**AISSCE (2019 - 2020)**

**COMPUTER SCIENCE PROJECT**

**“TAXI BOOKING SYSTEM”**

**IN**

**C++**

******

******

***PROJECT BY:*** S.PRAJOTH

DEPARTMENT OF COMPUTER SCIENCE

**TEAM MEMBERS:**

**N.Vittal**

**S.Prajoth**

Smt. Ramkuwar Devi Fomra Vivekananda Vidyalaya

Chromepet, Chennai – 44

Unit of VIVEKANANDA EDUCATIONAL SOCIETY (Regd)

 Department of Computer Science

**BONAFIDE CERTIFICATE**

This is to certify that this bonafide project work has been done by

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of class XII in

Smt. Ramkuwar Devi Fomra Vivekananda Vidyalaya, Chromepet,

Chennai – 44 during the year 2019 – 2020.

**Principal**  **Staff Incharge**

Submitted for (All India Senior Secondary Certificate Examination)

**Computer Science Practical Examination** held on \_\_\_\_\_\_\_\_\_\_\_\_\_ at

Smt. Ramkuwar Devi Fomra Vivekananda Vidyalaya , Chromepet , Chennai – 44.

**Internal Examiner** **External Examiner**

**Date: School Seal:**

**Acknowledgement**

I, hereby place my humble obeisance to our **Honourable Correspondent Sri.Venkatasubramani,** for equipping us with exemplary infrastructure and laboratory amenities. In light of this fact, I also take the privilege of thanking our **Respected Principal Smt. Indra Shankar,** for her able support.

The colourful project as it seems, is but the product of the inspiration, enthusiasm and guidance, provided by our **Computer Science Teacher** **Smt.V.Vijayalakshmi** to whom, I owe a lot.

I also clinch this opportunity to thank my parents and friends who have served as the backbone, in completion of the project.

Finally I would like to thank CBSE for giving me this opportunity to undertake this project.

**INDEX:**

* C++ OVERVIEW
* INTRODUCTION ABOUT THIS PROJECT
* HEADER FILES AND THEIR PURPOSES
* SYSTEM CONFIGURATION
* CLASSES & FUNCTIONS DECLARED
* SOURCE CODE
* OUTPUT
* BIBLIOGRAPHY

C++ OVERVIEW

**INTRODUCTION**

C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming. C++ is regarded as a **middle-level** language, as it comprises a combination of both high-level and low-level language features.

C++ was developed by **Bjarne Stroustrup** starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983. C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

**FEATURES OF OBJECT-ORIENTED PROGRAMMING**

* *CLASS:*

In OPP Languages it is must to create a class for representing data. Class contains variables for storing data and functions to specify various operations that can be performed on data. Class will not occupy any memory space and hence it is only logical representation of data.

* *DATA ENCAPSULATION:*

Within a class variables are used for storing data and function to specify various operations that can be performed on data. This process of wrapping up of data and functions that operate on data as a single unit is called as data encapsulation.

* *DATA HIDING:*

Within a class if a member is declared as **private,** then that member cannot be accessed from outside the class (i.e.) that member is hidden from rest of the program. This process of hiding the details of a class from rest of the program is called as data hiding.

* *DATA ABSTRACTION:*

The ability to represent data at a very conceptual level without any details (i.e.) filtering out the required brief information from the whole is called data abstraction. It is specifying what to do, but not how to do. It is a flexible feature for having an overall view of an object’s functionality.

* *MODULARITY:*

The operations / methods should be defined separately for each module of the software project. One of the object’s features can be used by other objects as a reusable component.

* *OBJECT AND INSTANCE:*

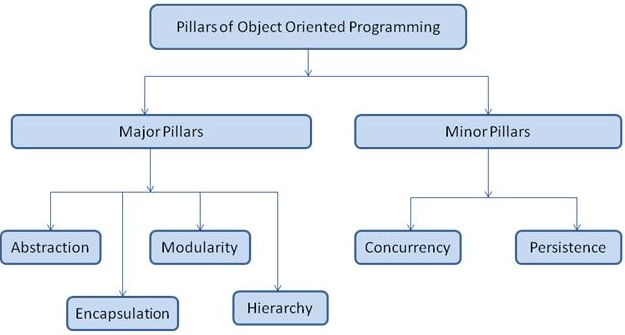
Class will not occupy any memory space. Hence to work with the data represented by the class you must create a variable for the class, which is called as an object. When an object is created by using the keyword new, then memory will be allocated for the class in heap memory area, which is called as an instance and its starting address will be stored in the object in stack memory area.

