

OBJECT ORIENTED PROGRAMMING

PROJECT REPORT



➤ **Group Leader:**

- **SYED ASAD ABBAS (1534-2020)**

➤ **Members:**

- I. SANIA ISHAQUE (1836-2020)**
- II. KIRAN KHALID (1687-2020)**

❖ **SUBMITTED TO: MISS SADIA MOOQADDAS**

JANUARY 11, 2022

BSCS

3rd SEMESTER

➤ **PROJECT NAME:**

MOVIE TICKET BOOKING SYSTEM

➤ **INTRODUCTION:**

The project we have chosen is a Movie Ticket Booking System. This project is based upon the concept of Object-Oriented Programming. The program is in C++ language. The program strictly follows the instruction that were instructed by our instructor. Moreover, the concepts that are being used in our project has been completely and clearly conveyed and taught by our class Teacher.

➤ **GOALS:**

We have tried to achieve the concepts of Inheritance, Encapsulation, Abstraction, Aggregation, Polymorphism, Overriding of Function, suitable use of access specifiers, Constructors, Setters and Getters, Exception and File Handling.

This Project will provide user the convenience of booking movie tickets from home. The Logics of this project can be used in E-Ticket. Moreover, through this project, the time consumption of booking tickets will be decreased. And user can view their booking at any time and also can delete their bookings.

This project can be modified in the future to add advanced features. We have tried to make this project easy to use and understand. The information of user is secured.

➤ **PROJECT INFO:**

In our project, we have used 3 classes. The names and information of classes are given below;

1. **class Ticket:** class Ticket is of a type **stand-alone**. This **stand-alone** class saves the data members in private and the data members are **static int ticket** and **int serial**.

The methods of this class are **Constructor** i.e., **Ticket()** and **int getTicket()**.

- i. **Ticket()**

This constructor of class Ticket initializes the data member of this class.

- ii. **getTicket()**

This method return the class member **serial**.

2. **class Booking:**

class Booking is of a type **Aggregated class**. This class is in the Aggregation to the class Ticket. This class Booking saves the data members in private and the data members are **Ticket *ticket, string name, string cnic, string gender** and **string phoneNo**. The methods of this class are **Constructors** i.e., **Booking()** and Parameterized Constructors **Booking(string name, string cnic, string gender, string phoneNo)**.

Moreover, the class contains methods of **getName(), getCnic(), getGender(), getPhoneNo(), setName(), setCnic(), setGender(), setPhoneNo()** and **void showConfirmedStatus()**.

3. **class MovieBooking:**

class MovieBooking is of a type **Inherited class**. This class is Inherited by the class **Booking**. This class **MovieBooking** saves the data member in private and the data member is **string mName**.

The data methods in this class are **Booking booking = Booking()** and Constructors which are **MovieBooking () : Booking ()** and

Parameterized Super Constructors **MovieBooking** (string name, string cnic, string gender, string phoneNo, string mName) : **Booking** (name, cnic, gender, phoneNo).

More methods of this class are **string getMovieName()** and **void showConfirmedStatus()**.

4. **class BookingSystem:**

class **BookingSystem** is of a type **Inherited class**. This class is Inherited by the class **MovieBooking**. This class **BookingSystem** saves the data member in private and the data member is **float ticketPrice**.

The data methods in this class are **Booking booking = Booking()** and Constructors which are **BookingSystem () : MovieBooking()** and Parameterized Super Constructors **BookingSystem(string name, string cnic, string gender, string phoneNo, string mName, float ticketPrice) : MovieBooking (name, cnic, gender, phoneNo, mName)**.

Methods of this class are **void setTicketPrice(float ticketPrice) , float getTicketPrice()** and **void showConfirmedStatus()**.

Note: The **void showConfirmedStatus()** function is the over-ridden function that we have use in class **Booking, MovieBooking** and **BookingSystem**.

