**Lab Manual – Static Members in Class**

You are given following in this manual:

1. “LabManualStatic.cpp” (code is given below)
2. “CarsData.txt” (Data is given below)
3. “RequiredOutput”

Partial definitions of classes **Car** and **Helper** are given in the cpp file. Your task is to update the classes’ definitions according to the RequiredOutput.

**Important Instructions:**

* Do not change Main and ReadDataFromFile functions.
* Your program should not leak any memory.
* Your program should not throw any exception.

**LabManualStatic.cpp** (Copy paste following code in your cpp file. Comment all the code. Uncomment main function line by line, uncomment related code to run that line successfully. Add new code if required.)

|  |
| --- |
| #include<iostream>  #include<fstream>  using namespace std;  class Helper  {  public:  static int StringLenght( char\* str )  {  //Write Yourself  // This function should return lenght of str    }  static void StringCopy(char\*& dest, char\*& src){  //Deep Copies src into dest.  }  static char\* GetStringFromBuffer(char\* str)  {  //Write Yourself  //This function should allocate required memory on heap,  //copy string in this memory using StringCopy and return newly created cstring.    }  };  class Car  {  private:  static int totalCars; // initialize it to zero yourself  int model;  char\* make;  char\* name;  char\* color;  public:  Car()  {  model = 0;  color = make = name = 0;  totalCars++;  }  //Write Getter of totalCars yourself    void ReadDataFromFile(ifstream& fin)  {  char temp[20];  fin>>model;  fin>>temp;    //This is how we call static functions without object  make = Helper::GetStringFromBuffer(temp);    //Set rest of the values yourself.  }  void PrintListView()  {  //Write yourself  }  void PrintDetailView()  {  //Write yourself  }  ~Car()  {  cout << "Destroying ";  PrintListView();  //Deallocate memory yourself  }  void Input()  {  //Do not consume one extra byte on heap  //Use only one temp buffer    }  };  Car\* ReadDataFromFile(char\* fileName)  {  ifstream fin;  int totalCars = 0;  char buffer[80];  fin.open()  if (fin.is\_open(fileName))  {  fin >> totalCars;  //cout << totalCars;  fin.getline(buffer, 80, '\n'); //We are not saving comment  //cout << buffer << endl;  Car\* carsList = new Car[totalCars];  int i = 0;  while (!fin.eof())  {    carsList[i].ReadDataFromFile(fin);  i++;  }  return carsList;  }  else  {  return 0;  }  }  void main()  {  //Comment all the code. Then uncomment it line by line.  //Implement/add functionality for uncommented line.  //Execute and verify result with output.  Car\* carsList = ReadDataFromFile("CarsData.txt”);  int count = 0;  if (carsList != 0)  {    count = Car::GetTotalCars(); //Calling static function  cout << "Total Number of Cars in System:\t" << count << endl;    //Test one Print at a time.  cout << "\nCars List:\n\n";  {  for (int i = 0; i < count ; i++)  {  carsList[i].PrintListView();  }  }  //Comment above printing and Test Printing 2  cout << "\nCars List:\n\n";  {  Car temp = carsList[0];  temp.PrintListView();    for (int i = 1; i < count ; i++)  {  temp = carsList[i];  temp.PrintListView();  }  }  if(carsList)  delete[] carsList;  }    cout << "Total Number of Cars in System:\t" << Car::GetTotalCars() << endl << endl << endl;  Car testCar;  testCar.Input(); // Take car data from user  testCar.PrintDetailView();    cout << "Total Number of Cars in System:\t" << testCar.GetTotalCars() << endl << endl << endl;  } |

**CarsData.txt** (Copy paste following data in your “CarsData.txt” file. Follow the file name. You may remove the comments from file. Don’t forget to submit your data file.)

|  |
| --- |
| 4 //Total no. of cars  2015 Suzuki WagonR Grey  2012 Honda City White  2011 Toyota Corolla Black  2013 Suzuki Cultus Grey |

**Required Output:**

