE TA

Task 3:

(20)

1. Make a friend function "tune_machine()" which should make all class parameters (variables or arrays) go to their initial condition. Following are initial conditions: 5
 - a. "available_items" should contain list of items provided in Task 1.1a
 - b. "sold_items" should be empty
 - c. "earned_money" should be zero

It is basically tuning the machine. It will show that you are a good programmer and hiring you was a good decision.

2. Overload the "==" operator such that it should compare one vending machine to other and tell if both vending machines have same available items, same sold items and same earned money. (usage: vending_object_1 == vending_object_2) 5
3. Overload the "-" operator (minus sign), such that it should work like opposite of adding an item i.e it should remove an item, if available, from vending machine. (usage: vending_object - "biscuits") 5
4. Overload the "+" operator (plus sign), such that it should work exactly like "add_items()" member function. (usage: vending_object + "biscuits") 5

STOP AND SHOW YOUR WORK TO THE TA

Task 4: (20)

1. Write an independent inline function “activateVM()”:
 - a. Which should take a reference to a given vending machine. It should: 2
 - b. Enlist names of functions which that vending machine can perform so that users can use it easily. 3
 - c. Allow a user to select an appropriate function and execute it. For example if a user wants to buy an item, the “sell_item()” could be used. 5
2. Write main() function to test the VendingMachine class.
 - a. Declare three objects of class Vending Machine. 2
 - b. Add a loop so that the program prompts the user to select one of the three vending machines. 3
 - c. Call the activateVM() function by passing reference of the selected vending machine. 2
 - d. The loop should terminate if any input other than 1, 2, or 3 is given otherwise it should continue. 3

Your company will give you permanent job based on your performance. ;-)

STOP AND SHOW YOUR WORK TO THE TA

Zip your tasks into one folder with format:

YourRollNo-Lab02

example "**2001001-Lab02**" and upload on LMS before the tab is closed. You will not be given extra time.