A Project Report on

**Bus Reservation System** Using C++

Submitted for the partial fulfillment of the

Degree of Bachelor of Technology

By

**KARAN SINGH**

**SAUD AKHTAR KHAN**

**YASH AGRAWAL**

Under the Supervision of

*Swati Gaur*



Department of Computer Science & Engineering

SCHOOL OF MANAGEMENT SCIENCES

Campus Address: 19th Kilometer Stone Sultanpur Road,

Lucknow - Uttar Pradesh, India

Pincode: 226501

**INDEX**

|  |  |
| --- | --- |
| **S.No.** | **Description** |
| 1. | Acknowledgement |
| 2. | Abstract |
| 3. | Introduction |
| 4. | Objective |
| 5. | Feature |
| 6. | Materials And Methodology   * Introduction * History * Features of C++ * Hardware Implementation * Operating system |
| 7. | Software Description |
| 8. | Program (Using C++) |
| 9. | Project Overview |
| 10. | Conclusion And Recommendations |
| 11. | References |

**ACKNOWLEDGEMENT**

It is our regards that we were supervised under our mentors Dr. Hemant Kumar Singh & Ms. Swati Gaur as well as well wishers. It is a great honor to represent our thoughts and ideas by the face of programming in c++ language to our mentors. Any of the time that our mentor has spended on us is a very valuable for us. It is our mentor who helped us in any of our problems (related to carrier, education).

It is a great pleasure for us to work on our mini project (Bus Reservation System). It has enormous value in our carrier to be learn our self how to be in team, how to be work as team and talents which are hidden in our self.

**ABSTRACT**

Traveling is a large growing business across all countries. Bus reservation system deals with maintenance of records of details of each passenger. It also includes maintenance of information like schedule and details of each bus.

We observed the working of the Bus reservation system and after going through it, we get to know that there are many operations, which they have to do manually. It takes a lot of time and causing many errors while data entry. Due to this, sometimes a lot of problems occur and they were facing many disputes with customers. To solve the above problem, and further maintaining records of passenger details, seat availability, price per seat, bill generation and other things, we are offering this proposal of computerized reservation system.

By using this software, we can reserve tickets from any part of the world via internet. Customer can check availability of bus and reserve selective seats. The project provides and checks all sorts of constraints so that user does give only useful data and thus validation is done in an effective way.

**INTRODUCTION**

The focus of the project is to computerize traveling company to manage data, so that all the transactions become fast and there should not be any error in transaction like calculation mistake, bill generation and other things. It replaces all the paper work.

This reservation system has three modules. First module is for Bus registration. Second module is for Bus availability. Using third module passenger can book a ticket.

It can generate bill during booking.

**OBJECTIVE**

The main **objective** of the proposed **system**” **Bus Reservation System**” is to eliminate the manual **reservation system**. Making the **reservation system**, fast, user friendly avoid the unnecessary delay in **reservation**. The new **system** needs to develop that can handle lots of records and reports efficiently.

**Features**

1. Interactive and user friendly

2. Simple and easy to Book a ticket.

3. Easy to add new bus.

**MATERIALS AND METHODOLOGY**

* **Introduction**

The C++ programming language supports multiparadigm programming. We can write programs in procedural, object-oriented, generic way at the same time. However, it is difficult to figure out exercises for the terminal examinations since not easy to separate the algorithmic cogitation from the knowledge of the programming language. There are some basic elements that programmer students have to know: constructors, parameter passing, objects, inheritance, standard library, handling constants, copying objects, functions and member functions, etc. Exercises must be multiparadigm according to the C++ language. Using only one paradigm in C++ is not enough. This results in that we have to distinguish the different linguistic constructs on the basis of its complexity. Many questions are arisen in connection with the exercises of terminal examinations. How can we gauge the procedural, the object-oriented, and the generic paradigms at the same time? How can we gauge students' C++ knowledge when we do not lay stress on the algorithmic cogitation? What kind of exercises may be interesting by the Standard Template Library? Which C++ constructs are reckoned to be more difficult and which ones considered to be easier? What are the most important ones? In this paper we give answers to the previous questions, we describe our methodology to assessment of students' C++ knowledge in a semi-automatic grading way. We also present exercise examples that worked out according to our methodology. We take stock of students' results in the paper.

* **History**

The C++ programming language has a history going back to 1979, when [Bjarne Stroustrup](http://www2.research.att.com/~bs/) was doing work for his Ph.D. thesis. One of the languages Stroustrup had the opportunity to work with was a language called Simula, which as the name implies is a language primarily designed for simulations. [The Simula 67 language](http://staff.um.edu.mt/jskl1/talk.html) - which was the variant that Stroustrup worked with - is regarded as the first language to support the object-oriented programming paradigm. Stroustrup found that this paradigm was very useful for software development, however the Simula language was far too slow for practical use.  
  
