Train Coach Seat Reservation System

Problem Description:

1. There are 80 seats in a coach of a train with only 7 seats in a row and last row of only 3 seats. For simplicity, there is only one coach in this train.

2. One person can reserve up to 7 seats at a time.

3. If a person is reserving seats, the priority will be to book them in one row.

4. If seats are not available in one row, then the booking should be done in such a way that the nearby seats are booked.

5. User can book as many tickets as s/he wants until the coach is full.

6. You don't have to create login functionality for this application.

How it should function?

1. Input required will only be the required number of seats. Example: 2 or 4 or 6 or 1 etc.

2. Output should be seats numbers that have been booked for the user along with the display of all the seats and their availability status through color or number or anything else that you may feel fit.

Source Code:

```
import java.util.Scanner;


public class TrainCoachBooking {
   private static final int TOTAL_ROWS = 11; // 10 rows
with 7 seats and 1 row with 3 seats
   private static final int SEATS_PER_ROW = 7;
   private static final int LAST_ROW_SEATS = 3;
   private static final int MAX_BOOKING = 7;


   private static boolean[][] seats = new
boolean[TOTAL_ROWS][SEATS_PER_ROW];
   private static boolean[] lastRowSeats = new
boolean[LAST_ROW_SEATS];


   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);


      while (true) {
         System.out.println("Enter the number of seats
to book (0 to exit): ");
         int numSeats = scanner.nextInt();
```

```
            if (numSeats == 0) {
                break;
            }

            if (numSeats > MAX_BOOKING) {
                System.out.println("You can book up to " +
MAX_BOOKING + " seats at a time.");
                continue;
            }

            bookSeats(numSeats);
            displaySeats();
        }
    }

    private static void bookSeats(int numSeats) {
        int booked = 0;

        // Try to book seats in one row
        for (int i = 0; i < TOTAL_ROWS - 1; i++) {
          int available = getAvailableSeatsInRow(i);
          if (available >= numSeats) {
             bookSeatsInRow(i, numSeats);
             booked = numSeats;
             break;
          }
       }

       // If not enough seats available in one row, book
nearby seats
       if (booked < numSeats) {
          int remaining = numSeats - booked;
          bookNearbySeats(remaining);
       }
    }

    private static int getAvailableSeatsInRow(int row) {
       int available = 0;
       for (int i = 0; i < SEATS_PER_ROW; i++) {
          if (!seats[row][i]) {
             available++;
```

```
            }
        }
        return available;
    }

    private static void bookSeatsInRow(int row, int
numSeats) {
        for (int i = 0; i < SEATS_PER_ROW; i++) {
            if (!seats[row][i]) {
                seats[row][i] = true;
                numSeats--;
                if (numSeats == 0) {
                    break;
                }
            }
        }
    }

    private static void bookNearbySeats(int numSeats) {
        for (int i = 0; i < TOTAL_ROWS - 1; i++) {
            for (int j = 0; j < SEATS_PER_ROW; j++) {
```

```
            if (!seats[i][j]) {
                seats[i][j] = true;
                numSeats--;
                if (numSeats == 0) {
                    return;
                }
            }
        }
    }

    // Book seats in the last row
    for (int i = 0; i < LAST_ROW_SEATS; i++) {
        if (!lastRowSeats[i]) {
            lastRowSeats[i] = true;
            numSeats--;
            if (numSeats == 0) {
                return;
            }
        }
    }
}
```

```
    private static void displaySeats() {

        System.out.println("Coach Seats:");

        for (int i = 0; i < TOTAL_ROWS - 1; i++) {

            for (int j = 0; j < SEATS_PER_ROW; j++) {

                System.out.print(seats[i][j] ? "X " : "O ");

            }

            System.out.println();

        }


        System.out.print("Last Row: ");

        for (int i = 0; i < LAST_ROW_SEATS; i++) {

            System.out.print(lastRowSeats[i] ? "X " : "O ");

        }

        System.out.println();

    }

}
```

Output:

```
Enter the number of seats to book (0 to exit):

2
```

```
Coach Seats:

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

Last Row: O O O

Enter the number of seats to book (0 to exit):

3

Coach Seats:

O O X X O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O
```

```
O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

Last Row: O O O

Enter the number of seats to book (0 to exit):

4

Coach Seats:

O O X X O O O

O O O X X X O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

Last Row: O O O

Enter the number of seats to book (0 to exit):

5
```

```
Coach Seats:
O O X X O O O
O O O X X X O
O O O X X X O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
Last Row: O O O
Enter the number of seats to book (0 to exit):
7
Coach Seats:
O O X X O O O
O O O X X X O
O O O X X X O
O O O X X X O
O O O O O O O
```

```
O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

Last Row: O O O

Enter the number of seats to book (0 to exit):

1

Coach Seats:

O O X X O O O

O O O X X X O

O O O X X X O

O O O X X X O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

Last Row: X O O
```

```
Enter the number of seats to book (0 to exit):

2

Coach Seats:

O O X X O O O

O O O X X X O

O O O X X X O

O O O X X X O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

O O O O O O O

Last Row: X X O

Enter the number of seats to book (0 to exit):

1

Coach Seats:

O O X X O O O

O O O X X X O

O O O X X X O
```

```
O O O X X X O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
Last Row: X X X
Enter the number of seats to book (0 to exit):
0
```

Expiation of the Source Code:

Main Method:

The main method is the entry point of the program. It creates a Scanner object to read input from the user and enters an infinite loop to continuously prompt the user to book seats.

Booking Seats:

The BookSeats method is called when the user wants to book seats. It takes the number of seats to book as an argument. The method tries to book seats in one row first, and if not enough seats are available, it books nearby seats.

Booking Seats in One Row:

The bookSeatsInRow method is called when there are enough available seats in one row. It books the specified number of seats in the row by setting the corresponding elements in the seats array to true.

Booking Nearby Seats:

The bookNearbySeats method is called when there are not enough available seats in one row. It books the remaining seats by iterating through the seats array and setting the first available seats to true. If there are still remaining seats, it books seats in the last row.

Displaying Seats:

The displaySeats method is called to display the current state of the coach seats. It prints the seats in each row, with 'X' representing booked seats and 'O' representing available seats.

Variables and Constants:

The program uses several variables and constants:

- TOTAL_ROWS: The total number of rows in the coach, including the last row with 3 seats.
- SEATS_PER_ROW: The number of seats in each row, except for the last row.
- LAST_ROW_SEATS: The number of seats in the last row.

- MAX_BOOKING: The maximum number of seats that can be booked at a time.
- seats: A 2D Boolean array representing the seats in the coach, where true indicates a booked seat and false indicates an available seat.
- lastRowSeats: A Boolean array representing the seats in the last row.

Overall, the program provides a simple implementation of a train coach booking system, allowing users to book seats and displaying the current state of the coach seats.