

## SOFTWARE CONSTRUCTION AND DEVELOPMENT

### LAB # 13

#### Good practices of programming

##### LAB TASK:

1. Create a design for the mark sheet by taking runtime value of student name, total marks, obtained marks and calculate its percentage, grade and GPA. Use good practices of programming that we have studied and ensure that the outcomes should be presented in a proper Viewable approach.

##### CODE:

```
package lab10;
import java.util.Scanner;
public class MarkSheet {
    private String studentName;
    private double totalMarks;
    private double obtainedMarks;
    public MarkSheet(String studentName, double totalMarks, double obtainedMarks) {
        this.studentName = studentName;
        this.totalMarks = totalMarks;
        this.obtainedMarks = obtainedMarks;
    }
    public double calculatePercentage() {
        return (obtainedMarks / totalMarks) * 100;
    }
    public String calculateGrade(double percentage) {
        if (percentage >= 90) return "A+";
        else if (percentage >= 80) return "A";
        else if (percentage >= 70) return "B";
        else if (percentage >= 60) return "C";
        else if (percentage >= 50) return "D";
        else return "Fail";
    }
    public double calculateGPA(double percentage) {
        if (percentage >= 90) return 4.0;
        else if (percentage >= 80) return 3.7;
        else if (percentage >= 70) return 3.0;
        else if (percentage >= 60) return 2.5;
        else if (percentage >= 50) return 2.0;
        else return 0.0;
    }
    public void display() {
        double percentage = calculatePercentage();
        System.out.println("\n===== MARK SHEET =====");
        System.out.println("Name : " + studentName);
        System.out.println("Total Marks: " + totalMarks);
        System.out.println("Obtained : " + obtainedMarks);
        System.out.printf("Percentage : %.2f%%\n", percentage);
        System.out.println("Grade : " + calculateGrade(percentage));
        System.out.println("GPA : " + calculateGPA(percentage));
        System.out.println("=====");
    }
}
```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter Student Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Total Marks: ");
    double total = sc.nextDouble();
    System.out.print("Enter Obtained Marks: ");
    double obtained = sc.nextDouble();
    MarkSheet ms = new MarkSheet(name, total, obtained);
    ms.display();
    sc.close();
}
}

```

**OUTPUT:**

```

Enter Student Name: Ayesha
Enter Total Marks: 100
Enter Obtained Marks: 78

```

```

===== MARK SHEET =====
Name      : Ayesha
Total Marks: 100.0
Obtained   : 78.0
Percentage : 78.00%
Grade      : B
GPA        : 3.0
=====

```

2. Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide methods that calculate the rectangle's perimeter and area. It has set and get methods for both length and width. The set methods should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0. Write a program to test class Rectangle.

**CODE:**

```

package lab10;
public class Rectangle {
    private double length = 1.0;
    private double width = 1.0;
    public void setLength(double length) {
        if (length > 0.0 && length < 20.0)
            this.length = length;
        else
            System.out.println("Invalid length!");
    }
    public void setWidth(double width) {
        if (width > 0.0 && width < 20.0)
            this.width = width;
        else
            System.out.println("Invalid width!");
    }
    public double getLength() {
        return length;
    }
    public double getWidth() {
        return width;
    }
}

```

```

    }
    public double area() {
        return length * width;
    }
    public double perimeter() {
        return 2 * (length + width);
    }
    public static void main(String[] args) {
        Rectangle r = new Rectangle();
        r.setLength(10.5);
        r.setWidth(6.5);
        System.out.println("\n===== RECTANGLE =====");
        System.out.println("Length : " + r.getLength());
        System.out.println("Width : " + r.getWidth());
        System.out.println("Area : " + r.area());
        System.out.println("Perimeter : " + r.perimeter());
        System.out.println("===== =====");
    }
}

```

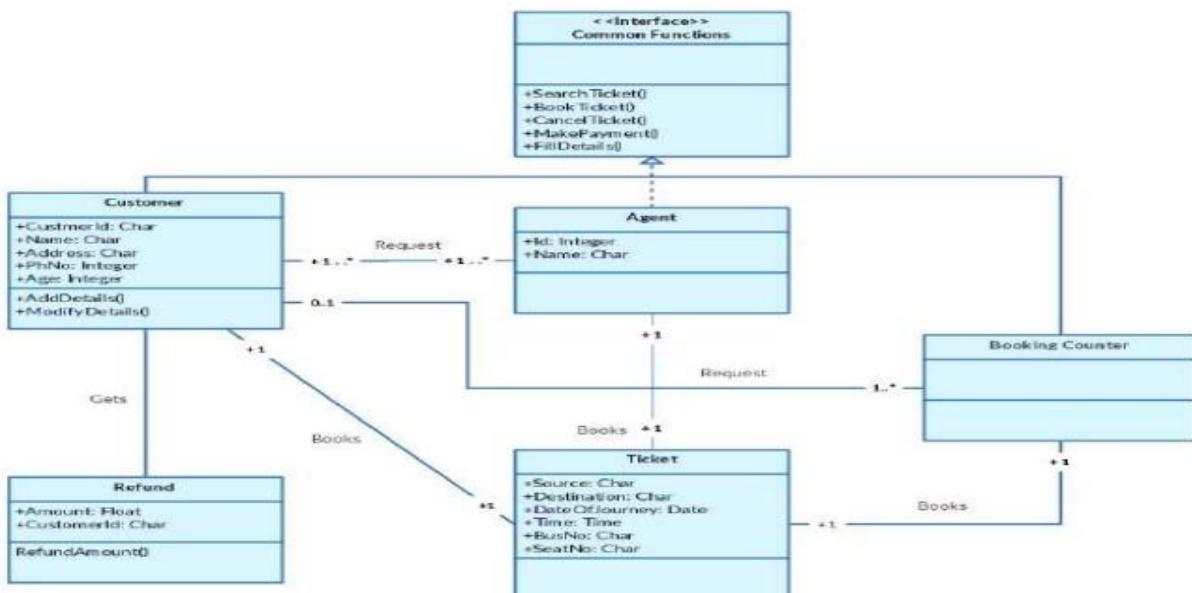
**OUTPUT:**

