Objective:

- Use of class level information (data/operations).

Task-1: discussed in class/lecture

```
class CMath
{
public:
    static float calcPower(int base, int exponent);
    static int calcGCD(int numerator, int denominator);
    static CString toCString(int num);
    static CString toInteger(CString);
    //you may add other mathematical functions in the same way
};
```

Task-2

There is still a possibility of creating more than one objects of the following class (discussed in lecture as well). Hunt that flaw but if you get exhausted then do discuss with me.

```
class Singleton
{
private:
    Singleton()
    {};
    ~Singleton()
    {}
    static Singleton * ptr;
public:
    static Singleton * CreateObject()
    {
        if (!ptr)
            ptr = new Singleton;
        return ptr;
    }
    static void freeObject()
    {
        if (ptr)
        {
            delete ptr;
            ptr=0;
        }
    }
};
Singleton * Singleton::ptr=0;
```

Task-3:

Design a class called 'Date'. The class should store a date in three integers: month, day, and year. There should be member functions to print the date in the following forms:

12/25/2012

December 25, 2012

25 December 2012

Demonstrate the class by writing a complete program implementing it.

Your setter functions should make sure following:

- A valid year is between 1900 and 2100
- A valid month is between 1-12
- A valid day can be between 1-31 (according to the respective month)

Make following `daysInMonth` array as class's private data member to know the number of days in each month.

```
static const int daysInMonth[ 13 ] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
31 }
```

Note:

- A leap year is a year which is either divisible by 4 yet not by 100, or it is divisible by 400.

```
class Date
{
private:
    int day;
    int month;
    int year;
                //need to static const
array
public:
    Date();
    Date(int,int,int);
    void setDate(int,int,int);
    void setDay(int);
    void setMonth(int);
    void setYear(int);
    int getDay() const;
    int getMonth()const;
    int getYear()const;
    void printFormat1()const;
    void printFormat2()const;
    void printFormat3()const;
    void incDay(int=1);
    void incMonth(int=1);
    void incYear(int=1);
    CString getDateInFormat1()const;
//if *this object contains day=25,
month=12 and year=2012 then it returns a
CSting object conataining "12/25/2012"
    CString getDateInFormat2()const;
    CString getDateInFormat3()const;
};
```

Task-4:
I hope you know the functioning/concept of a Toll Plaza. In this task, we need to create a system for a toll plaza through which the user will be able to keep track of the vehicles pass through toll plaza and will also be keeping the count and type of vehicles passing through plaza, it will also keep record of the amount of money earned for the current day.

As an initial version, we want to develop an electronic system of toll plazas with basic features. Read the description below and develop the system accordingly.

Write a class TollPlaza for the system. Whenever a vehicle passes a toll plaza it is charged some amount depending on the vehicle type. Assume there are only 5 types of vehicles on the roads Car, Bike, Bus, HeavyDutyTruck, LightDutyTruck (pick-up trucks, wagons etc.). We assign a constant value to each of these categories. Toll fee charged to each of them are stored in variables _carToll, _bikeToll, _busToll, _heavyDutyTruckToll, _lightDutyTruckToll respectively. Since this information is same for every toll plaza, it is good to share this information rather than keeping a separate copy of all of them in every class instance.

For instance Car and _carToll can be declared in following manner. static const int _CAR = 1; static int _carToll = 20; //20 Rs. Will be charged to vehicles of type Car Set Bike, Bus, HeavyDutyTruck, LightDutyTruck to const values 2, 3, 4 and 5 respectively. Tolls charged to each of them are 10, 50, 100 and 70 respectively.

Every toll plaza has a variable double collectedToll representing the toll fee collected at each of them.

Moreover each toll plaza has to collect a maximum amount of toll per day that is const long int MAX_TOLL. It is different for every toll plaza depending on whether the plaza is located on a busy road or not. Its value will be given whenever a TollPlaza instance is created. Once the collected collectedToll reaches the MAX_TOLL it shows the toll plaza was successful in attaining its today's goal amount. Also maintain an identifier variable int tollPlazaId and Address (char array) for every toll plaza.

Member Functions for TollPlaza:

- `TollPlaza(int id, char *ads, long int mt)`
  Receives plaza id, plaza address and plaza target max total
- `void resetTollPlaza()`
  Initializes data members for a new day.
- `void chargeVehicle(int vehicleType )`
  Charges the vehicle according to its type and adds to the collectedToll.
- `long int getCollectedTollToday() const`
  Returns the collected Toll.
- `long int getMaxToll() const`
  Returns the Target Toll to be collected.

