**AMITY UNIVERSITY, NOIDA**

**UTTAR PRADESH**



**AMITY INSTITUTE OF   
INFORMATION TECHNOLOGY**

**Core Java**

**Lab File**

**CSIT – 751**

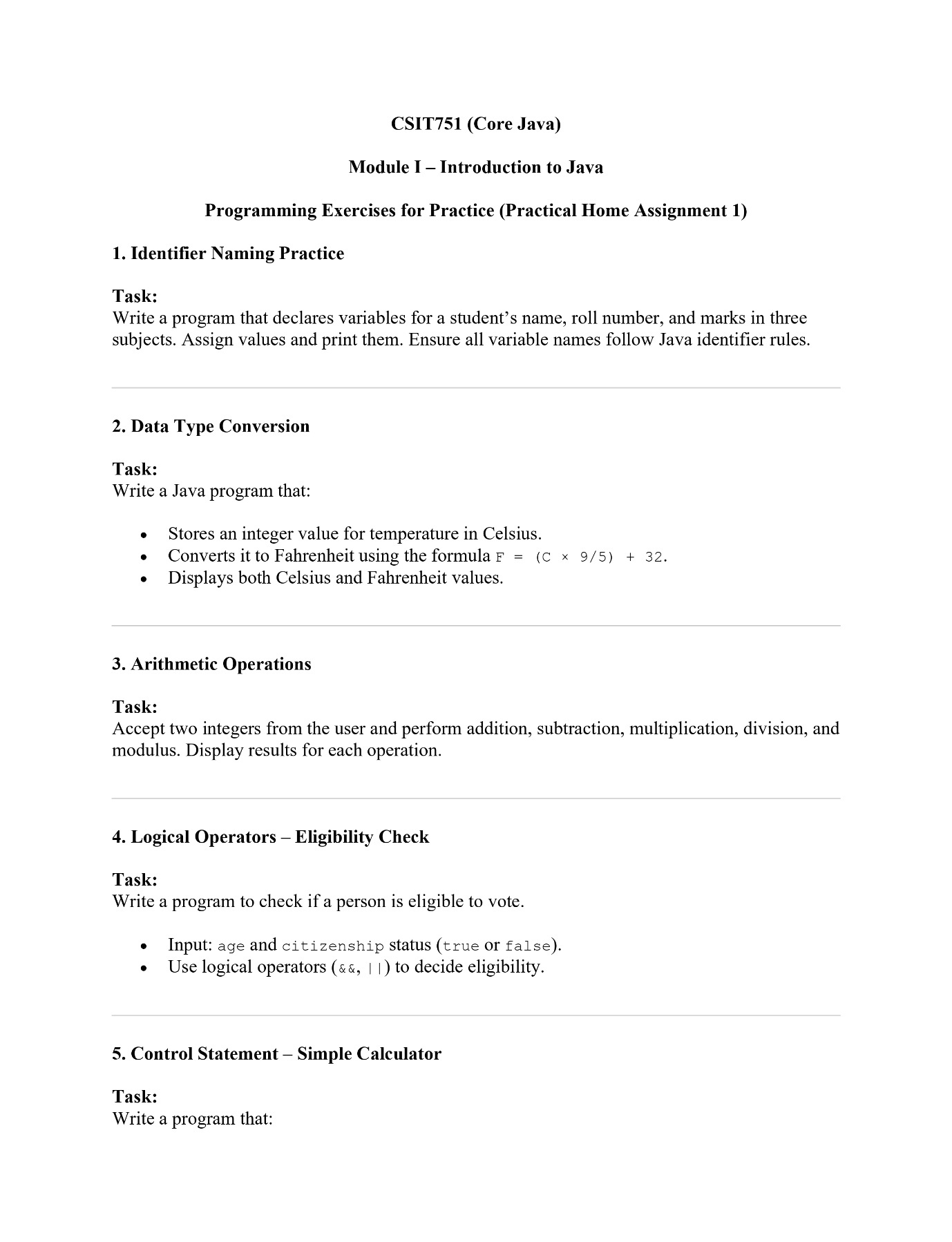