➤ **MAIN FUNCTION:**

The main function will do the functioning to run the program. The functions in the **main()** will save and retrieve data to and from classes respectively.

The main function has the functioning of

1. **BookingSytem *setTicket ()**

2. **int mainMenu (vector <BookingSytem *> *bookingList)**

3. **void bookTicket (vector <BookingSytem *> *bookingList)**
4. **void deleteTicket (vector <BookingSytem *> *bookingList)**
5. **void searchBooking (vector <BookingSytem *> bookingList)**
6. **bool viewTicket (vector <BookingSytem *> bookingList)**
7. **void movieList ()**
8. **string chooseMovie (string selectMovie)**
9. **float TicketPrice()**

➤ **FUNCTIONING OF FUNCTIONS USED IN MAIN () :**

1. BookingSytem *setTicket () :

This function ***setTicket ()** is a type of **class BookingSystem**. In this function, we are taking all the required information of user to Book the Movie Ticket. This function is being called in the function **void bookTicket (vector <BookingSytem *> *bookingList)** from where the Booking starts. This Function acts as an object of class BookingSystem due to which we don't have to make class object. This function takes all the information from user and saves it in a constructor of class BookingSystem and returns its value.

2. int mainMenu (vector <BookingSytem *> *bookingList) :

This function acts as a Main Menu of the program. This function asks the user to choose the option that user needs to perform. If user presses 1 , the bookTicket function is called. Whereas if user enters 2, 3, 4, 5 and 6 then the function viewTicket, deleteTicket, movieList and exit is called respectively.

If user enters any other option from the options stated above, then the mainMenu function is called again.

3. void bookTicket (vector <BookingSytem *> *bookingList) :

This function calls the setTicket function by pushing the return value of setTicket in the vector *bookingList of type BookingSystem* .

It then asks the user to book another ticket or not.

4. void deleteTicket (vector <BookingSytem *> *bookingList) :

This function shows the bookings by calling viewTicket function and then asks the user to enter the serial number of the booking to delete that booking. When user enters the correct serial number, the function asks for the final confirmation of deleting the ticket. If user enters y, the ticket gets deleted and asks the user whether to delete another ticket or not. If user enters n, the program does not delete your Ticket function and then asks whether to go to main menu or exit the program.

5. void searchBooking (vector <BookingSytem *> bookingList):

This function provides user to find their booking. It asks the user to enter name of customer. If the customer's name exists, it shows the booking detail with customer name, serial number, movie name and ticket number. If the name doesn't exists in the data base, it shows that NO RECORD FOUND.

6. bool viewTicket (vector <BookingSytem *> bookingList) :

This function is used to show ticket bookings. This function takes the vector bookingList size, if size of vector bookingList is less than equal to zero, it shows the message that NO RECORD EXISTS. And in the either case it takes the information from the vector bookingList through a for loop and a vector function at() to get all the information of ticket booking.

7. void movieList () :

This function shows the Movie list. The interface shows 1. Pakistani Movies, 2. Hollywood Movies and 3. Animated Movies. The user enters the desired option and program opens that list, moreover the program asks whether to go back or to continue.

8. string chooseMovie (string selectMovie) :

This function shows the Movie list. The interface shows 1. Pakistani Movies, 2. Hollywood Movies and 3. Animated Movies. The user enters the desired option and program opens that list, moreover the program asks the user to select the serial number of movie the user wants to book.

9. float TicketPrice() :

This function shows the options 1. Premium Screen, 2. Gold Screen and 3. Silver Screen. If user enters 1, the Premium screen is selected and the price of ticket is set to be 2000 Rs. If user enters 2, the ticket price is set to be 1500 Rs. And if user enters 3, the ticket price is set to be 1000 Rs.

❖int main () :

The function int main () do the main functioning. In the int main() function, we have declared vector bookingList of type BookingSystem.

