# Chittagong University of Engineering and Technology

## Department of Electrical and Electronic Engineering


Report for the project titled “CUET Student Bus Seat Reservation System”

Course No: CSE 224

Course Title: Object Oriented Programming Sessional

Report prepared for:

Afroza Akter

Lecturer, Department of CSE

Report Prepared by:

Tahamidul Islam Samin , ID: 2102057

Tahsin Tajwar, ID: 2102026

SM Sazidur Rahman, ID: 2102058

# CUET Student Bus Seat Reservation System

Tahamidul Islam Samin[1], Tahsin Tajwar[2], SM Sazidur Rahman[3]

[1,2,3]*Department of Electrical and Electronic Engineering,*

*Chittagong University of Engineering and Technology,*

*Raozan, Chittagong- 4349, Bangladesh*

[1]u2102057@student.cuet.ac.bd

[2]u2102026@student.cuet.ac.bd

[3]u2102058@student.cuet.ac.bd

## Introduction:

This “CUET Student Bus Seat Reservation System” is a basic OOP application designed to help students at Chittagong University of Engineering & Technology (CUET) easily book seats on student buses. This system makes the seat reservation process simple and organized.

Main features of the system include:

1) Viewing Seat availability: students can check which seats are available before booking.

2) Boking multiple seats by one student.

3) Saving the reservation data.

4) Support for Multiple Users: The system can handle bookings from several students at the same time.

We have used C++ and it’s object-oriented programming feature to manage buses and students data. Overall, the CUET Student Bus Seat Reservation System helps students reserve bus seats conveniently, making transportation management smoother for CUET transportation authority.

## Motivation:

The everyday hustle after classes at CUET is a familiar scene, as students gather at the bus stand to head back to the city. Amidst the rush, the process of reserving bus seats, where students mark their territory by placing cards, papers, or bags on seats, reserving multiple seats for friends. This made us think: why not make a system that works exactly like this? This approach to seat reservation inspired us

to develop a system that replicates this real-life process in a digital environment.


Our motivation stems from the desire to create a user-friendly, object-oriented seat reservation system that resonates with the practicalities of student life at CUET. By emulating the familiar method of seat reservation, we aim to simplify the booking process and enhance the overall experience for students. Our goal is to make it easy for students to book seats, just like they do in real life. This way, getting on the bus after class will be smoother for everyone.

**Objectives:**

1. Replicating Real-Life Booking: To create a system that works like students reserving bus seats in real life.

2. Easy Interface: To makee it simple for students to book seats, just like they do with cards or bags.

3. Organized Code: Using OOP to keep the code neat and easy to manage.

4. Seat Availability: To ebsure students can book multiple seats and follow gender-specific seating rules.

5. Efficient Process: To make sure booking seats is quick and accurate, without errors.

6. Storing Booking Data: Saving booked seats data in a file for reference.

7. Handling Errors: Dealing with any mistakes in booking or invalid inputs smoothly.

8. User Experience: To improve how students interact with the system, making it straightforward and informative.

9. Prepare for Future Changes: Designing the system to grow and adapt to any future updates or needs.

**Methodology:**

Bus seats index and structure in this project (driver seat excluded):

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | | 31 | 32 |
| 4 | 5 | 6 | | 33 | 34 |
| 7 | 8 | 9 | | 35 | 36 |
| 10 | 11 | 12 | | 37 | 38 |
| 13 | 14 | 15 | | 39 | 40 |
| 16 | 17 | 18 | | 41 | 42 |
| 19 | 20 | 21 | | 43 | 44 |
| 22 | 23 | 24 | | 45 | 46 |
| 25 | 26 | 27 | | 47 | 48 |
| 28 | 29 | 30 | | 49 | 50 |

Overall Flow of the whole program:

1. Starts:

Begins the program execution.

2. Initializes buses:

Creates 7 student bus objects: Meghna, Bhuriganga, Rupsha, Surma, Matamuhuri, Shangu, Turag by calling the Bus class constructor.

3. Gets the number of students or users:

Prompts the user for the number of students booking seats.

4. Students loop:

Loops through the number of students (numStudents times).

Inside the Loop:

- Collects Student Info:

Prompts and gets student details: name, student ID, gender to create an object for every user.

- Chooses Bus:

Prompt the user to select a bus from the list.

- Creates Student Object:

Creates a Student object with the collected details.

- Displays Student Info:

Displays the student's information.

- Booking Seats:

Calls the bookSeats method on the Student object, passing the chosen Bus object.

Inside the bookSeats Method:

- Displays Seat Availability:

Displays the current seat availability of the selected bus.

- Gets the number of seats to book:

Prompts and gets the number of seats the student wants to book.

- Seat looking loop:

Loops through the number of seats to book.

Inside the Loop:

- Gets seat number:

Prompts and gets the seat number.

- Validates the seat number:

Checks if the seat number is valid (1 to 50).

- Checks seat occupancy:

Checks if the seat is already occupied.

- Checks gender restrictions:

For male students, ensures they do not book seats reserved for females (1-9 and 31-36).

