CS2400 **Laboratory Assignment #12** **(100 points)**

Writing classes with constructors

1. A simple class

In this lab, you are going to get some more practice with writing classes with constructors and passing stream parameters as arguments to a member functions etc.

Create a file called counter.cc:

Add this class definition:

```
class Counter
{
  public:
   Counter ();             //initializes the counter value to 0.

   Counter(int new_val);  // value is set according to the
                          // incoming argument.

   void increment();      //increment counter value by 1.

   int get_value();       //returns the value of member
                          //variable
  private:
        int value;
};
```

a. [5] Write the definition for the default constructor.

b. [5] Write the definition for the constructor with one argument.

c. [5] Write the definition for the accessor function.

d. [5] Write the definition for the increment function

e. [10] Write a short program (`main` function) to do the following. Create `Counter` object with a value you choose (hint: use the constructor with one argument. If the value is **less than 10 increment** the value **by 1**. Print the original value and the incremented value.

```
 Value at the beginning 5
 Value at the end 6
```

2. [50] Tollbooth class - Write a complete program.

Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a `50` cent toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected. Model this tollbooth with a class called T`ollbooth`. The two data members are type `int` for **the total number of cars**, and type `double` **to hold the total amount** of money collected. A **constructor** initializes both of these to `0`. A member function called `payingCar()` **increments the car total and adds 0.50 to the cash total.** Another function called `nopayCar()`**, increments the car total** but adds nothing to the cash total. Finally, write a member function called `display(ostream& fileout)` to **display the two totals to the screen.**

Include a `main` function to test this class. The program should allow the user to enter `'p'` to count a paying car, and `'n'` to count a nonpaying car and `'q'` should cause the program to print out the total number of cars and total amount collected. Save this program as `tollbooth.cc`

Sample output of this program can be as follows (user input in *italics*):

```
P - paid  N - Not paid  Q - Quit ->  p
P - paid  N - Not paid  Q - Quit ->  p
P - paid  N - Not paid  Q - Quit ->  n
P - paid  N - Not paid  Q - Quit ->  p
P - paid  N - Not paid  Q - Quit ->  n
P - paid  N - Not paid  Q - Quit ->  q

 Total number of cars 5
 Total amount collected $1.50
```

3. Submit the programs electronically.
4. Don’t forget to do the documentation.