

**R V COLLEGE OF ENGINEERING®**

**Bengaluru – 560059**

(*Autonomous Institution Affiliated to Visvesvaraya*

*Technological University, Belagavi*)

INTRODUCTION TO C++ PROGRAMMING

EXPERIENTIAL LEARNING

**A REPORT ON**

**CAR RENTAL SYSTEM**

## Submitted By,

## SHRIVARSHA RVCE22BCS095

## Submitted to , SUHAS PERI RVCE22BCS108

## PROF.PRIYA D KIRAN V RVCE22BCS212

## DEPARTMENT OF ISE SAMARTH G RVCE22BCS180

TABLE OF CONTENTS:



|  |  |
| --- | --- |
| SI NO. | CONTENTS |
| 1. | INTRODUCTION |
| 2. | CONCEPTS IMPLEMENTED |
| 3. | EXPLANATION OF THE CONCEPTS USED |
| 4. | EXPLANATION OF EXECUTION OF PROGRAM |
| 5. | OUTPUT |
| 6. | CONCLUSION |



**INTRODUCTION**

* A car rental management system is an autonomous system that will preserve the records of all the cars available, cars rented, etc. The user can rent a car based on its efficiency, performance, effort or cost.
* This system allows car rental companies to manage their fleet of vehicles, including information such as make, model, year, and availability.
* This feature allows car rental companies to manage customer information, including contact details, driver's license information, and payment information.
* This feature allows car rental companies to generate bills and invoices for customers based on the rental period and other charges, such as insurance or additional services.
* We have created an interface through which the customers can rent the cars. This interface displays the number of cars and type of cars which are available and also their features so that it is easy for customers to make their choice.
* The bill to be paid for renting will be displayed after the customers choose the type of car and the number of days to rent.
* This interface is password protected as it initially asks the password for logging in and if the password entered is wrong it will redirect the user back again to the login page.





**The concepts which are used in preparing this project are**

**1.INHERITANCE**

It allows us to create a new class (derived class) from an existing class (base class). The derived class inherits the features from the base class and can have additional features of its own.

Types of inheritance used

**Multiple inheritance**: It is a type of inheritance where a class can inherit from more than one classes.

**hierarchical inheritance**: The feature of the base class is inherited onto more than one sub-class.

**2.POLYMORPHISM**

Polymorphism means the same entity (function or object) behaves differently in different scenarios.

**3.ENCAPSULATION**

Encapsulation is defined as binding together the data and the functions that manipulate them.

**4.ABSTRACTION**

It is the process of only showing the necessary details to the user and hiding the

other details in the background

**5.POINTERS**

The pointer in C++ language is a variable, it is also known as locator or indicator that points to an address of a value.

**6.VIRTUAL FUNCTIONS**

A virtual function is a base class member function that you can redefine in a derived class to achieve polymorphism.

**7.FILE HANDLING**

File handling in C++ is a mechanism to store the output of a program in a file and help perform various operations on it.

**8.VECTORS**



Vectors in C++ are sequence containers representing arrays that can change their size during runtime.

**EXPLANATION OF THE CONCEPTS USED IN THIS PROJECT**



**1.Header files**

**iostream**:

We use this header file as it is the standard library for reading input and writing output from our program.

**Iomanip:**

This is a library which has many function that are used to manipulate the output of the program. We have used the functions like setw to manipulate the output.

**conio:**

The word conio.h stands for console input-output some of the important functions are given by the header file conio.h are-

clrscr(); used for clearing the window of the console.

getch(); used for pushing the console window.

**unistd and direct:**

These are the libraries which has many function such as system(“CLS”),system(“PAUSE).

**vector:**

This is the class template that contains the vector container and its member functions.

**fstream:**

This helps in implementing high level input/output on file based streams. Here in this program we have to access many text files so this header file is of prime importance for us.

**2.Classes**



* A Class is a user defined data-type which has data members and member functions.
* An Abstract class is a class that is designed to be specifically used as a base class. An abstract class contains at least one pure virtual function.
* In this program we use six different types of classes namely **Car, user, Benz, Honda, hyundai, welcome.**
* Here **Car** is an abstract class**, user** is a base class, **Benz, Honda, hyundai** are derived classes.
* **Benz, Honda, hyundai,** inherit both classes, **Car and user**. So multiple inheritance is used here.

**3. Inbuilt functions**

Built-in functions are ones for which the compiler generates inline code at compile time.

* **System function[system()]:**

**system()** is used to invoke an operating system command from a C/C++ program.(stdlib.h or cstdlib needs to be included to call **system**). Using **system(),** we can execute any command that can run on terminal if operating system allows. Here system function is used to clear the screen and pause the screen.

* **Sleep function[sleep()]:**

**Sleep()** function in C++ will sleep the present executable for the specified time by the thread.

Parameter:  Time\_period is in seconds it represents the sleep time taken.

Return Type:  The return type of sleep function is an integer where if the function is successfully executed then the value returned will be 0, else minus the value of the time period returned.

Here we use to sleep function to delay the the execution for 1 or 2 sec.



* **Getch function[\_getch()]:**

**getch()** is a  function and is present in [conio.h header](http://en.wikipedia.org/wiki/Conio.h) file. **getch()** also reads a single character from the keyboard. But it does not use any buffer, so the entered character is immediately returned without waiting for the enter key. Here this function is used to return character which is stored in pass(password) with the help of pushback function.

* **Getline function[getline()]:**

The C++ **getline()** is a standard library function that is used to read a string or a line from an input stream. The **getline()** function extracts characters from the input stream and appends it to the string object until the delimiting character is encountered. While doing so the previously stored value in the string object *str* will be replaced by the input string if any. Here this function is used to store the text in files (like welcome text) in a variable and later it is printed on terminal.