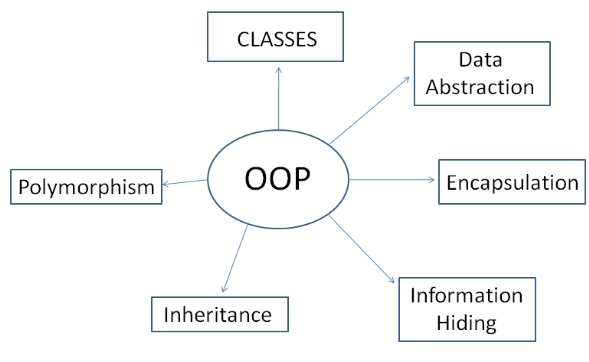
* *INHERITANCE:*

This is the process by which a class can be derived from a base class with all features of base class and some of its own. Advantage of inheritance is reusability of the code with the help of functions which can be inherited. During inheritance, the class that is inherited is called derived class and the class from which it is derived is called the base class.

* *POLYMORPHISM:*

Polymorphism means having more than one form. Polymorphism can be achieved with the help of overloading and overriding concepts. Polymorphism is classified into compile time polymorphism and runtime polymorphism.

**FLOWCHART OF FEATURES OF OOPS**

***Concept:*** **

**STANDARD LIBRARIES**

Standard C++ consists of three important parts:

* The core language giving all the building blocks including variables, data types and literals, etc.
* The C++ Standard Library giving a rich set of functions manipulating files, strings, etc.
* The Standard Template Library (STL) giving a rich set of methods manipulating data structures, etc.

**THE ANSI STANDARD**

The ANSI standard is an attempt to ensure that C++ is portable -- that code you write for Microsoft's compiler will compile without errors, using a compiler on a Mac, UNIX, a Windows box, or an Alpha. The ANSI standard has been stable for a while, and all the major C++ compiler manufacturers support the ANSI standard.

**LEARNING C++**

The most important thing to do when learning C++ is to focus on concepts and not get lost in language technical details. The purpose of learning a programming language is to become a better programmer; that is, to become more effective at designing and implementing new systems and at maintaining old ones. C++ supports a variety of programming styles. You can write in the style of FORTRAN, C, Smalltalk, etc., in any language. Each style can achieve its aims effectively while maintaining runtime and space efficiency.

**USES OF C++**

C++ is used by hundreds of thousands of programmers in essentially every application domain. C++ is being highly used to write device drivers and other software that rely on direct manipulation of hardware under real-time constraints. Anyone who has used either an Apple Macintosh or a PC running Windows has indirectly used C++ because the primary user interfaces of these systems are written in C++.

INTRODUCTION ABOUT THIS PROJECT

**Taxi Booking System:**

“Taxi booking system” aims at developing a new and simple software that can be used to book a taxi from a desired location.

This project converts all the manual work i.e. calling a cab that is time consuming and error prone (no proper location, fare) to fully automated system which helps save time, improve customer services like decent fares etc...

**Objectives of the proposed system:**

* To increase the efficiency and accuracy of this process.
* Hassle-free process.
* To make the retrieval of information faster.
* To make the system more feasible.
* To make the system more reliable to avoid ambiguity.
* To make the system more flexible.

HEADER FILES AND THEIR

PURPOSES

**HEADER FILES AND THEIR PURPOSES**

1. **<IOSTREAM.H> -** for stream of objects (cin and cout).
2. **<FSTREAM.H> -** for file handling purposes.
3. **<CONIO.H> -** for clrscr ( ) and getch ( ) functions.
4. **<STDIO.H> -** for standard I/O Operations.
5. **<STRING.H> -** for string handling.
6. **<STDLIB.H> -** for standard library functions.
7. **<PROCESS.H> -** for exit ( ) function.

SYSTEM REQUIREMENTS

**SYSTEM REQUIREMENTS**

**HARDWARE CONFIGURATION REQUIREMENTS**

1. *Microprocessor Minimum Required:* Pentium III/ Intel Duo Core.
2. *Minimum Speed Required:* 1.4 GHz.
3. *Minimum RAM Capacity Required:*512MB.
4. *Minimum Hard Disk Space Required:*900MB.
5. *Colour Monitor* **–** Dell 17” Monitor / Any other suitable colour monitor.