- Books seat:

If valid and unoccupied, books the seat by bookSeat method and adds to the booked seats list.

- Saves Booking Info:

Saves the booked seats information to the file and displays the booked seats.

- Displays Updated Availability

Displays the updated seat availability.

5. End of students/users loop:

Ends the loop for all students.

6. Saving and closing the reservation file:

Saves the information about available seats to the file.

7. End:

End of the program.

**Implementation:**

Module Description:

1. The "Student" Class:

Attribues:

a) studentID,
b) gender
c) name.

```
public:
    string studentID;
    string gender;
    string name;
```

Methods/Functions:

a) bookSeats

```
// Method 1: to book multiple seats on a bus. Will define it outside the class later.
void bookSeats(Bus& bus);
```

Takes a Bus class's object as an argument.

2. The "Bus" Class:

Attribues:

a) name
b) seats

```
public:
    string name;
    vector<bool> seats;
```

Methods/Functions:

a) bookSeat:

```
// Method 1: to book only one seat per call
void bookSeat(int seatNumber) {
    seats[seatNumber] = true;
}
```

Takes an integer as an arguement.

b) isSeatOccupied:

```
// Method 2: to check if a seat is occupied or not
bool isSeatOccupied(int seatNumber) const {
    return seats[seatNumber];
}
```

Takes an integer as an arguement.

c) displaySeatAvailability:

```
// Method 3: to display seat availability
void displaySeatAvailability() const {
```

3. The Main Funnction:

```
int main()
```

Creates objects and calls constructor, methods according to program logic.

Output/User Experience:

User 1:

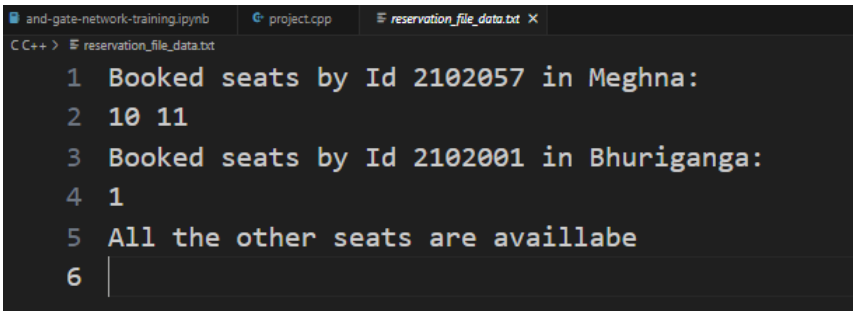
```
Welcome to the CUET Student Bus Seat Reservation System!
How many users/students are here to book seat?
2
Colecting info for user/student no 1
Choose a bus from Meghna(1), Bhuriganga(2), Rupsha(3), Surma(4), Matamuhuri(5), Shangu(6), Turag(7)
1
Enter your name: Samin
Enter your student ID: 2102057
Enter your gender (Male/Female): Male
Name: Samin
Student ID: 2102057
Gender (Male/Female): Male
Seat availability for bus Meghna:
Seat 1: Available
Seat 2: Available
Seat 3: Available
Seat 4: Available
Seat 5: Available
Seat 6: Available
Seat 7: Available
Seat 8: Available
Seat 9: Available
Seat 10: Available
```

```
How many seats do you want to book?
2
Enter specific seat numbers to book (1 to 50):
1
Seat from 1 to 9 and 31 to 36, these seats are for females only. You should be ashamed.
DSW must be informed!!!
Choose another seat no. please, which is valid for male students!
10
11
Booked seats by Id 2102057 in Meghna:
10 11
Bus seat reservation data updated
Seat availability for bus Meghna:
Seat 1: Available
Seat 2: Available
Seat 3: Available
Seat 4: Available
Seat 5: Available
Seat 6: Available
Seat 7: Available
Seat 8: Available
Seat 9: Available
Seat 10: Occupied
Seat 11: Occupied
```

User 2:

```
Colecting info for user/student no 2
Choose a bus from Meghna(1), Bhuriganga(2), Rupsha(3), Surma(4), Matamuhuri(5), Shangu(6), Turag(7)
2
Enter your name: Femmale
Enter your student ID: 2102001
Enter your gender (Male/Female): Female
Name: Femmale
Student ID: 2102001
Gender (Male/Female): Female
Seat availability for bus Bhuriganga:
Seat 1: Available
Seat 2: Available
Seat 3: Available
Seat 4: Available
Seat 5: Available
Seat 6: Available
Seat 7: Available
Seat 8: Available
```

Data saved in a txt file:




**Conclusion:**

By using object-oriented programming principles, we have created a user-friendly system that allows students to book bus seats in a manner similar to their daily routine at CUET. Using classes and methods, we have encapsulated the details of students and buses, making the code modular, maintainable, and easy to understand. The system enforces rules such as seat availability and gender-specific seating areas, ensuring fairness and compliance with university guidelines. Overall, this project demonstrates the power of object-oriented programming in solving practical problems and improving everyday life.