* **Push back function[push\_back()]:**

C++ **push\_back()** is a pre-defined function that is used to insert data or elements at the end of the vector or it pushes the element in the vector from the back. Here this function is used to push characters obtained with help of **\_getchar()** into the pass variable.

* **Close function[close()]:**

This function helps to close an existing file. Here this function is used to close files like welcome.txt.



**4. User defined functions**

User-defined functions in C++ are**functions that are created by the user to perform a specific task.**

Pure virtual functions are the methods which are declared in the base class and should be defined in each and every derived class.

* **userinput():**

Using this function we collect name of user, driving license number, registration number of car.

* **spec():**

Using this function we display the specialities of the car. Since this is a **Pure virtual function** this has to be defined inside derived class.

* **comprent():**

Using this function we calculate the rent for the car for required number of days. Since this is a **Pure virtual function** this has to be defined inside derived class.

* **display():**

Using this function we display the invoice with total amount and due date to pay the rent.

* **login():**

This function is used to for entering login credentials by passing which further program is executed otherwise not. Here we use getchar and push back functions with if and else loop.With this function we have implemented abstraction.

* **welcum():**

Using this function we have displayed starting welcome message and further messages.



**5.Vectors:**

Vectors are the same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container. Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators.

A 2D vector is a vector of the vector. Like 2D arrays, we can declare and assign values to a 2D vector.

* **cars:**

Using this vector we have stored car names and their rent cost per day. This is a **2D** **vector.**

**6.While loop and switch:**

A while loop in C++ is an**entry-controlled loop** that is best suited for cases where you are not sure about the exact number of iterations. The while loop works on a boolean condition that regulates the iteration of the loop. It terminates the loop once it violates the condition.

The switch statement allows us to execute a block of code among many alternatives.

We have used this to allow the user to enter the choice and this choice we have used it in switch to select different car choices. We have used a pointer to base classes Car and user. Using this pointer to point address of various derived class and pointing to the function to be executed, we have achieved runtime polymorphism.

**7.Ifstream:**

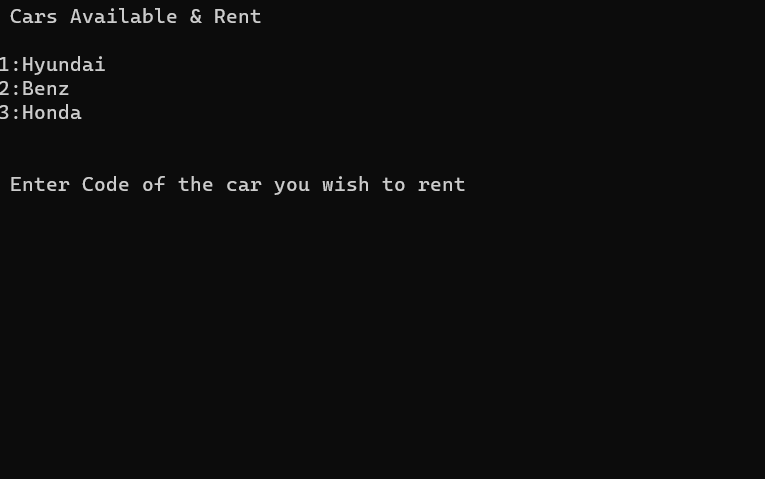
Used to read the files without permission to change it.

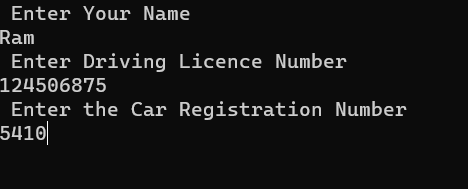
**OUTPUT**



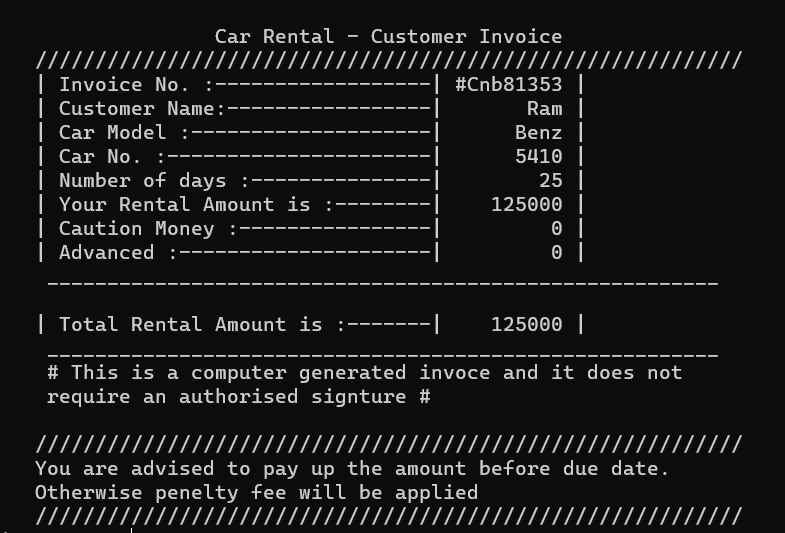














**CONCLUSION**

* The car rental system developed by us effectively manages car rentals and provides customers with user friendly interface to book the car, to view available car and their rental rates and finally generating the invoice for the respective car booked by the customer.
* Future enhancements in this project include integration of payment gateway to allow for online payment processing and implementation of strong robust booking system to handle Complex reservations.

* This project demonstrates versatility of C++ programming language and OOP’s principle.

