

Goal: Build a console app that simulates managing flight bookings for an airline. The app supports functions like viewing available flights, booking, canceling, and searching flights or passengers.

The system will be designed using **18 user-defined functions**, each serving a specific purpose, to help practice:

- No/optional parameters
- Return values
- Function overloading
- Named parameters
- ref and out parameters

Function Requirements for the Airline App (18 Total)

Startup & Navigation (1–4)

1. DisplayWelcomeMessage()

- No parameters, no return
- Shows a stylized welcome screen for the airline.

2. ShowMainMenu()

- No parameters, returns int
- Displays user options (Book Flight, Cancel, View Flights, Exit) and returns selected choice.

3. ExitApplication()

- No parameters, no return
- Prints a thank-you message and terminates program (or ends loop).

4. AddFlight(string flightCode, string fromCity, string toCity, DateTime departureTime, int duration)

- Inputs as named parameters, no return
- Adds a flight to a global list of available flights.

Flight and Passenger Management (5–8)

5. DisplayAllFlights()

- No parameters, no return
- Loops and prints all available flights.

6. FindFlightByCode(string code)

- Returns bool (found or not)
- Searches for flight with specific code.

7. UpdateFlightDeparture(ref DateTime departure)

- Uses ref to change departure time of a flight
- New time is input from the user.

8. CancelFlightBooking(out string passengerName)

- Prompts for booking ID and removes from list
- Returns passenger name via out.



Passenger Booking Functions (9 – 13)

9. **BookFlight(string passengerName, string flightCode = "Default001")**

- Optional parameter flightCode
- Adds passenger to booking list (default flight used if not specified).

10. **ValidateFlightCode(string flightCode)**

- Returns bool after checking if code exists in flights list.

11. **GenerateBookingID(string passengerName)**

- Returns booking ID string: e.g., "Ahmed2025"

12. **DisplayFlightDetails(string code)**

- Takes flight code, prints route and time info.

13. **SearchBookingsByDestination(string toCity)**

- Filters bookings where destination equals toCity.



Function Overloading (14 – 16)

14. **CalculateFare(int basePrice, int numTickets)**

- Returns total fare (basic version).

15. **CalculateFare(double basePrice, int numTickets)**

- Overloaded for float pricing (e.g., 799.99)

16. **CalculateFare(int basePrice, int numTickets, int discount)**

- Overloaded with discount included.



System Utilities & Final Flow (17 – 18)

17. **ConfirmAction(string action)**

- Asks: "Are you sure to [cancel booking]?" and returns true/false.

18. **StartSystem()**

- The main function that:

- Displays menu
- Accepts user choices
- Calls the relevant functions