**Submitted To:**Dr. Ajay Rastogi

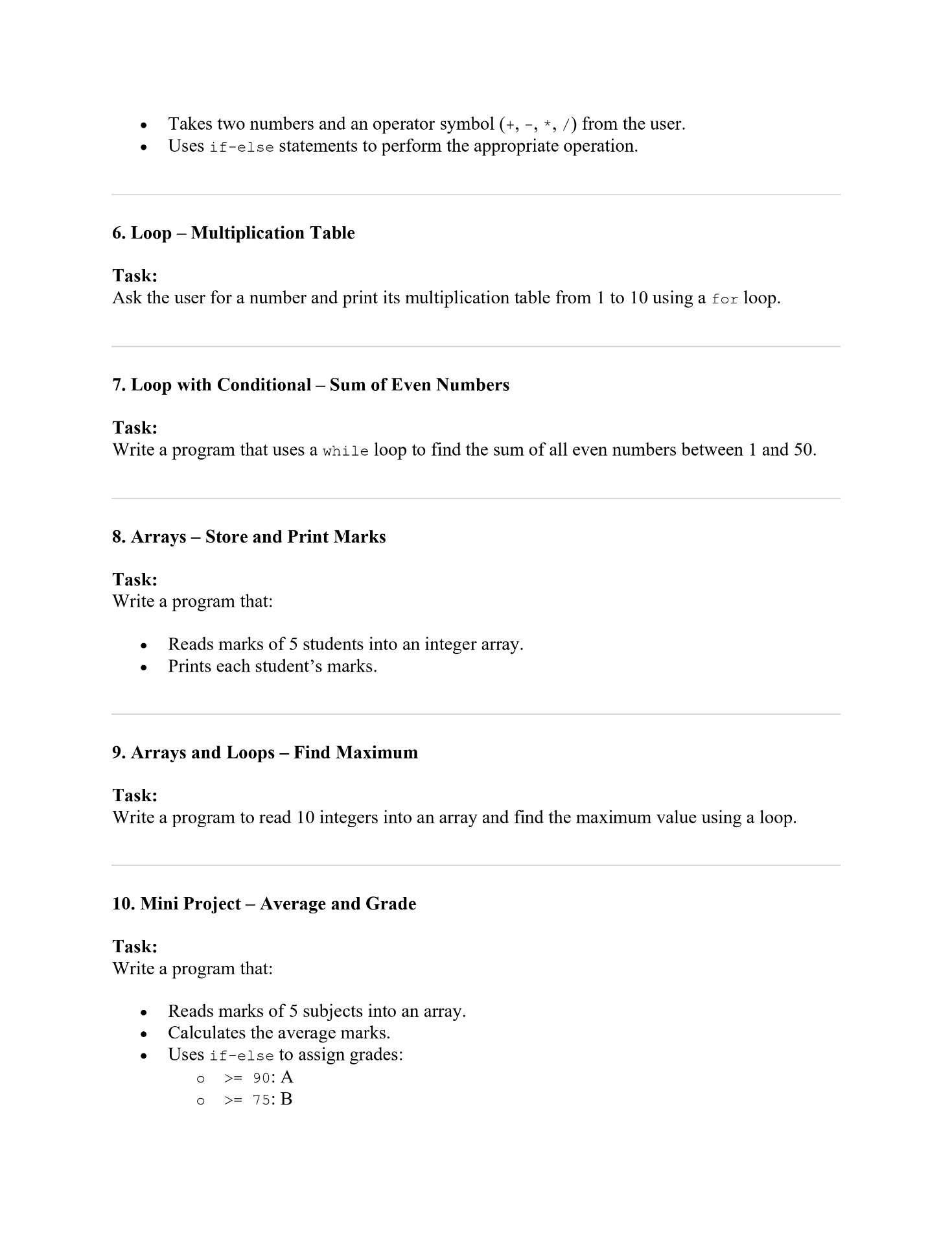
**Submitted By:**   
Tejas Upadhyay  
A010175625005  
MSc. AI and DA.

