

Evaluation test for week 06

Out of 5 solve any 3

Time: 2 Hours

1. Flight Seat Reservation System

Objective: Create a class to manage flight seat reservations. Each flight has a seat count and a record of bookings.

Class:

Flight

Attributes:

- Flight Number
- Total Seats
- Booked Seats

Methods:

- Constructor
- bookSeats(int)
- showAvailability()

Test Case 1:

```
Flight f1("AI203", 100);  
f1.bookSeats(30);  
f1.showAvailability();
```

Expected Output:

Flight: AI203 | Seats Available: 70

Test Case 2:

```
f1.bookSeats(80);
```

Expected Output:

Booking Failed: Not enough seats.

2. Hotel Room Booking Manager

Objective: Manage bookings for rooms in a hotel.

Class:

```
HotelRoom
```

Attributes:

- Room Number
- Type (AC/Non-AC)
- IsBooked (boolean)

Methods:

- Constructor
- `bookRoom()`
- `showStatus()`

Test Case 1:

```
HotelRoom h1(201, "AC");  
h1.bookRoom();  
h1.showStatus();
```

Expected Output:

```
Room 201 (AC) is now booked.
```

Test Case 2:

```
h1.bookRoom();
```

Expected Output:

```
Room already booked.
```

3. Food Delivery Tip Calculator

Objective: Calculate delivery tip based on food bill.

Class:

DeliveryTip

Attributes:

- Order ID
- Bill Amount
- Distance (km)
- Tip (calculated)

Methods:

- Constructor
- calculateTip()
- printDetails()

Tip Rule:

- < 5 km: 5% of bill
- 5-10 km: 10%
- > 10 km: 15%

Test Case 1:

```
DeliveryTip d1("ORD1", 500, 3);  
d1.printDetails();
```

Expected Output:

Order ORD1 | Tip: ₹25

Test Case 2:

```
DeliveryTip d2("ORD2", 1000, 12);  
d2.printDetails();
```

Expected Output:

Order ORD2 | Tip: ₹150

4. Luggage Weight Checker

Objective: Validate if luggage is overweight.

Class:

Luggage

Attributes:

- Passenger Name
- Weight (kg)
- Limit (default 20kg)

Methods:

- Constructor
- `checkOverweight()`

Test Case 1:

```
Luggage l1("Rita", 18);  
l1.checkOverweight();
```

Expected Output:

Luggage within limit.

Test Case 2:

```
Luggage l2("Sam", 25);  
l2.checkOverweight();
```

Expected Output:

Overweight! Exceeded by 5 kg.

5. Parcel Shipping Cost Estimator

Objective: Estimate shipping cost based on weight and distance.

Class:

Parcel

Attributes:

- Parcel ID
- Weight (kg)
- Distance (km)
- Cost

Cost Formula: ₹5/km for ≤5kg, else ₹8/km

Methods:

- Constructor
- calculateCost()
- printCost()

Test Case 1:

```
Parcel p1("PX001", 3, 50);  
p1.printCost();
```

Expected Output:

Parcel PX001 | Cost: ₹250

Test Case 2:

```
Parcel p2("PX002", 8, 50);  
p2.printCost();
```

Expected Output:

Parcel PX002 | Cost: ₹400
