Western University

CS 1037A

Assignment 3

Professor: Ali Bou Nassif, Ph.D., P.Eng.

Instructions: Read carefully before you start.

This assignment should be done individually. The deadline is Tuesday December 2, 2014 at 11:55 pm. Please refer to the course outline (**Assignments Submissions and Late Assignments Policy Section)** for the late penalty. Each student must submit a zip file (zip and NOT rar) that compresses the main folder of his/her assignment. The name of the main folder must be the student username. The main folder should contain three folders only; question1, question2 and question3 (all lower case), as well as the ReadMe file. Each folder must contain the source files only of its related question. The Readme file should be very simple. If you used visual studio for example, the Readme file should contain one or two lines like "project files were successfully compiled using visual studio xxx (say what year / version)". All files should work under Visual Studio C++. It is very important that you comment your code thoroughly. Marks will be deducted if the code is not commented properly or if it not running under Visual Studio.

Question 1: 20 points

The goal of this question is to demonstrate how a function of one class can access public members of another class.

Consider the skeleton code on the following page. You will demonstrate how a function (***accessclass2***) in one class (***class1***) can call a function (***print***) in another class (***class2***).

1. Complete the implementation of the function ***accessclass2*** (complete the parameters, as well as the body)
2. Complete the implementation of the ***main*** function

Note: When you run the code, the output should be ***10***. This is the output when the ***print*** function is called.

You can modify anything to make you code runnable. Please note that there should be no “Inheritance”, nor “aggregation/composition” between these two classes.

```
#include <iostream>
using namespace std;


class class1 {
 public:
  int x;
  class1()
  {
        x = 0;
  }
  void accessclass2(// to be implemented)
  {
 // call the print() function in class2
  }
};

class class2 {
public:
       int y;
       class2 ()
       {
              y=10;
       }
       void print()
       {
              cout<<y<<endl;
       }
};

       int main(){
       // to be implemented
       return 0;
              }
```

Question 2: 30 points

Write a program in C++ to ask the user to input a set of characters, one at a time separated by the carriage return (Enter). Each character, as well as, its corresponding ASCII value should be stored in a linked list node. This means that each node in the linked list should have 3 parts; (1) ***char*** that stores the input of the user, (2) ***int*** that stores the ASCII value of the character and (3), a ***pointer*** to its successor node. When the user inputs the character ***0*** (zero), the program should break the loop, prints each character with its ASCII value on a separate line, then exits. You can implement the node as ***struct*** or ***class***. Please note that the ASCII code of any character can be retrieved by casting its integer value.

E.g. ***int x= 'A';*** // here ***x*** stores the value of ***65*** which is the ASCII value of ***A***. Remember, the user can input any character such as 1, 2, !, #, a, W, ….. The ASCII value of ***1*** (number one) is ***49***. Although ***0*** (zero) has an ASCII value of ***48***, this character (0) will be used by the program as an indication to break the

loop, traverse the list then exists. You cannot use the Standard Template Library (STL). Check this link to learn more about ASCII table http://www.asciitable.com/.

Question 3: 50 points

Consider the following requirements:

a. Class ***Vehicle***

- Variables
  - ***weight*** of type float, initial value is zero
- Functions
  - ***print()*** abstract function of type void

b. class ***Truck*** inherits from ***Vehicle***
The class ***Truck*** is not abstract

c. class ***Engine***

- Functions
  - ***start()*** of type void. It should print "engine started"
  - ***stop()*** of type void. It should print "engine stopped"

d. class ***Tire***

- Variables
  - ***pressure*** of type integer, initial value is 30
- Functions
  - ***inflate()*** of type integer that takes one integer argument called ***psi***. This function will add the value of ***psi*** to ***pressure*** and then it returns the new ***pressure*** of the tire

e. class ***Window***

- Functions
  - ***rollup()*** of type void. It should print "window is up"
  - ***rolldown()*** of type void. It should print "window is down"

f. class ***Door***

- Variables
  - ***window*** of type ***Window***
- Functions
  - ***open()*** of type void. It should print "door is open"
  - ***close()*** of type void. It should print "door is close"

g. class ***Car*** inherits from ***Vehicle***

- Variables
  - ***engine*** of type ***Engine***
  - ***tire***, an array of 4 tires, of type ***Tire***
  - ***left*** of type ***Door***

- ***right*** of type ***Door***

1- Construct the class diagram based on the above requirements. If you do not know any tool to construct the class diagram, you can download NClass from http://nclass.sourceforge.net/downloads.html. NClass is free and very easy to use.
2- Write a program in C++ to implement the above requirements. In the Test Driver:
    a. Create an object of type Truck of weight 30,000
    b. Create an object of type car
    c. Inflate the first tire (the first element of the array) by 10 psi
    d. Close the right door
    e. Roll up the window of the left door
    f. Start the engine

Note:

1- All members of the above classes are public and the inheritance type is public.
2- All initial values of member variables must be done through constructors
3- The program must be implemented into three different files, the header file, implementation file and test driver. You should include guards
4- All functions that are not required to modify the objects must be declared as ***const***
5- You can add constructors to any of the above classes if required
6- You should add missing member variables or functions to any of the above class if necessary