```

===== RECTANGLE =====
Length : 10.5
Width : 6.5
Area : 68.25
Perimeter : 34.0
=====

```

3. Convert the following class diagram into optimized and meaningful java code.

**CODE:**

```
package lab10;
import java.util.Date;
interface CommonFunctions {
    void searchTicket();
    void bookTicket();
    void cancelTicket();
    void makePayment();
    void fillDetails();
}
class Customer implements CommonFunctions {
    private String customerId;
    private String name;
    private String address;
    private String phone;
    private int age;
    public Customer(String customerId, String name) {
        this.customerId = customerId;
        this.name = name;
    }
    public void addDetails(String address, String phone, int age) {
        this.address = address;
        this.phone = phone;
        this.age = age;
    }
    public void modifyDetails(String address, String phone) {
        this.address = address;
        this.phone = phone;
    }
    @Override
    public void searchTicket() {
        System.out.println("Customer searching ticket...");
    }
    @Override
    public void bookTicket() {
        System.out.println("Customer booking ticket...");
    }
    @Override
    public void cancelTicket() {
        System.out.println("Customer canceling ticket...");
    }
    @Override
    public void makePayment() {
        System.out.println("Customer making payment...");
    }
    @Override
    public void fillDetails() {
        System.out.println("Customer filling details...");
    }
    public String getCustomerId() {
        return customerId;
    }
}
```

```
        }
    }
}

class Agent implements CommonFunctions {
    private int id;
    private String name;
    public Agent(int id, String name) {
        this.id = id;
        this.name = name;
    }
    @Override
    public void searchTicket() {
        System.out.println("Agent searching ticket...");
    }
    @Override
    public void bookTicket() {
        System.out.println("Agent booking ticket...");
    }
    @Override
    public void cancelTicket() {
        System.out.println("Agent canceling ticket...");
    }
    @Override
    public void makePayment() {
        System.out.println("Agent processing payment...");
    }
    @Override
    public void fillDetails() {
        System.out.println("Agent filling customer details...");
    }
}
class BookingCounter {

    public void processRequest() {
        System.out.println("Booking counter processing request...");
    }

    public void bookTicket(Ticket ticket) {
        System.out.println("Ticket booked for seat: " + ticket.getSeatNo());
    }
}
class Ticket {

    private String source;
    private String destination;
    private Date dateOfJourney;
    private String busNo;
    private String seatNo;

    public Ticket(String source, String destination, String seatNo) {
        this.source = source;
        this.destination = destination;
        this.seatNo = seatNo;
        this.dateOfJourney = new Date();
    }
}
```

```
        public String getSeatNo() {
            return seatNo;
        }
    }
    class Refund {
        private float amount;
        private String customerId;
        public Refund(float amount, String customerId) {
            this.amount = amount;
            this.customerId = customerId;
        }
        public void refundAmount() {
            System.out.println("Refund of Rs " + amount + " issued to Customer ID: " +
customerId);
        }
    }
    public class BookingSystem {
        public static void main(String[] args) {
            Customer customer = new Customer("C101", "Ali");
            customer.addDetails("Karachi", "0300-1234567", 22);
            Agent agent = new Agent(1, "Ahmed");
            Ticket ticket = new Ticket("Karachi", "Lahore", "A1");
            BookingCounter counter = new BookingCounter();
            counter.processRequest();
            counter.bookTicket(ticket);
            Refund refund = new Refund(1500.0f, customer.getCustomerId());
            refund.refundAmount();
        }
    }
}
```

**OUTPUT:**

```
Booking counter processing request...
Ticket booked for seat: A1
Refund of Rs 1500.0 issued to Customer ID: C101
```