**SOFTWARE CONFIGURATION REQUIREMENTS**

1. *OS version:*Windows XP / Windows 7 / Windows 8 / Windows 10
2. *Language:*TURBO C++ VERSION 3.0
3. *Package:* MS WORD 10

CLASSES & FUNCTIONS DECLARED

**Class: car\_details**

Data Members:

*carcode:* Unique code for car // integer type

*regno:* Car registration number // integer type

*name [30]:* Name of the car // character array

*model:* Manufacturing year // integer type

*category [10]:* Car category based on size (micro, mini, prime etc.) // character array

**Class: route\_details**

*routecode:* unique code for route // integer type

*from [20]:* from place // character array

*to [20]:* to place // character array

*distance:* number of KM // integer type

**Class: charge\_details**

*chargecode:* unique code for charge // integer type

*timefrom:* Initial time // integer type

*timeto:* Final time // integer type

**Class: ride\_information**

*custcode:* unique customer code // integer type

*carcode:* unique car code // integer type

*timing:* ride timing // integer type

*routecode:* route travelled // integer type

**Functions declared**

*void login\_user():* // User login

*void register\_newuser():* // User registration

*void forgot\_usernamepassword():* // Username & password reset

*void cars\_display(car\_details[]):* // Displaying car details

*void routes\_display(route\_details[]):* // Displaying route details

*void charges\_display(charge\_details[]):* // Displaying timings

*void book\_ride (car\_details[],route\_details[],charge\_details[],ride\_information):* // Getting ride details from user

*void calc\_fare (car\_details[],route\_details[],charge\_details[],ride\_information):* // Fare calculation

SOURCE

CODE

#include<iostream.h>

#include<fstream.h>

#include<stdlib.h>

#include<string.h>

#include<conio.h>

#include<stdio.h>

#include<process.h>

**void** login\_user();

**void** register\_newuser();

**void** forgot\_usernamepassword();

**class** car\_details {

**int** carcode;

**int** regno;

**char** name[30];

**int** model;

**char** category[10];

public:

**void** heading() {

cout << "\nCarcode " << " REGNo\t" << " Name\t\t" << "Model\t" << "Category";

}

**void** display() {

cout << endl << carcode << "\t" << regno << "\t" << name << "\t" << model << "\t" << category << endl;

}

**int** getcarcode() {

**return** carcode;

}

**char** \* getcategory() {

**return** category;

}

}

cars[3];

**class** route\_details {

**int** routecode;

**char** from[20];

**char** to[20];

**int** distance;

public:

**void** heading() {

cout << "\nRoutecode " << "From\t" << " To\t " << " Distance";

}

**void** display() {

cout << endl << routecode << "\t" << from << "\t" << to << "\t" << distance << endl;

}

**int** getroutecode() {

**return** routecode;

}

**int** getdistance() {

**return** distance;

}

}

routes[5];

**class** charge\_details {

**int** chargecode;

**int** timefrom;

**int** timeto;

public:

**void** heading() {

cout << "\nCode" << " Timefrom " << " Timeto" << endl;

}

**void** display() {

cout << endl << chargecode << "\t" << timefrom << "\t" << timeto << endl;

}

**int** gettimefrom() {

**return** timefrom;

}

**int** gettimeto() {

**return** timeto;

}

}

charges[3];

**class** ride\_information {

**int** custcode;

**int** carcode;

**int** timing;

**int** routecode;

public:

ride\_information() {

timing,

custcode,

routecode,

carcode = 0;

}

public:

**void** confirm\_booking() {

cout << "Select your Car" << endl;

cin >> carcode;

cout << "Ride timing" << endl;

cin >> timing;

cout << "Enter route";

cin >> routecode;

}

**int** getcarcode() {

**return** carcode;

}

**int** gettiming() {

**return** timing;

}

**int** getroutecode() {

**return** routecode;

}

}

booking;

**void** cars\_display(car\_details[]);

**void** routes\_display(route\_details[]);

**void** charges\_display(charge\_details[]);

**void** book\_ride(car\_details[], route\_details[], charge\_details[], ride\_information);

**void** calc\_fare(car\_details[], route\_details[], charge\_details[], ride\_information);

**void** main() {

**int** choice;

cout << "\n\n^^^^^^^^^^^^^^^^^^^^^^^^CAB BOOKING SYSTEM^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n";

cout << "\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SRDF Vivekananda Vidyalaya\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

cout << "DONE BY :" << endl << "N.Vittal" << endl << "S.Prajoth" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n";

cout << " Welcome to login page \n\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MENU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

cout << "1.LOGIN" << endl;

cout << "2.REGISTER" << endl;

cout << "3.FORGOT PASSWORD (or) USERNAME" << endl;

cout << "\nEnter your choice :";

cin >> choice;

cout << endl;