Shortly thereafter, he began work on "C with Classes", which as the name implies was meant to be a superset of the C language. His goal was to add object-oriented programming into the C language, which was and still is a language well-respected for its portability without sacrificing speed or low-level functionality. His language included [classes](http://www.cplusplus.com/doc/tutorial/classes/), basic [inheritance](http://www.cplusplus.com/doc/tutorial/inheritance/#inheritance), [in lining](http://www.cplusplus.com/doc/tutorial/functions2/#inline), [default function arguments](http://www.cplusplus.com/doc/tutorial/functions2/#default_values), and strong type checking in addition to all the features of the C language.

* **Features of C++**

Here are some of the remarkable features of C++ language:

1. OOP (Object-Oriented Programming)

*C++ is an object-oriented language*, unlike C which is a procedural language. This is one of the most important features of C++. It employs the use of objects while programming. These objects help you implement real-time problems based on data abstraction, data encapsulation, data hiding, and polymorphism. We have briefly discussed all the 5 main concepts of object-oriented programming.

The OOP concepts are:

* **Data abstraction:** Data abstraction is an act of representing the important features of data without including the background details or the method applied to obtain it.
* **Data encapsulation:** Data encapsulation is nothing but a process to implement data abstraction by wrapping up the data and functions into an exclusive block.
* **Inheritance:** The term inheritance refers to transferring the properties of the parent class to the child class. We can implement the basic idea of inheritance by creating more than one class, which we formally refer to as derived classes by linking them with what we call the base class. This concept reduces the redundancy of the program and makes it easy to transfer/copy the properties of one class to another
* **Data hiding:** Data hiding refers to protecting data from unauthorized access. It is basically responsible for securing the data. It is important to note that data encapsulation is different from data hiding as encapsulation mainly focuses on shifting the focus on important data than explaining its complex nature.
* **Polymorphism:** The word poly means ‘many’ and morphism means ‘forms’. Clearly, polymorphism refers to displaying that data in more than one form.

2. Platform or Machine Independent/ Portable

In simple terms, portability refers to using the same piece of code in varied environments.

Let us understand this C++ feature with the help of an example. Suppose you write a piece of code to find the name, age, and salary of an employee in Microsoft Windows and for some apparent reason you want to switch your operating system to LINUX. This code will work in a similar fashion as it did in Windows.

3. Simple

When we start off with a new language, we expect to understand in depth. The simple context of [C++](https://en.wikipedia.org/wiki/C%2B%2B) gives an appeal to programmers, who are eager to learn a new programming language.

If you are already familiar with C, then you don’t need to worry about facing any trouble while working in C++. The syntax of C++ is almost similar to that of C. Afterall C++ is referred to as “C with classes”.

4. High-level programming language

It is important to note that C++ is a high-level programming language, unlike C which is a mid-level programming language. It makes it easier for the user to work in C++ as a high-level language as we can closely associate it with the human-comprehensible language, that is, English.

5. Popular

After learning C, it is the base language for many other popular programming languages which supports the feature of object-oriented programming. Bjarne Stroustrup found Simula 67, the first object-oriented language ever, lacking simulations and decided to develop C++.

6. Case sensitive

Just like C, it is pretty clear that the C++ programming language treats the uppercase and lowercase characters in a different manner. For instance, the meaning of the keyword**‘cout’** changes if we write it as **‘Cout’** or **“COUT”**. Other programming languages like HTML and MySQL are not case sensitive.

7. Compiler-Based

Unlike Java and Python that are interpreter-based, C++ is a compiler-based language and hence it a relatively much faster than Python and Java.

8. DMA (Dynamic Memory Allocation)

Since C++ supports the use of pointers, it allows us to allocate memory dynamically. We may even use constructors and destructors while working with classes and objects in C++.

9. Existence of Libraries

The C++ programming language offers a library full of in-built functions that make things easy for the programmer. These functions can be accessed by including suitable header files.

10. Speed

As discussed earlier, C++ is compiler-based hence it is much faster than other programming languages like Python and Java that are interpreter-based.

* **Hardware Implementation**

Hardware is a physical component of computer system which is visible by naked eyes.

The hardware that is used in my program are -

* Keyboard
* Display screen
* Mouse
* Hard disk
* RAM

**RAM:**

RAM stands for Random Access Memory. It is primary memory and volatile i.e. the data is lost if the system is turn off. It is costly component and of two types:

1. SRAM
2. DRAM

RAM required for this mini project is at least 10MB.

