Project Report

Group Members	Registration No.
Waleed Ahmed	FA18-BCE-065
Hasnain Ahmad	FA18-BCE-068

Project Title:	Integrated Highway Toll Management System
Subject:	Object Oriented Programming (CSC-241)
Submitted To.	Mam Amber Madeeha Zeb

Brief Description of Project

This is an Integrated Toll Management System Project. The program allows the user to enter the record of vehicle came at toll plaza then program calculate toll for the vehicle according to the specifications of the vehicle also record of the vehicle is saved in computer. It also have advance features of storing record and reviewing when needed, also it can calculate the total amount of toll collected from vehicles.

Program Structure

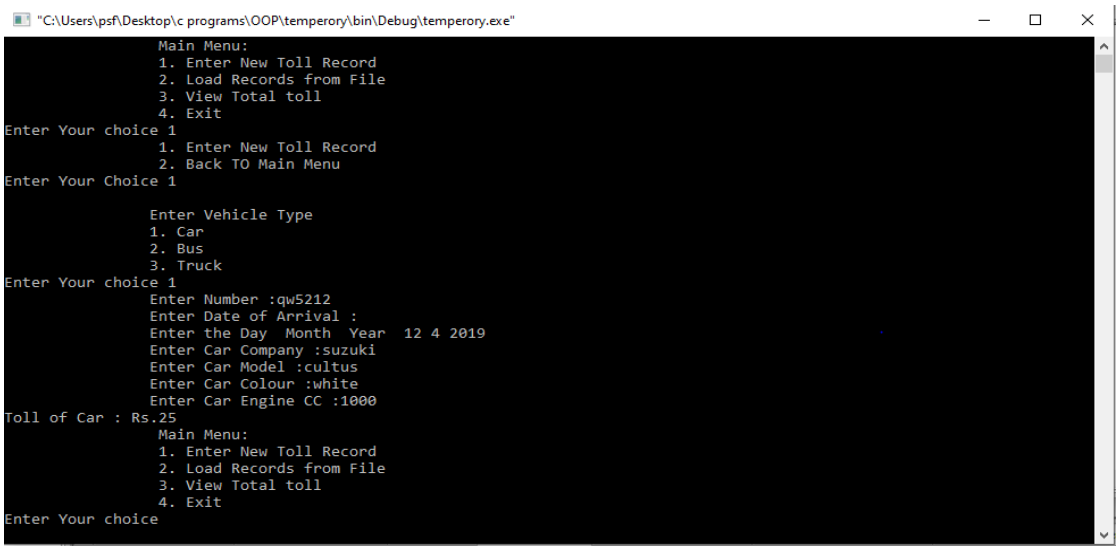
Creation of the menu is done in main function. Program has a class “Vehicle” at the top of hierarchy, then it has three derived classes which are “Car”, “Bus”, “Truck”. Vehicle class is the base class so, it has data members which are common in all vehicles (car, bus, truck). Each child class has its own data members and member functions for data input/output, toll calculation and file management.

Program Working

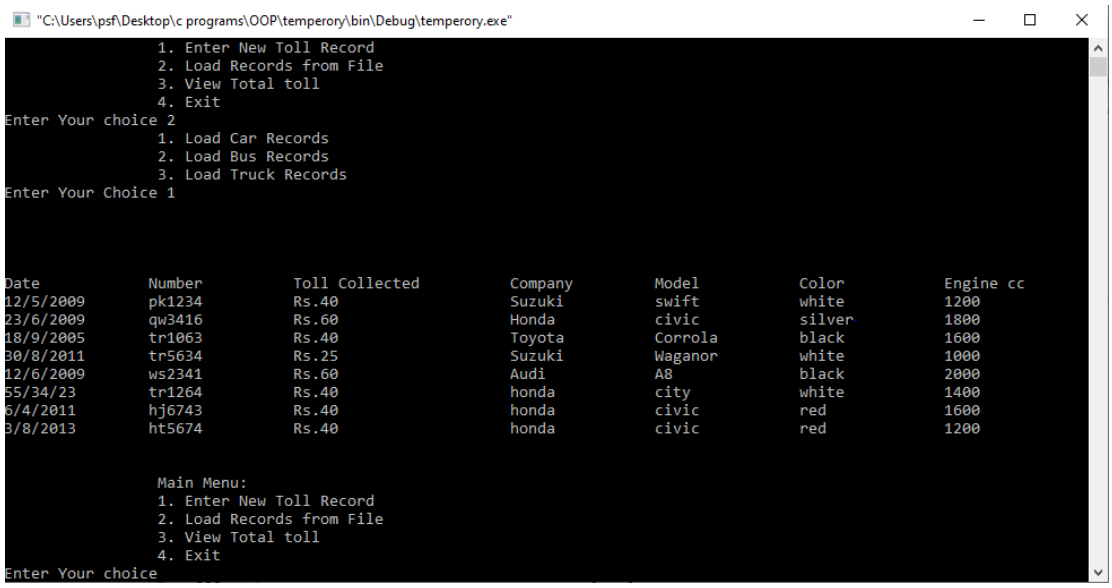
Program presents a main menu to user at the start (menu is made using an infinite loop and cascaded switch case statements). Then user have four choices for further actions i.e. he can enter the record for new vehicle, he can view the previously entered records, he can view the total amount of toll collected up till now and can exit from the program. Entering into the first choice leads user to another menu where it asks to enter the type of vehicle he wants to enter i.e. car, bus, truck. For executing this operation program allocates memory with help of “new” functions and assign its address to a pointer of “vehicle” type, then functions are called for data input, toll calculation and file writing. Now it is important to mention here that these functions are made virtual so that calling function from child class object calls its own function rather than calling parent class functions.

