Java Project: Airport

Java Concepts Covered:

- Class definition
- Object-oriented programming (encapsulation)
- ArrayLists
- Method overloading
- Loops and conditionals
- Basic input/output

The project simulates a basic airport system where you can add flights, display available flights, and book seats.

```
import java.util.ArrayList;
import java.util.Scanner;

// Define the Flight class to represent flight information
class Flight {
    private String flightNumber;  // Unique identifier for the flight
    private int availableSeats;   // Number of available seats

    // Constructor to initialize a new Flight object
    public Flight(String flightNumber, int availableSeats) {
        this.flightNumber = flightNumber;
        this.availableSeats = availableSeats;
    }

    // Getter method to retrieve the flight number
    public String getFlightNumber() {
        return flightNumber;
    }

    // Getter method to retrieve the number of available seats
    public int getAvailableSeats() {
```

```
        return availableSeats;
    }

    // Method to book seats on the flight
    public void bookSeats(int seats) {
        if (seats <= availableSeats) {
            availableSeats -= seats;
            System.out.println("Successfully booked " + seats + " seats for flight " + flightNumber);
        } else {
            System.out.println("Not enough seats available.");
        }
    }
}

// Define the Airport class to manage multiple flights
class Airport {
    private ArrayList<Flight> flights;  // List to hold Flight objects

    // Constructor to initialize the Airport object
    public Airport() {
        flights = new ArrayList<>();
    }

    // Method to add a flight to the airport
    public void addFlight(String flightNumber, int availableSeats) {
        flights.add(new Flight(flightNumber, availableSeats));
    }

    // Method to display information about available flights
    public void displayAvailableFlights() {
```

```
        System.out.println("Available flights:");
        for (Flight flight : flights) {
            System.out.println("Flight Number: " + flight.getFlightNumber() + ", Available Seats: " +
flight.getAvailableSeats());
        }
    }

    // Method to book a flight by its number
    public void bookFlight(String flightNumber, int seats) {
        for (Flight flight : flights) {
            if (flight.getFlightNumber().equals(flightNumber)) {
                flight.bookSeats(seats);
                return;
            }
        }
        System.out.println("Flight not found.");
    }
}

// Main class that contains the main method
public class AirportApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);  // Scanner object to read input from the user
        Airport airport = new Airport();  // Create an Airport object

        // Add some initial flights to the airport
        airport.addFlight("AA101", 50);
        airport.addFlight("BB202", 30);
        airport.addFlight("CC303", 40);
```

```
        // Main menu loop
        while (true) {
            System.out.println("1. Display available flights");
            System.out.println("2. Book a flight");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();  // Read user choice

            // Execute action based on user choice
            switch (choice) {
                case 1:
                    airport.displayAvailableFlights();  // Display list of available flights
                    break;
                case 2:
                    System.out.print("Enter flight number to book: ");
                    String flightNumber = scanner.next();
                    System.out.print("Enter number of seats: ");
                    int seats = scanner.nextInt();
                    airport.bookFlight(flightNumber, seats);  // Book the flight
                    break;
                case 3:
                    System.out.println("Exiting...");  // Exit the application
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");  // Handle invalid input
            }
        }
    }
}
```