**ROM:**

ROM stands for Read Only Memory. It is secondary memory and non-volatile i.e. the data is not lost if the system is turn off. It is cheap and of two types:

* EROM
* EPROM
* ROM required for this mini project is at least 10MB.
* **Operating System**

It is an interface between user and computer system. It is a computer program that organises a number of other programs at the same time. It is system software that manages the computer hardware, software resources and provides common services for computer programs.

Operating system required for this mini project will not older than version Windows 7.

**SOFTWARE DESCRIPTION**

Dev-C++ is a [free](https://en.wikipedia.org/wiki/Free_software) full-featured [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) distributed under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) for programming in [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). It is written in [Delphi](https://en.wikipedia.org/wiki/Delphi_(programming_language)).

It is bundled with, and uses, the [MinGW](https://en.wikipedia.org/wiki/MinGW) or [TDM-GCC](https://en.wikipedia.org/wiki/TDM-GCC) 64bit port of the [GCC](https://en.wikipedia.org/wiki/GNU_Compiler_Collection) as its compiler. Dev-C++ can also be used in combination with [Cygwin](https://en.wikipedia.org/wiki/Cygwin) or any other GCC-based compiler.

Dev-C++ is generally considered a [Windows-only](https://en.wikipedia.org/wiki/Microsoft_Windows) program, but there are attempts to create a Linux version: header files and path delimiters are switchable between platforms

An additional aspect of Dev-C++ is its use of DevPaks: packaged extensions on the programming environment with additional libraries, templates, and utilities. DevPaks often contain, but are not limited to, [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) utilities, including popular toolkits such as [GTK+](https://en.wikipedia.org/wiki/GTK%2B), [wxWidgets](https://en.wikipedia.org/wiki/WxWidgets), and [FLTK](https://en.wikipedia.org/wiki/FLTK). Other DevPaks include libraries for more advanced function use. Users of Dev-C++ can download additional libraries, or packages of code that increase the scope and functionality of Dev-C++, such as graphics, compression, animation, sound support and many more. Users can create Devpaks and host them for free on the site. Also, they are not limited to use with Dev-C++ - the site says "A typical devpak will work with any MinGW distribution (with any IDE for MinGW)".

**PROGRAM (Using C++)**

#include<iostream>

using namespace std;

int r=0,p=0,k=20;

float TotalSale=0;

class bus{

char PessengerName[99];

char deprt[9];

char from[9];

char to[9];

char arriva[9];

int T4tutorials\_Total\_Seats, T4tutorials\_Total\_Fare;

int BusNo, T4tutorials\_Departure\_Timing, s;

public:

void T4tutorials\_Bus\_Registration();

void show();

void book();

void T4tutorialsExit();

}b[8];

void bus::T4tutorials\_Bus\_Registration(void)

{

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"\nEnter Bus No:";

cin>>BusNo;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"\nFrom: ";

cin>>from;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"\nTo: ";

cin>>to;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"\nDeparture: ";

cin>>T4tutorials\_Departure\_Timing;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"\nTotal seats: ";

cin>>T4tutorials\_Total\_Seats;

p++;

cout<<"\n";

cout<<"Bus Information Added!!\n";

system("PAUSE");

system("CLS");

}

void bus::show(void)

{

cout<<"\nTotal bus available:"<<endl;

for(int i=0;i<p;i++)

{ cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"Bus No: ";

cout<<b[i].BusNo<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"From: ";

cout<<b[i].from<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"To: ";

cout<<b[i].to<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"Departure: ";

cout<<b[i].T4tutorials\_Departure\_Timing<<" O'clock"<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(15);

cout<<"Seats: ";

cout<<b[i].T4tutorials\_Total\_Seats<<endl<<endl<<endl;

}

system("PAUSE");

system("CLS");

}

void bus::book(void)

{

int number;

float T4tutorials\_Total\_Fare;

cout<<"\nEnter Bus No: ";

cin>>number;

int n;

for(n=0;n<p;n++)

{

if(b[n].BusNo==number)

{

if(b[n].T4tutorials\_Total\_Seats<=0)

{

cout<<"\tSORRY!"<<endl<<"\tNo Seat Available\t";

}

else

{

cout<<endl<<"Total seat available: "<<b[n].T4tutorials\_Total\_Seats;

cout<<endl<<"Enter Passenger's Name: ";

cin>>PessengerName;

cout<<endl<<"Number of seats: ";

cin>>s;

while((b[n].T4tutorials\_Total\_Seats=b[n].T4tutorials\_Total\_Seats-s)<0)