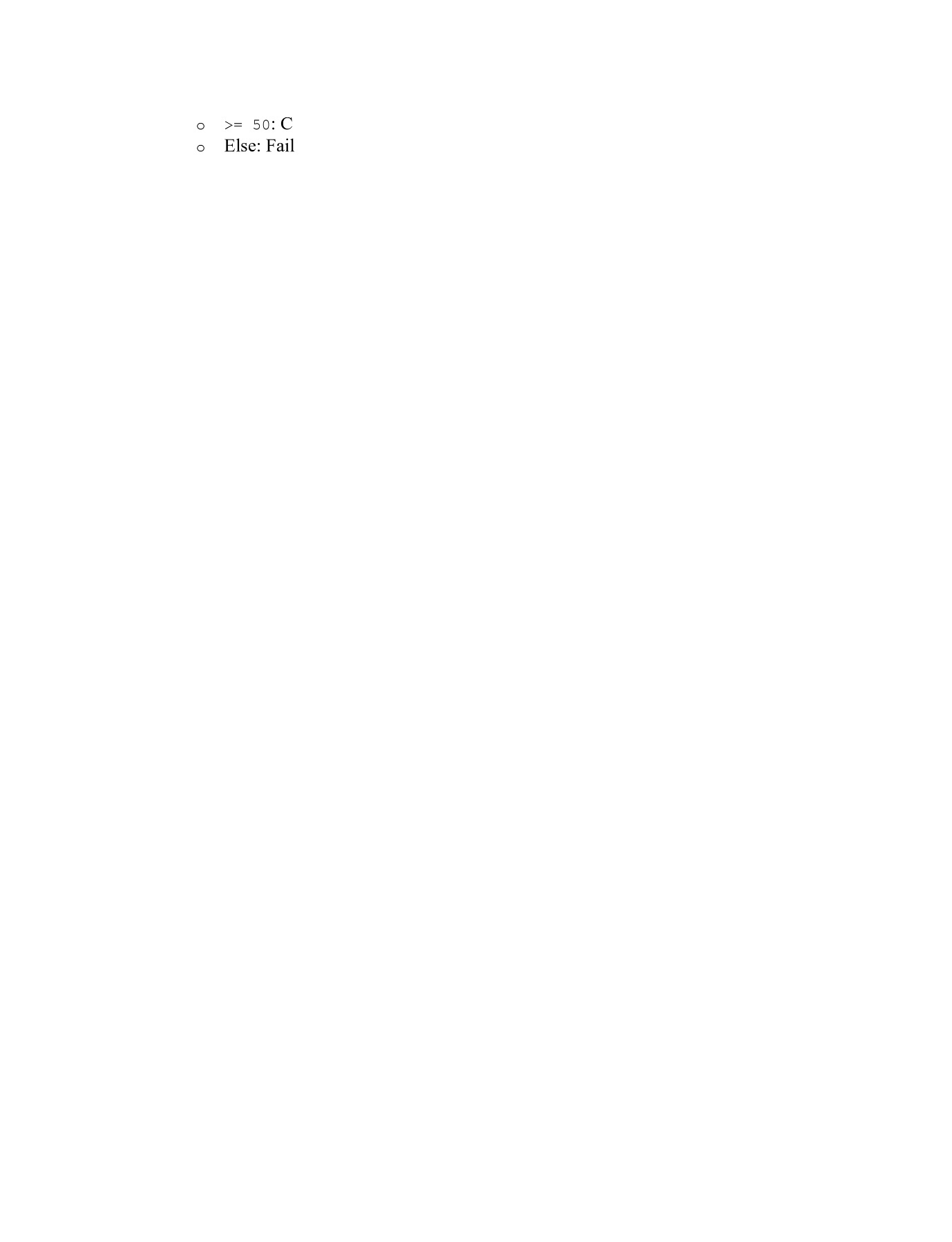
**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **List of Practicals (CSIT751: Core Java)** | | | | |
|
| **MODULE I** | | 1 | Identifier Naming Practice |  |
| 2 | Data Type Conversion |  |
| 3 | Arithmetic Operations |  |
| 4 | Logical Operators – Eligibility Check |  |
| 5 | Control Statement – Simple Calculator |  |
| 6 | Loop – Multiplication Table |  |
| 7 | Loop with Conditional – Sum of Even Numbers |  |
| 8 | Arrays – Store and Print Marks |  |
| 9 | Arrays and Loops – Find Maximum |  |
| 10 | Mini Project – Average and Grade |  |
|  |  |  |  |  |
| **MODULE II** | | 1 | University – Department – Professor (One-to-Many Relationship) |  |
| 2 | Flight Booking System (Abstraction + Inheritance) |  |
| 3 | Employee Payroll System (Constructor Chaining + Overriding) |  |
| 4 | Online Shopping Cart (Aggregation) |  |
| 5 | Hospital Management (Multi-level Inheritance) |  |
| 6 | Library – Book – Author (One-to-One Relationship) |  |
| 7 | Sports League (Polymorphism – Method Overriding) |  |
| 8 | Loan Management System (Abstraction + Polymorphism) |  |
| 9 | Online Course Platform (Association + Inheritance) |  |
| 10 | Smart Home Devices (Interfaces + Polymorphism) |  |
| 11 | University Hostel Allocation (One-to-One Relationship) |  |
| 12 | Vehicle Rental System (Inheritance + Overriding) |  |
| 13 | Hotel Reservation System (Aggregation) |  |
| 14 | Banking ATM Simulation (Encapsulation) |  |
| 15 | Passport – Citizen (One-to-One Relationship) |  |
|  |  |  |  |  |
| **MODULE III** | | 1 | ATM Withdrawal System – Handling Insufficient Balance |  |
| 2 | Student Age Validation – User Defined Exception |  |
| 3 | Array Division – Handling Arithmetic and Array Index Exception (Multiple Catch) |  |
| 4 | Student Marks Entry – Input Validation |  |
| 5 | University Login System – Null Pointer Handling |  |
| 6 | Online Shopping – Minimum Purchase Amount Exception |  |
| 7 | Railway Ticket Booking – Synchronization of Threads |  |
| 8 | University Printing System – Multiple Threads |  |
| 9 | IoT Sensor Monitoring System |  |
| 10 | Bank Transaction System – Thread Priorities |  |
|  |  |  |  |  |
| **MODULE IV** | | 1 | Library Book Management – Using ArrayList |  |
| 2 | Student Attendance System – Using HashSet |  |
| 3 | Bank Account Directory – Using HashMap |  |
| 4 | Product Inventory Management – Using HashMap and Iterator |  |
| 5 | Online Course Enrollment – Using LinkedHashSet |  |
| 6 | Student Registration Form — GUI with AWT Components |  |
| 7 | Calculator Form — Event-Driven Programming |  |
|  |  |  |  |  |
| **MODULE V** | | 1 | Java Web Application(Feedback Form) |  |
|  |  |  |  |  |
| **MODULEVI** | | 1 | Java Desktop Application with Database(Employee Management Application) |  |