**switch** (choice) {

**case** 1:

login\_user();

**break**;

**case** 2:

register\_newuser();

**break**;

**case** 3:

forgot\_usernamepassword();

**break**;

default:

system("cls");

cout << "You've made a mistake , give it a try again\n" << endl;

main();

}

car\_details cars[3] = {

{

9306,

12172,

"Tata Indica",

2017,

"Micro"

},

{

9307,

12173,

"Toyota Etios",

2018,

"Prime"

},

{

9308,

12174,

"Maruti Ertiga",

2019,

"SUV"

}

};

route\_details routes[5] = {

{

1001,

" Chennai",

"Salem\t",

364

},

{

1002,

" Chennai",

"Tiruchirapalli ",

330

},

{

1003,

" Chennai",

"Tiruvannamalai ",

195

},

{

1004,

" Chennai",

"Coimbatore ",

507

},

{

1005,

" Chennai",

"Tiruchendur ",

638

}

};

charge\_details charges[3] = {

{

8305,

3,

10

},

{

8304,

11,

17

},

{

8303,

18,

24

}

};

**char** option;

**while** (option == 'N' || 'n') {

**int** choice = 0;

cout << "1.Display available cars:\n";

cout << "2.Display available Routes:\n";

cout << "3.Display cab availability time:\n";

cout << "4.Book your Ride\n";

cout << "5.EXIT\n";

cin >> choice;

**switch** (choice) {

**case** 1:

cars\_display(cars);

**break**;

**case** 2:

routes\_display(routes);

**break**;

**case** 3:

charges\_display(charges);

**break**;

**case** 4:

book\_ride(cars, routes, charges, booking);

**break**;

**case** 5:

cout << "Thanks for using this program: ^\_^ ^\_^\n\n";

clrscr();

exit(0);

**break**;

}

cout << "\nDo you want to exit?(Y/N)\n";

cin >> option;

}

getch();

}

**void** showheaders(**char** title[50]) {

cout << "\*\*\*\*\*\*\*\*\*\*\*\*" << title << "\*\*\*\*\*\*\*\*\*\*\*\*";

}

**void** cars\_display(car\_details cars[3]) {

showheaders("Select Cars");

cars[0].heading();

**for** (**int** i = 0; i < 3; i++) {

cars[i].display();

}

}

**void** routes\_display(route\_details routes[5]) {

showheaders("Select Routes");

routes[0].heading();

**for** (**int** i = 0; i < 5; i++) {

routes[i].display();

cout << "-------------------------------------------------";

}

}

**void** charges\_display(charge\_details charges[3]) {

showheaders("Select Charges");

charges[0].heading();

**for** (**int** i = 0; i < 3; i++) {

charges[i].display();

cout << "-------------------------------------------------";

}

}

**void** book\_ride(car\_details cars[], route\_details routes[], charge\_details charges[], ride\_information booking) {

showheaders("Book your Riding");

cout << endl;

cout << "Select Car --->\n";

cout << "Cars Available:" << endl << "Carcodes:" << endl << "Hope you have seen the details of carcodes listed below!" << endl;

**for** (**int** i = 0; i < 3; i++) {

cout << i + 1 << ")" << cars[i].getcarcode() << " " << endl;

}

cout << " Timing --->\n";

cout << " Timefrom " << "Timeto " << endl;

**for** (**int** j = 0; j < 3; j++) {

cout << cars[j].getcarcode() << "-->" << charges[j].gettimefrom() << "\t" << charges[j].gettimeto() << endl;

}

cout << " Select Route --->\n";

cout << "Available Routes:" << endl << "Route Codes:" << endl << "Hope you have seen the details of routecodes listed below!" << endl;

**for** (**int** k = 0; k < 5; k++) {

cout << k + 1 << ")" << routes[k].getroutecode() << " " << endl;

}

booking.confirm\_booking();

calc\_fare(cars, routes, charges, booking);

}

**void** calc\_fare(car\_details cars[], route\_details routes[], charge\_details charges[], ride\_information booking) {

cars[0].heading();

**for** (**int** i = 0; i < 3; i++) {

**if** (cars[i].getcarcode() == booking.getcarcode()) {

cars[i].display();

}

}

**int** distance = 0;

routes[0].heading();

**for** (**int** j = 0; j < 5; j++) {

**if** (routes[j].getroutecode() == booking.getroutecode()) {

distance = routes[j].getdistance();

routes[j].display();