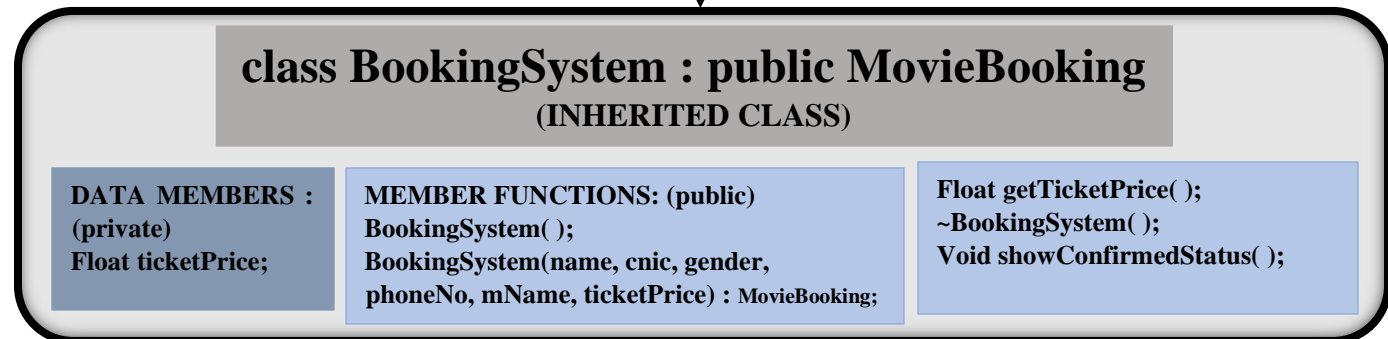
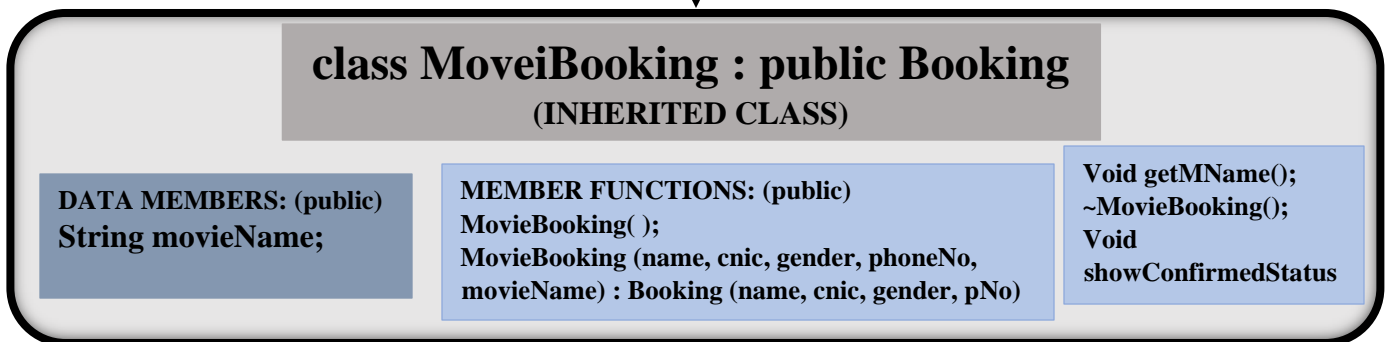
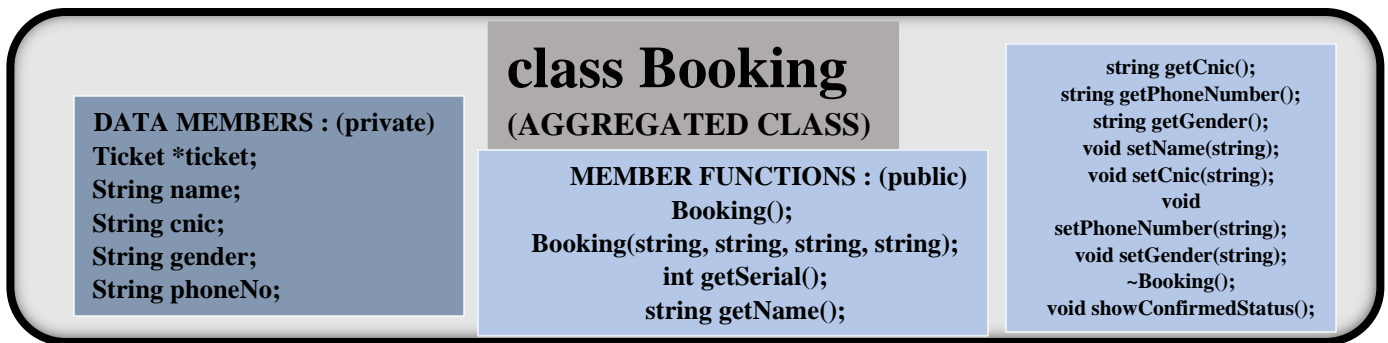
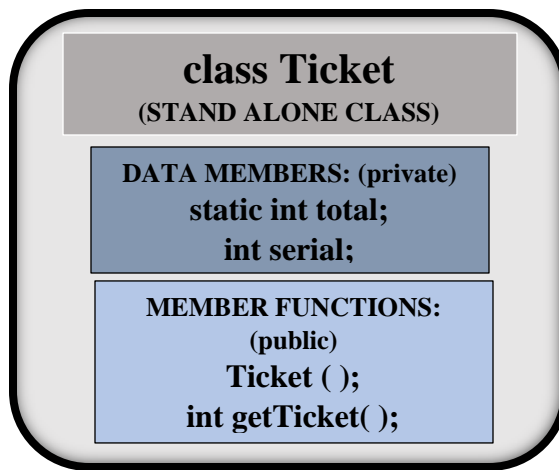
Then we have used file handling in the main() function to save the data of Movie Ticket Booking Details. We have made a file object of **fstream** as **textFile**.