{cout<<endl<<"Limit Exceed...Please re-enter ";

b[n].T4tutorials\_Total\_Seats=b[n].T4tutorials\_Total\_Seats+s;

cin>>s;

}

cout<<endl<<"Your purchase is completed"<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(30);

cout<<"Bus No: ";

cout<<b[n].BusNo<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(30);

cout<<"From: ";

cout<<b[n].from<<" to "<<b[n].to<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(30);

cout<<"Departure: ";

cout<<b[n].T4tutorials\_Departure\_Timing<<" O'clock"<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(30);

cout<<"Total seat: ";

cout<<s<<endl;

T4tutorials\_Total\_Fare=30\*s;

T4tutorials\_Total\_Fare=T4tutorials\_Total\_Fare+(T4tutorials\_Total\_Fare\*.28);

TotalSale=TotalSale+T4tutorials\_Total\_Fare;

cout.setf(ios::right,ios::adjustfield);

cout.width(30);

cout<<"Total Fare: ";

cout.setf(ios::showpoint);

cout.precision(3);

cout.setf(ios::fixed,ios::floatfield);

cout.width(8);

cout<<T4tutorials\_Total\_Fare<<endl;

cout.setf(ios::right,ios::adjustfield);

cout.width(30);

cout<<"Thank You!!!"<<endl<<endl;

}

}

}

system("PAUSE");

system("CLS");

}

int main()

{

int w,g=1;

while(g){

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Bus Reservation System\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"by KARAN SINGH,SAUD AKHTAR KHAN & YASH AGRAWAL\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl; cout<<"1. Bus Registration\n2. List of Available Bus\n3. Book Tickets\n4. Exit";

cout<<"\n\nEnter your choice: ";

cin>>w;

switch(w){

case 1:

b[p].T4tutorials\_Bus\_Registration();

break;

case 2:

b[0].show();

break;

case 3:

b[p].book();

break;

case 4:

{

g=0;

cout<<endl<<"Total Sales:"<<TotalSale<<endl;

break;

}

}

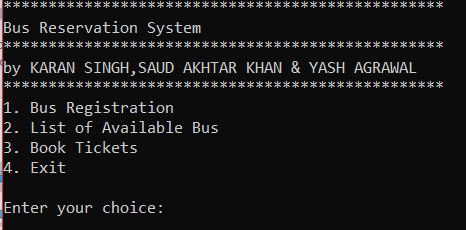
}

return 0;

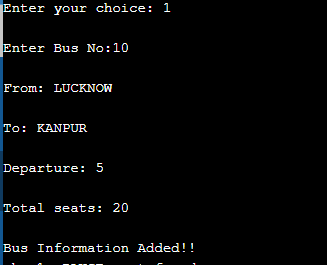
}

**PROJECT OVERVIEW**

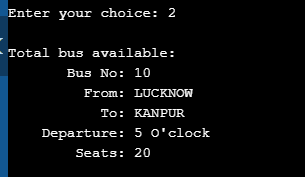
* ***OPENING VIEW***



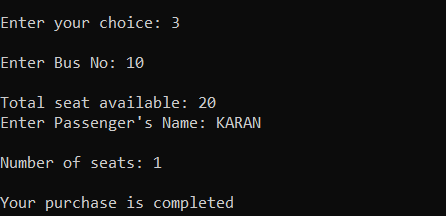
* ***BUS REGISTRATION***



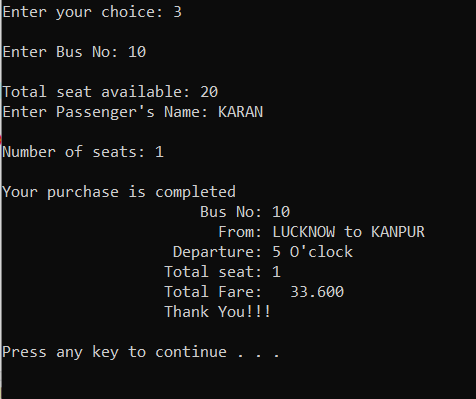
* ***AVAILABILITY OF BUS***



* ***TICKET BOOKING***



* ***TICKET INVOICE***



**CONCLUSION AND**

**RECOMMENDATIONS**

Online ticket booking system is an application where the customer can book a ticket online and 24\*7 hours a day from anyplace in the world. Customers can also interact with the ticket booking website to know any other details they want. Online ticket booking system has been developed successfully. System performance is also found to be satisfactory. This is a user-friendly application. Through this application, the cost can be reduced and efficiency is increased. There are several procedures that can be selected by customers. With the help of this application customers can book tickets.

**REFERENCES**

1. “A tour of C++” by Bjarne Stroustrup
2. [www.w3school.com](http://www.w3school.com)