}

}

**int** rate, DriverBeta = 0;

**for** (**int** k = 0; k < 3; k++) {

**if** (booking.gettiming() >= charges[k].gettimefrom() && booking.gettiming() <= charges[k].gettimeto()) {

**if** (strcmpi(cars[k].getcategory(), "Micro") == 0) {

rate = 9;

DriverBeta = 250;

}

**if** (strcmpi(cars[k].getcategory(), "Prime") == 0) {

rate = 11;

DriverBeta = 350;

}

**if** (strcmpi(cars[k].getcategory(), "SUV") == 0) {

rate = 15;

DriverBeta = 450;

}

}

}

cout << "Rate per KM" << rate << endl;

cout << "--------------------------------------------";

cout << "\n Your ride fare for selected ride is=" << distance \* rate << endl;

cout << "--------------------------------------------" << endl;

cout << " Extra Driver Beta alloted is=" << DriverBeta << endl;

cout << "--------------------------------------------" << endl;

cout << " Your total fare is=" << (distance \* rate) + DriverBeta << endl;

cout << "--------------------------------------------" << endl;

cout << "A driver will be assigned to you shortly. Also, driver details will be" << endl << "sent through SMS and email." << endl;

}

**void** login\_user() {

**int** login\_r = 0;

**char** username\_r[50], password\_r[50], user\_r[50], pass\_r[50];

system("cls");

cout << "Enter the following details" << endl;

cout << "USERNAME :";

cin >> username\_r;

cout << "PASSWORD :";

cin >> password\_r;

ifstream login\_file("database\_Cab\_Booking.txt");

**while** (login\_file >> user\_r >> pass\_r) {

**if** (strcmp(user\_r, username\_r) == 0) {

**if** (strcmp(pass\_r, password\_r) == 0)

{

login\_r = 1;

system("cls");

}

}

}

login\_file.close();

**if** (login\_r == 1) {

cout << "\nHello " << user\_r << "\nLOGIN SUCCESSFUL\nWe're glad that you're here.\nThanks for logging in\n";

cin.get();

} **else** {

cout << "\nLOGIN ERROR\nPlease check your username and password ^\_^\n";

main();

}

}

**void** register\_newuser() {

**struct** database {

**char** Fname[40], Lname[40], mbno[12], email[30];

}

db;

cout << "Welcome for the registration process:\n";

system("cls");

cout << "Enter your First Name: \n";

cin >> db.Fname;

cout << "Enter your Last Name: \n";

cin >> db.Lname;

cout << "Enter your Mobile Number: \n";

cin >> db.mbno;

cout << "Enter your Email: \n";

cin >> db.email;

cout << "Don't worry! We dont't share your email and mobile number" << endl << " to third party resources!!";

**char** reg\_username[50], reg\_password[50];

cout << "\n Enter the username: ";

cin >> reg\_username;

cout << "\nEnter the password :";

cin >> reg\_password;

ofstream register\_file("database\_Cab\_Booking.txt", ios::app);

register\_file << reg\_username << ' ' << reg\_password << endl;

system("cls");

cout << "\nRegistration Sucessful\n";

main();

}

**void** forgot\_usernamepassword() {

**int** forgot\_user;

system("cls");

cout << "Forgotten your Username or Password? We're here for help\n";

cout << "1.Search your id by username" << endl;

cout << "2.Search your id by password" << endl;

cout << "3.Main menu" << endl;

cout << "Enter your choice :";

cin >> forgot\_user;

**switch** (forgot\_user) {

**case** 1:

{

**int** fuser = 0;

**char** search\_user[50],

user\_f[50],

pass\_f[50];

cout << "\nEnter your remembered username :";

cin >> search\_user;

ifstream f\_user("database\_Cab\_Booking.txt");

**while** (f\_user >> user\_f >> pass\_f) {

**if** (strcmp(user\_f, search\_user) == 0) {

**char** s\_password[50];

fuser = 1;

strcpy(s\_password, pass\_f);

}

}

f\_user.close();

**if** (fuser == 1) {

**char** s\_password[50];

cout << "\n\nHurray, account found\n";

cout << "\nYour password is " << s\_password;

main();

} **else** {

cout << "\nSorry, Your user ID is not found in our database\n";

cout << "\n Kindly contact our Customer Care and visit our website for more help. \n";

cin.get();

main();

}

**break**;

}

**case** 2:

{

**int** fpass = 0;

**char** search\_pass[50],

user1[50],

pass1[50];

cout << "\nEnter the remembered password :";

cin >> search\_pass;

ifstream f\_pass("database\_Cab\_Booking.txt");