When you will be done with the TollPlaza class, then you need figure the interface for the Toll Plaza application. For this Purpose, you need to create the following class named as 'TollPlazaSystem' which contains a static function 'startApplication', the function 'startApplication' will be responsible for creating menu and management for a TollPlaza object.

```
class TollPlazaSystem
{
public:
    static void startApplication()
};
int main()
{
    TollPlazaSystem::startApplication();
}
```

Solution of Task-4:

The solution is given for your help incase if you stuck somewhere or want to compare your solution with the one given below.
So please ☺ don't see below, do your full effort in implementing the above task and then you may compare it with the solution given below.

```
//TollPlaza.h
class TollPlaza
{
private:
    static const int _CAR;
    static const int _BIKE;
    static const int _BUS;
    static const int _HEAVY_DUTY_TRUCK;
    static const int _LIGHT_DUTY_TRUCK;

    static int _carToll;
    static int _bikeToll;
    static int _busToll;
    static int _heavyDutyTruckToll;
    static int _lightDutyTruckToll;

    long int collectedToll;
    const long int MAX_TOLL;

    int tollPlazaId;
    char tollPlazaAddress[200];

public:
    TollPlaza(int id, char *ads, long int mt);
    void resetTollPlaza();
    void chargeVehicle(int vehicleType );
    long int getCollectedTollToday() const;
    long int getMaxToll() const;
};

//TollPlaza.cpp
#include "TollPlaza.h"
const int TollPlaza::_CAR=1;
const int TollPlaza::_BIKE=2;
const int TollPlaza::_BUS=3;
const int TollPlaza::_HEAVY_DUTY_TRUCK=4;
const int TollPlaza::_LIGHT_DUTY_TRUCK=5;

int TollPlaza::_carToll=20;
int TollPlaza::_bikeToll=10;
int TollPlaza::_busToll=50;
int TollPlaza::_heavyDutyTruckToll=100;
int TollPlaza::_lightDutyTruckToll=70;

TollPlaza::TollPlaza(int id, char *ads, long int mt):MAX_TOLL(mt)
{
    collectedToll = 0;
}
void TollPlaza::resetTollPlaza()
{
    collectedToll = 0;
}
void TollPlaza::chargeVehicle(int vehicleType )
{
```

```
    switch (vehicleType)
    {
        case 1:
            collectedToll += _carToll;
            break;
        case 2:
            collectedToll += _bikeToll;
            break;
        case 3:
            collectedToll += _busToll;
            break;
        case 4:
            collectedToll += _heavyDutyTruckToll;
            break;
        case 5:
            collectedToll += _lightDutyTruckToll;
            break;
    }
}
long int TollPlaza::getCollectedTollToday() const
{
    return collectedToll;
}
long int TollPlaza::getMaxToll() const
{
    return MAX_TOLL;
}

//TollPlazaSystem.h
class TollPlazaSystem
{
public:
    static void startApplication();
};

//TollPlazaSystem.cpp
#include "TollPlazaSystem.h"
#include <iostream>
using namespace std;

void TollPlazaSystem::startApplication()
{
    TollPlaza tp(1,"Thokar Niaz Toll Plaza", 10000);
    char choice = '0';
    while(true)
    {
        cout<<"Press 1 to Charge Vehicle\n"
            <<"Press 2 to see Collected Toll\n"
            <<"Press 3 to Start New Day\n"
            <<"Press 0 to Shut Down System ";
        cin.get(choice);
        cin.ignore();
        switch(choice)
        {
            case '1':// to charge car
            {
                cout<<"\nEnter Vehicle Type :\n"
                    <<"\t 1 for Car\n"
                    <<"\t 2 for Bike\n"
                    <<"\t 3 for Bus\n"
                    <<"\t 4 for Heavy Duty Truck\n"
```

```
                    <<"\t 5 for Light Duty Truck ";
                char vType='0';
                cin.get(vType);
                cin.ignore();
                tp.chargeVehicle(vType - '0');
                if (tp.getCollectedTollToday()>=tp.getMaxToll())
                    cout<<"\n\nCongratulations: You achieved the target today :)\n";
            }
            break;

            case '2'://display collected toll so far
            {
                cout<<"\n\tCollected Toll Today:
"<<tp.getCollectedTollToday()<<"\n\n";
            }
            break;

            case '3'://reset: to start new day
            {
                tp.resetTollPlaza();
                cout<<"\n\n\tGood Luck for Today :)\n\n";
            }
            break;

            case '0'://To Exit
            {
                exit(0);
            }
            break;
        }

    }
}
//driver.cpp
#include <iostream>
using namespace std;
int main()
{
    TollPlazaSystem::startApplication();
}
```