



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF COMPUTER SCIENCE ENGINEERING
AND INFORMATION SYSTEMS**

FALL SEMESTER 2024-2025

PMCA502P – JAVA PROGRAMMING LAB

DIGITAL ASSIGNMENT – 1

SUBMITTED ON: 16 – AUG - 2024

SUBMITTED BY-

AKASH KUMAR BANIK

PROGRAM: MCA

REGISTER No.: 24MCA0242

QUESTION 1: AIRLINE RESERVATION SYSTEM

A small airline has just purchased the computer for its new automated reservations system. You have been asked to program the new system in Java to assign seats on each flight of the airline's two planes, each of capacity: 10.

Define a user defined class to represent the reservation details like passenger name, mobile number, flight number and reserved seat number.

Keep the flight details in two static String arrays for each flight. The first five seats (index 0 to 4) represent the First Class whereas the next five seats (index 5 to 9) represent the Economy Class. Initially, both the arrays should be assigned with the value Available through static block so, no booking has done. It should be updated as Reserved for each corresponding booking.

Define a static method to display the flight details. Sample is here:

Flight-1	Flight-2
1-Reserved	1-Reserved
2-Reserved	2-Reserved
3-Available	3-Available
4-Available	4-Available
5-Reserved	5-Available
6-Reserved	6-Available
7-Available	7-Available
8-Reserved	8-Available
9-Available	9-Available
10-Available	10-Available

Define a constructor with the parameters passenger name, mobile number, flight number and reserved seat number.

Create a static method booking for every reservation. It should get the flight number and travel class (First or Economy) as parameters. If the seat is available in the corresponding flight it

should return the seat number, otherwise -1. Also, the status of the corresponding flight seat should be updated as “Reserved” when it is available.

Create a non-static method to display the reservation details.

Create a demo class which contains main method. Declare array of objects with the size 20 to store the reservation details. Create a menu driven loop to do the following with the choices from 1 to 4.

1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a seat
4. Stop

The flight details should be displayed when the user press 1. The reservation details should be displayed when the user press 2. If the user press 3, the system should get the flight number and travel class as input. Then it should check the availability of the seat. If it is available, then the system collects the user name and mobile number. Now, it should create an object belonging to reservation class with complete details. Suppose the seat is not available, print the message “Next Flight leaves in 3 hours”.

Stop this iteration when user press 4. Display ‘choice is wrong, try again’ when user didn’t press the correct choice.

CODE:

```
import java.util.Scanner;

class Reservation {

    private String passengerName;

    private long mobileNumber;

    private int flightNumber, reservedSeatNumber;

    private static String[] flight1 = new String[10];

    private static String[] flight2 = new String[10];

    static {
```

```

        for (int i = 0; i < 10; i++) {

            flight1[i] = "Available";

            flight2[i] = "Available";

        }

    }

    public Reservation(String passengerName, long mobileNumber, int flightNumber, int
reservedSeatNumber) {

        this.passengerName = passengerName;

        this.mobileNumber = mobileNumber;

        this.flightNumber = flightNumber;

        this.reservedSeatNumber = reservedSeatNumber;

    }

    public static void displayFlightDetails() {

        System.out.println("\nFlight-1\tFlight-2");

        for (int i = 0; i < 10; i++) {

            System.out.println((i + 1) + "-" + flight1[i] + "\t" + (i + 1) + "-" + flight2[i]);

        }

    }

    public static int booking(int flightNumber, String travelClass) {

        int start = 0;

        int end = 0;

        if (travelClass.equalsIgnoreCase("First")) {

            start = 0;

            end = 5;

        } else if (travelClass.equalsIgnoreCase("Economy")) {

```

```

        start = 5;

        end = 10;

    } else {

        System.out.println("Invalid travel class.");

        return -1;

    }

    String[] flightSeats;

    if (flightNumber == 1) {

        flightSeats = flight1;

    } else if (flightNumber == 2) {

        flightSeats = flight2;

    } else {

        System.out.println("Invalid flight number.");

        return -1;

    }

    for (int i = start; i < end; i++) {

        if (flightSeats[i].equals("Available")) {

            flightSeats[i] = "Reserved";

            return i + 1;

        }

    }

    return -1;

}

public void displayReservationDetails() {

    System.out.println("Passenger Name: " + passengerName);