**while** (f\_pass >> user1 >> pass1) {

**if** (strcmp(search\_pass, pass1) == 0) {

**char** s\_username[50];

fpass = 1;

strcpy(s\_username, user1);

}

}

f\_pass.close();

**if** (fpass == 1) {

**char** s\_username[50];

cout << "\nYour password is found in the database \n";

cout << "\nYour Id is : " << s\_username;

main();

} **else** {

cout << "Sorry, We cannot found your password in our database \n";

cout << "\n Kindly contact our Customer Care\n";

cin.get();

main();

}

**break**;

}

**case** 3:

{

cin.get();

main();

}

default:

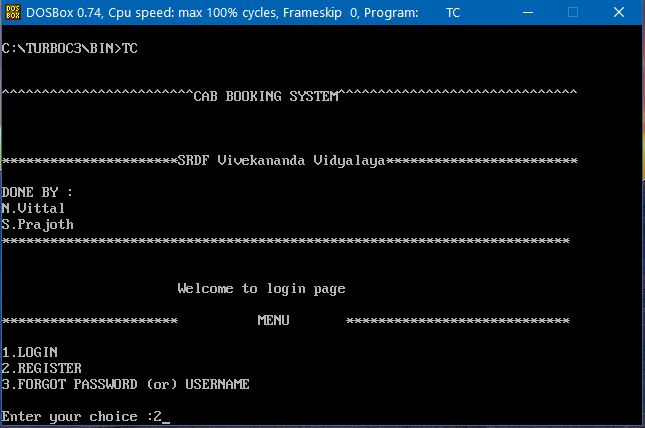
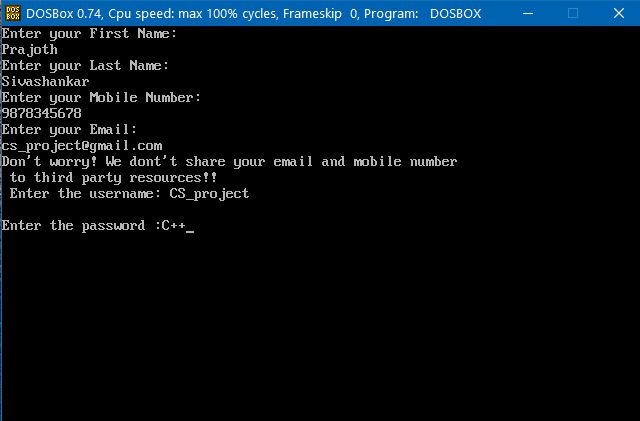
cout << "Sorry, You entered wrong choice. Kindly try again" << endl;

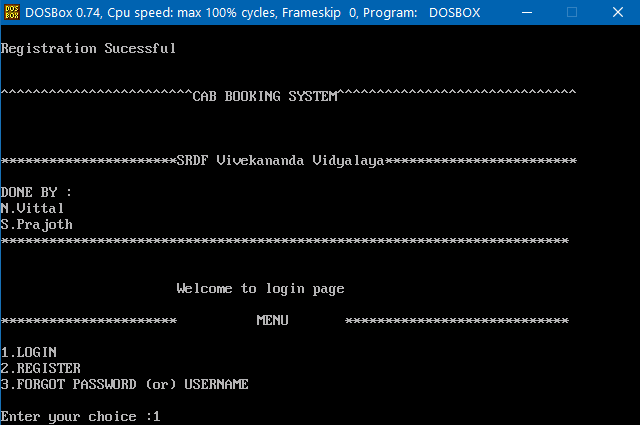
forgot\_usernamepassword();