**MODULE 1**

****

****

****

**Q1 Identifier Naming Practice**

**:**

package Module1;

import java.util.Scanner;

public class PHA\_M1\_Q1 {

public static void main(String[] args){

String name;

int rollNo;

double marks;

Scanner scanner = new Scanner(System.in);

System.out.print("Please enter your name: ");

name = scanner.nextLine();

System.out.print("Please enter your roll number: ");

rollNo = scanner.nextInt();

System.out.print("Please enter your marks : ");

marks = scanner.nextFloat();

System.out.println("Name : " + name);

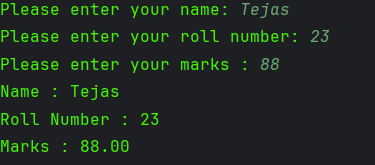
System.out.println("Roll Number : " + rollNo);

System.out.printf("Marks : %.2f",marks);

scanner.close();

}}

**Output:**

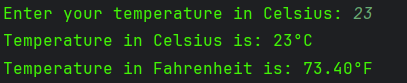
****

**Q2 Data Type Conversion**

**Code:**

import java.util.Scanner;  
public class PHA\_M1\_Q2 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter your temperature in Celsius: ");  
 int tempCelsius = scanner.nextInt();  
 double tempFahrenheit = (tempCelsius \* 1.8) + 32;  
 System.*out*.printf("Temperature in Celsius is: %02d°C \n", tempCelsius);  
 System.*out*.printf("Temperature in Fahrenheit is: %.2f°F ", tempFahrenheit);  
  
 scanner.close();  
 }  
}

**Output:**

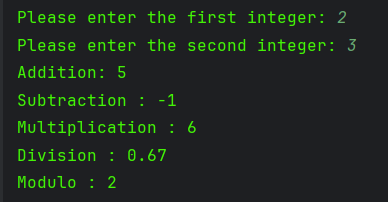
****

**Q3. Arithmetic Operations**.

**Code:**

import java.util.Scanner;  
public class PHA\_M1\_Q3 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Please enter the first integer: ");  
 int num1 = scanner.nextInt();  
 System.*out*.print("Please enter the second integer: ");  
 int num2 = scanner.nextInt();  
 if (num2 == 0){  
 System.*out*.println("The second number cannot be 0 as division isn't possible.");  
 }  
 else {  
 System.*out*.println("Addition: " + (num1 + num2));  
 System.*out*.println("Subtraction : " + (num1 - num2));  
 System.*out*.println("Multiplication : " + (num1 \* num2));  
 System.*out*.printf("Division : %.2f \n", (num1\*1.0/num2));  
 System.*out*.println("Modulo : " + (num1%num2));}  
 scanner.close();  
 }  
}

**Output:**

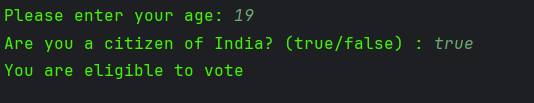
****

**4. Logical Operators – Eligibility Check**

**Code:**

import java.util.Scanner;  
public class PHA\_M1\_Q4 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Please enter your age: ");  
 int age = scanner.nextInt();  
 System.*out*.print("Are you a citizen of India? (true/false) : ");  
 boolean citizenship = scanner.nextBoolean();  
 if (age >= 18 && citizenship){  
 System.*out*.println("You are eligible to vote");  
 }  
 else{  
 System.*out*.println("You are not eligible to vote");  
 }  
 scanner.close();  
 }  
}