```

```

        System.out.println("Mobile Number: " + mobileNumber);

        System.out.println("Flight Number: " + flightNumber);

        System.out.println("Reserved Seat Number: " + reservedSeatNumber);

    }

}

public class AirlineReservationSystem {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Reservation[] reservations = new Reservation[20];

        int reservationCount = 0;

        while (true) {

            System.out.println("\n1. Display Flight Details");

            System.out.println("2. Display Reserved Passenger Details");

            System.out.println("3. Reserve a Seat");

            System.out.println("4. Stop");

            System.out.print("Enter your choice: ");

            int choice = sc.nextInt();

            switch (choice) {

                case 1:

                    Reservation.displayFlightDetails();

                    break;

                case 2:

                    if (reservationCount == 0) {

                        System.out.println("\nNo reservations made yet.");

                    } else {

```

```

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

            System.out.println("\nReservation " + (i + 1) + ":");

            reservations[i].displayReservationDetails();

            System.out.println();

        }

    }

    break;

case 3:

    System.out.print("\nEnter Flight Number (1 or 2): ");

    int flightNumber = sc.nextInt();

    System.out.print("Enter Travel Class (First or Economy): ");

    String travelClass = sc.next();

    int seatNumber = Reservation.booking(flightNumber, travelClass);

    if (seatNumber != -1) {

        sc.nextLine();

        System.out.print("Enter Passenger Name: ");

        String passengerName = sc.nextLine();

        System.out.print("Enter Mobile Number: ");

        long mobileNumber = sc.nextLong();

        reservations[reservationCount++] = new Reservation (passengerName,
mobileNumber, flightNumber, seatNumber);

        System.out.println("\nReservation confirmed. Seat Number: " + seatNumber);

    } else {

        System.out.println("\nFlight Full. Next Flight leaves in 3 hours.");

    }

}

```

```

        break;

    case 4:

        System.out.println("\nExiting the system. Thank you!");

        sc.close();

        return;

    default:

        System.out.println("\nInvalid choice. Try again!!");

    }

}

}

}

```

OUTPUT:

```
C:\Users\akash\OneDrive\Desktop\Java DA-1>javac AirlineReservationSystem.java
```

```
C:\Users\akash\OneDrive\Desktop\Java DA-1>java AirlineReservationSystem
```

```

1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 1

```

Flight-1	Flight-2
1-Available	1-Available
2-Available	2-Available
3-Available	3-Available
4-Available	4-Available
5-Available	5-Available
6-Available	6-Available
7-Available	7-Available
8-Available	8-Available
9-Available	9-Available
10-Available	10-Available

```

1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 2

No reservations made yet.

```



```
1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 3

Enter Flight Number (1 or 2): 1
Enter Travel Class (First or Economy): First
Enter Passenger Name: Gopu J Shekhar
Enter Mobile Number: 9876543210

Reservation confirmed. Seat Number: 1
```

```
1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 3

Enter Flight Number (1 or 2): 2
Enter Travel Class (First or Economy): Economy
Enter Passenger Name: Nishant Dhawale
Enter Mobile Number: 7410852963

Reservation confirmed. Seat Number: 6
```

```
1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 1

Flight-1      Flight-2
1-Reserved    1-Available
2-Available    2-Available
3-Available    3-Available
4-Available    4-Available
5-Available    5-Available
6-Available    6-Reserved
7-Available    7-Available
8-Available    8-Available
9-Available    9-Available
10-Available   10-Available
```

```
1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 2

Reservation 1:
Passenger Name: Gopu J Shekar
Mobile Number: 9876543210
Flight Number: 1
Reserved Seat Number: 1

Reservation 2:
Passenger Name: Nishant Dhawale
Mobile Number: 7410852963
Flight Number: 2
Reserved Seat Number: 6
```

```
1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 3

Enter Flight Number (1 or 2): 1
Enter Travel Class (First or Economy): First

Flight Full. Next Flight leaves in 3 hours.
```

```
1. Display Flight Details
2. Display Reserved Passenger Details
3. Reserve a Seat
4. Stop
Enter your choice: 4

Exiting the system. Thank you!
```