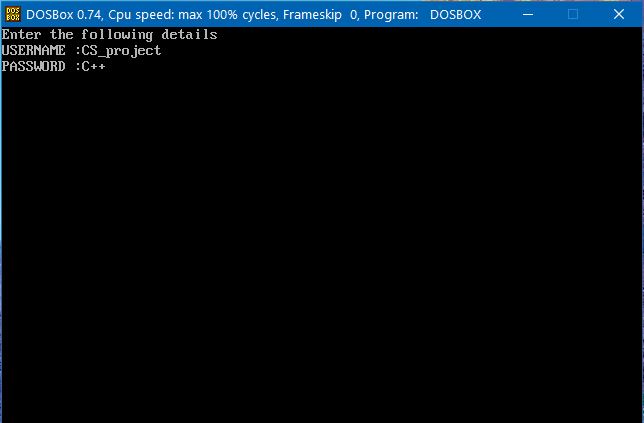
}

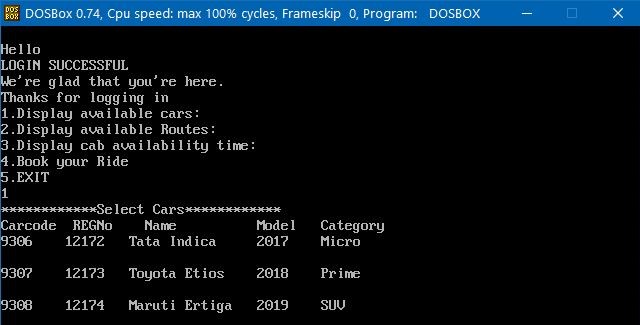
}

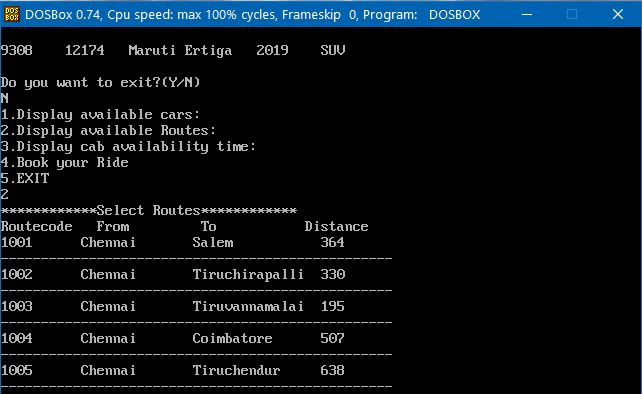
OUTPUT

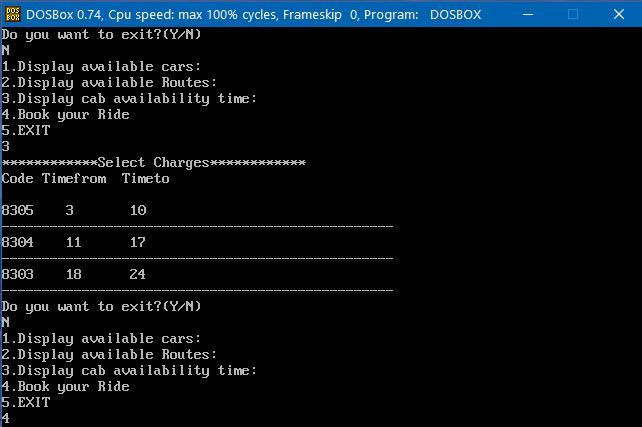
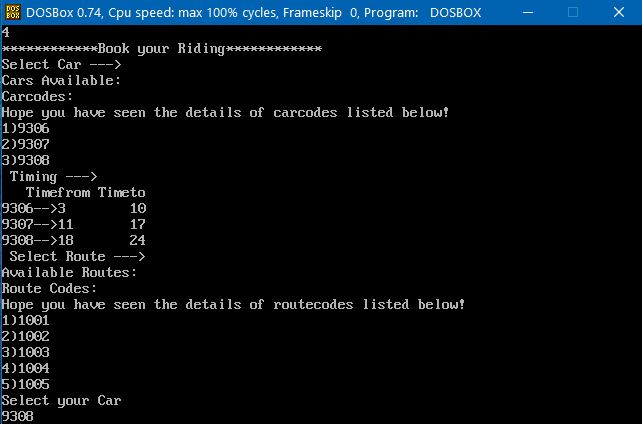


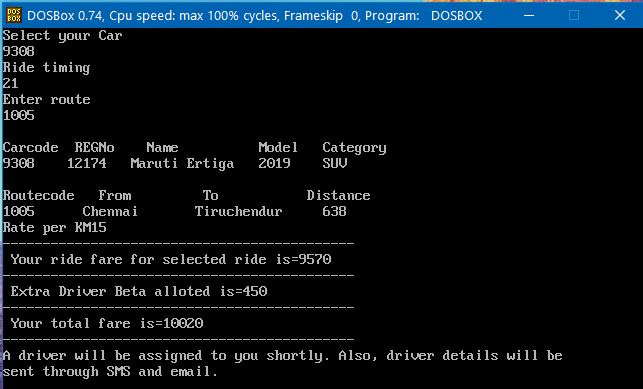
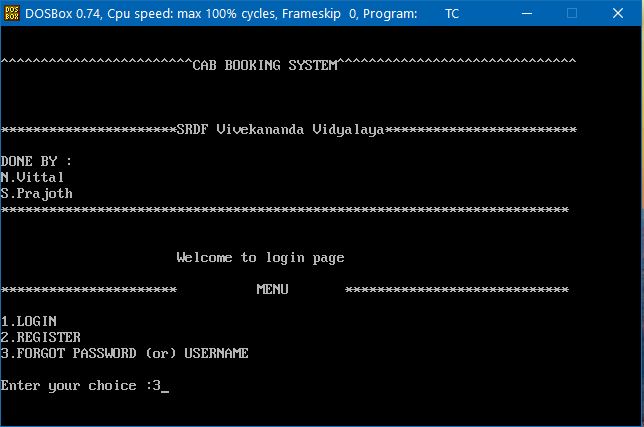