Then we have opened the file to read all the saved detail in the text file. The textFile enters all the detail in char ch. Then we read the char ch in the format we have saved all the detail.

Then the mainMenu function is called for all the operations and after completion of all the function and when the user ends the program, we again come in the main () function for the final time and we open the textFile to write all the detail of the Movie Ticket Booking in the format we want to save in the text file and then we close the file and the program ends here.

➤ CLASSES HIERARCHY:

The Classes Hierarchy is given below;



MULTI-LEVEL INHERITANCE IS USED

➤ **CONCEPTS USED IN THE PROJECT:**

The project comprises of the following Logics/Methods/Concepts;

1. **Inheritance**
2. **Abstraction**
3. **Encapsulation**
4. **Function Over-riding**
5. **Polymorphism**
6. **Exception**
7. **File-handling**
8. **Vector**

Here we used vector instead of list and array. We used vector because in arrays, the arrays take the memory for the elements we initially provide. Although we don't use those spaces completely but it takes that space. Whereas in list, we need an iterator to read or write in the list. So in vector, the memory is used when we want it to be used. As we enter or delete any variable in a vector the memory is used and cleared respectively. Also, we don't need an iterator to read or write in and from the vector.

To use vector, we need to use the library `#include <vector>` to get all the functions of vector in our program.

➤ **LIBRARIES USED IN PROJECT:**

We have used several Libraries in our project which are given below;

1. **`#include <iostream>`**
2. **`#include <cstring>`**
3. **`#include <fstream>`**
4. **`#include <vector>`**
5. **`#include "bookingsys.h"`**
6. **`#include "bookingsys.cpp"`**
7. **`#include "booking.cpp"`**
8. **`#include "mbooking.cpp"`**

9. **#include “ticket.cpp”**

We have also used **using namespace std;** for cout, cin and endl commands.

➤ **CONCLUSION:**

By following the given parameters, we have met all the requirements of our project and we will try to implement more concepts based on OOP. This will conclude our 3rd Semester's OOP project.