As soon as user enters the data of vehicle, program calculates the toll according to the entered data of vehicle and prints it on the console screen. After the toll calculation the program writes the data of vehicle into a binary file. File writing is done after opening it in append mode with binary flag. Data is written in file using write function of c++. Also it is important to mention here that

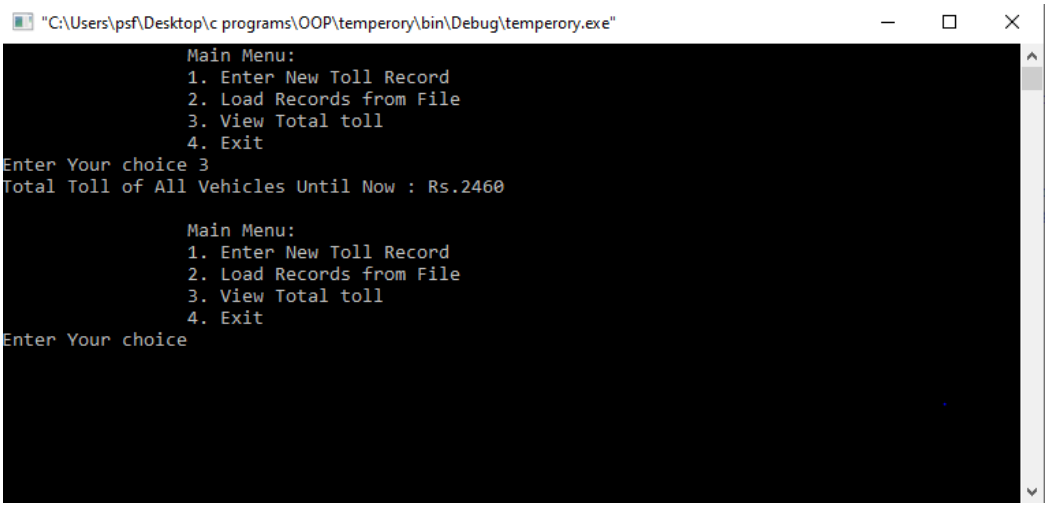
each child class has its own functions for writing data in file also each class has its own file for saving the records.



The second choice in the main menu is loading the records from file. As soon as user enter the second choice in main menu, it displays another menu for selection of the vehicle type i.e. car, bus, truck whose records have to be loaded from file, then after choosing one of them program displays a list of vehicles of that particular type which user had chosen. This operation is performed using the read function of c++. First the file is opened in input mode with binary flag then read function is called. The record is read in a loop one by one and each time the records is printed. So a list of records of vehicles is displayed when user enter the choice.



The third choice in main menu is to calculate total toll of all the vehicles up till now. After choosing this option the program calls a function to calculate the toll of all vehicles of one type i.e. cars. This is done by reading the toll of vehicle from the record in the file and depositing it in a variable, similarly this process is done for bus and truck. Then All three results are added to calculate total toll.



Concepts Used

1. **Inheritance**

   The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important feature of Object Oriented Programming. This also provides an opportunity to reuse the code functionality and fast implementation time.

2. **Polymorphism**

   The word polymorphism means having many different forms. Typically, polymorphism occurs when there is a hierarchy of classes and they are related by inheritance. C++ polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function. If the called functions is made virtual then when it is called from a pointer of base class which has address of derived class object it will call the function present inside derived class.

2.1 Virtual Functions

A virtual function is a function in a base class that is declared using the keyword virtual. Defining in a base class a virtual function, with another version in a derived class, signals to the compiler that we don't want static linkage for this function.

2.2 Pure Virtual Functions

A virtual function whose declaration ends with = 0 is called a pure virtual function. A pure virtual function is a virtual function in C++ for which we need not to write any function definition and only we have to declare it.

2.3 Abstract Classes

Abstract Class is a class which contains at least one Pure Virtual function in it. Abstract classes are used to provide an Interface for its sub classes. Classes inheriting an Abstract Class must provide definition to the pure virtual function, otherwise they will also become abstract class

3. File Handling

These are classes used to perform output and input of characters to/from files:

- **ofstream:** Stream class to write on files.
- **ifstream:** Stream class to read from files.
- **fstream:** Stream class to both read and write from/to files.

These classes are derived directly or indirectly from the class istream and ostream. We have already used objects whose types were these classes: cin is an object of class istream and cout is an object of class ostream. Therefore, we have already been using classes that are related to our file streams.

4. Operator Overloading

C++ allows us to specify more than one definition for an operator in the same scope, which is called operator overloading.

An overloaded declaration is a declaration that is declared with the same name as a previously declared declaration in the same scope, except that both declarations have different arguments and obviously different definition (implementation).

When you call an overloaded operator, the compiler determines the most appropriate definition to use, by comparing the argument types you have used to call the function or operator with the parameter types specified in the definitions. The process of selecting the most appropriate overloaded function or operator is called overload resolution.

5. Friend Functions

Private members are accessed only within the class they are declared.
Friend function is used to access the private and protected members of different classes. It works as bridge between classes

- Friend function must be declared with friend keyword
- Friend function must be declare in all the classes from which we need to access private or protected members.
- Friend function will be defined outside the class without specifying the class name.
- Friend function will be invoked like normal function, without any object.