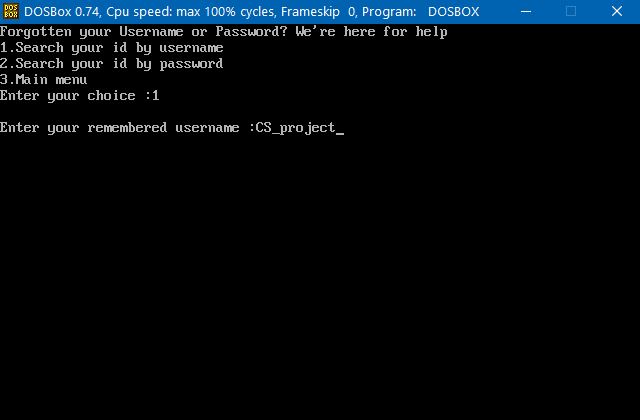
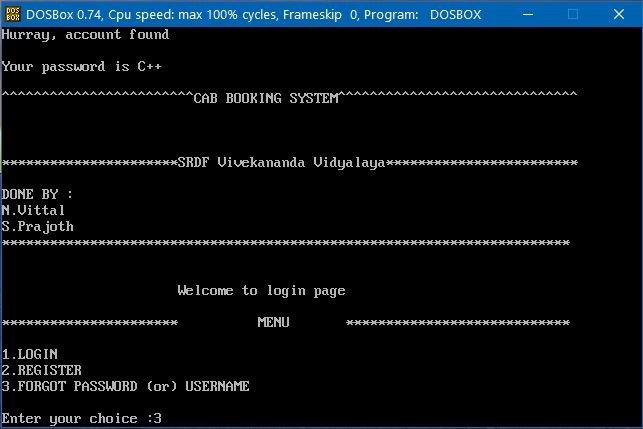


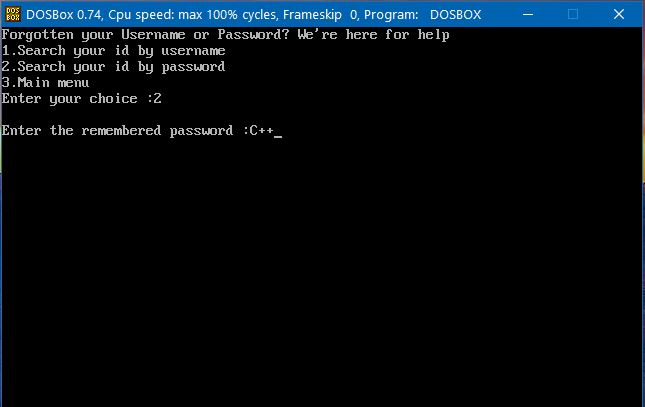
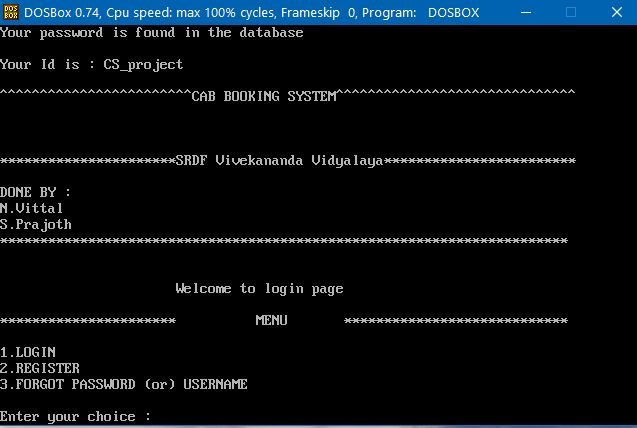






FORGOT USERNAME AND PASSWORD





Bibliography

**BIBLIOGRAPHY**

*Book referred:-* **Computer Science with C++ by Sumita Arora**