QUESTION 2: GEOMETRY SHAPE

Create a class to represent the geometry shape with the members co-ordinate x, coordinate y and color. Create constructor to initialize these members. Define a method to print the details. Create the following classes which inherit the property of geometry class:

- Class circle which contains radius and area as members. Create constructor to initialize the members except area. Area should be calculated as $\text{Pi} * \text{radius} * \text{radius}$. Print the details of the circle class.
- Class rectangle which contains length, width and area as members. Create constructor to initialize the members except area. Find the area as $\text{length} * \text{width}$. Print the details of rectangle class.

Here, overrides the base class method print in Circle and Rectangle class.

Design a Demo class and demonstrate these methods through object for each classes.

CODE:

```
class Geometry{

    int x,y;

    String color;

    public Geometry(int x,int y,String color){
```

```
        this.x=x;

        this.y=y;

        this.color=color;
    }

    public void printDetails(){

        System.out.println("Coordinate x: " +x);

        System.out.println("Coordinate y: " +y);

        System.out.println("Color: " +color);

    }

}

class Circle extends Geometry{

    float radius;

    double area;

    public Circle(int x,int y,String color,float radius){

        super(x,y,color);

        this.radius=radius;

        this.area= Math.PI * radius * radius;

    }

    public void printDetails(){

        super.printDetails();

        System.out.println("Shape: Circle");

        System.out.println("Radius: "+radius);

        System.out.println("Area of circle: "+area);

    }

}
```

```
class Rectangle extends Geometry {  
    float length,width;  
    double area;  
    public Rectangle(int x,int y,String color,float length,float width){  
        super(x, y, color);  
        this.length=length;  
        this.width=width;  
        this.area=length*width;  
    }  
    public void printDetails(){  
        super.printDetails();  
        System.out.println("Shape: Rectangle");  
        System.out.println("Length: "+length);  
        System.out.println("Width: "+width);  
        System.out.println("Area of Rectangle: "+area);  
    }  
}  
  
public class Demo {  
    public static void main(String[] args) {  
        Geometry obj = new Geometry(1, 2, "Green");  
        System.out.println("Geometry Details");  
        obj.printDetails();  
        System.out.println();  
        Circle circle_obj = new Circle(10,20,"Red",3);  
        System.out.println("Circle Details");  
    }  
}
```

```

        circle_obj.printDetails();

        System.out.println();

        Rectangle rect_obj = new Rectangle(30,40,"Blue",4,5);

        System.out.println("Rectangle Details:");

        rect_obj.printDetails();

    }
}

```

OUTPUT:

```

C:\Users\akash\OneDrive\Desktop\Java DA-1>javac Demo.java

C:\Users\akash\OneDrive\Desktop\Java DA-1>java Demo
Geometry Details
Coordinate x: 1
Coordinate y: 2
Color: Green

Circle Details
Coordinate x: 10
Coordinate y: 20
Color: Red
Shape: Circle
Radius: 3.0
Area of circle: 28.274333882308138

Rectangle Details:
Coordinate x: 30
Coordinate y: 40
Color: Blue
Shape: Rectangle
Length: 4.0
Width: 5.0
Area of Rectangle: 20.0

```

QUESTION 3: BANK SYSTEMS

Create a Java program to achieve the dynamic method dispatch through the following scenario.

- Create a class RBI which couldn't be instantiated, only for inheritance and insist the sub classes to do the processes like getting input, printing details and calculate the interest for the balance amount in the account.
- Create a class to represent customers of Indian Bank, ICICI and CUB with account number, account holder name, balance amount, interest amount and rate of interest as

static member which hold the interest percentage of each banks. Initiate the rate of interest through the static block.

- c. All these bank classes should implement the methods mentioned in the super class RBI.

Design a demo class to create object for each class and demonstrate the methods.

CODE:

```
abstract class RBI {  
    long accNo;  
  
    String accName;  
  
    double balanceAmt;  
  
    double interestAmt;  
  
    abstract void getInput(long accNo, String accName, double balanceAmt);  
  
    abstract void calculateInterest();  
  
    abstract void printDetails();  
}  
  
class IndianBank extends RBI {  
    static double interest;  
  
    static {  
        interest = 5.0;  
    }  
  
    void getInput(long accNo, String accName, double balanceAmt) {  
        this.accNo = accNo;  
  
        this.accName = accName;  
  
        this.balanceAmt = balanceAmt;  
    }  
  
    void calculateInterest() {
```

```

        interestAmt = (balanceAmt * interest)/100;
    }

    void printDetails() {

        System.out.println("Bank: Indian Bank");

        System.out.println("Account Number: " +accNo);

        System.out.println("Account Holder Name: " +accName);

        System.out.println("Balance Amount: " +balanceAmt);

        System.out.println("Interest Amount: " +interestAmt);

    }

}

class ICICI extends RBI {

    static double interest;

    static {

        interest = 6.0;

    }

    void getInput(long accNo, String accName, double balanceAmt) {

        this.accNo = accNo;

        this.accName = accName;

        this.balanceAmt = balanceAmt;

    }

    void calculateInterest() {

        interestAmt = (balanceAmt * interest)/100;

    }

    void printDetails() {

        System.out.println("Bank: ICICI");

```

```

        System.out.println("Account Number: " +accNo);

        System.out.println("Account Holder Name: " +accName);

        System.out.println("Balance Amount: " +balanceAmt);

        System.out.println("Interest Amount: " +interestAmt);

    }

}

class CUB extends RBI {

    static double interest;

    static {

        interest = 5.5;

    }

    void getInput(long accNo, String accName, double balanceAmt) {

        this.accNo = accNo;

        this.accName = accName;

        this.balanceAmt = balanceAmt;

    }

    void calculateInterest() {

        interestAmt = (balanceAmt * interest)/100;

    }

    void printDetails() {

        System.out.println("Bank: CUB");

        System.out.println("Account Number: " +accNo);

        System.out.println("Account Holder Name: " +accName);

        System.out.println("Balance Amount: " +balanceAmt);

        System.out.println("Interest Amount: " +interestAmt);

```



```
    }  
}  
  
public class Demo {  
    public static void main(String[] args) {  
        RBI bank1 = new IndianBank();  
        bank1.getInput(123456789012L, "Gopu Shekhar", 10000);  
        bank1.calculateInterest();  
        bank1.printDetails();  
        System.out.println();  
        RBI bank2 = new ICICI();  
        bank2.getInput(987654321098L, "Tamilkumaran S", 20000);  
        bank2.calculateInterest();  
        bank2.printDetails();  
        System.out.println();  
        RBI bank3 = new CUB();  
        bank3.getInput(564738291046L, "Ranjith Kumar", 15000);  
        bank3.calculateInterest();  
        bank3.printDetails();  
    }  
}
```

OUTPUT:

```
C:\Users\akash\OneDrive\Desktop\Java DA-1>javac Demo.java
```

```
C:\Users\akash\OneDrive\Desktop\Java DA-1>java Demo
```

```
Bank: Indian Bank
```

```
Account Number: 123456789012
```

```
Account Holder Name: Gopu Shekhar
```

```
Balance Amount: 10000.0
```

```
Interest Amount: 500.0
```

```
Bank: ICICI
```

```
Account Number: 987654321098
```

```
Account Holder Name: Tamilkumaran S
```

```
Balance Amount: 20000.0
```

```
Interest Amount: 1200.0
```

```
Bank: CUB
```

```
Account Number: 564738291046
```

```
Account Holder Name: Ranjith Kumar
```

```
Balance Amount: 15000.0
```

```
Interest Amount: 825.0
```