**Output:**

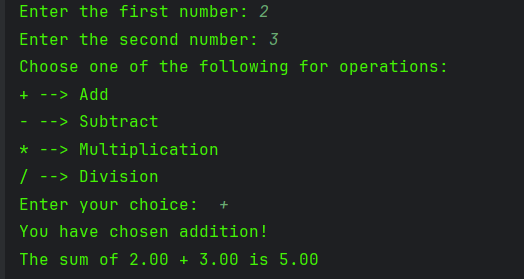


**5. Control Statement – Simple Calculator**

**Code:**

import java.util.Scanner;  
  
public class PHA\_M1\_Q5 {  
public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter the first number: ");  
 double num1 = scanner.nextDouble();  
 System.*out*.print("Enter the second number: ");  
 double num2 = scanner.nextDouble();  
 System.*out*.println("Choose one of the following for operations: \n+ --> Add\n- --> Subtract\n\* --> Multiplication\n/ --> Division");  
 System.*out*.print("Enter your choice: ");  
 scanner.nextLine();  
 char choice = scanner.next().charAt(0);  
 if (choice == '+'){  
 System.*out*.println("You have chosen addition!");  
 System.*out*.printf("The sum of %.2f + %.2f is %.2f", num1, num2, num1 + num2 );  
 } else if (choice == '-') {  
 System.*out*.println("You have chosen Subtraction!");  
 System.*out*.printf("The difference of %.2f - %.2f is %.2f", num1, num2, num1 - num2 );  
 } else if (choice == '\*') {  
 System.*out*.println("You have chosen Multiplication!");  
 System.*out*.printf("The multiplication of %.2f x %.2f is %.2f", num1, num2, num1\*num2 );  
 } else if (choice == '/') {  
 if (num2 == 0){  
 System.*out*.println("Number 2 is 0, thus cannot divide!!");  
 }  
 else {  
 System.*out*.println("You have chosen Division!");  
 System.*out*.printf("The division of %.2f / %.2f is %.2f", num1, num2, num1/num2 );  
 }  
   
 }  
 else {  
 System.*out*.println("Please choose a valid option");  
 }  
 scanner.close();  
}  
}

**Output:**

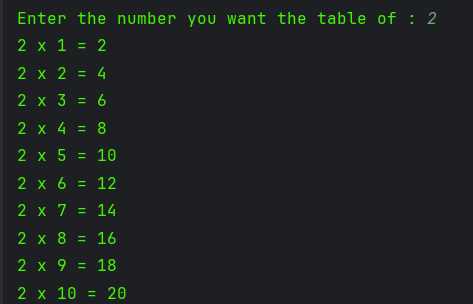
****

**6. Loop – Multiplication Table**

**Code:**

package Module1;  
import java.util.Scanner;  
public class PHA\_M1\_Q6 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter the number you want the table of : ");  
 int num = scanner.nextInt();  
 for (int i = 1; i <= 10 ;i++ ){  
 System.*out*.printf("%d x %d = %d\n",num,i,num\*i);  
 }  
 scanner.close();  
 }  
}

**Output:**



**7. Loop with Conditional – Sum of Even Numbers**

**Code:**

public class PHA\_M1\_Q7 {  
 public static void main(String[] args){  
 int num = 1;  
 int sumEven = 0;  
 while (num <= 50) {  
 if (num % 2 == 0){  
 sumEven += num;  
 }  
 num++;  
 }  
 System.*out*.println("The sum of even numbers from 1 to 50 is : "+sumEven);  
 }  
}

**Output:**

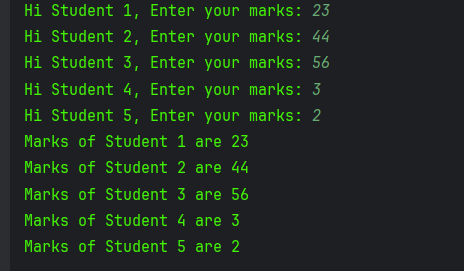
****

**8. Arrays – Store and Print Marks**

**Code:**

package Module1;  
import java.util.Scanner;  
public class PHA\_M1\_Q8 {  
 public static void main(String[] args){  
 int[] marksArray = new int[5];  
 Scanner scanner = new Scanner(System.*in*);  
 for (int i = 0; i < 5; i++){  
 System.*out*.printf("Hi Student %d, Enter your marks: ", i+1);  
 marksArray[i] = scanner.nextInt();  
 }  
 for (int i = 0; i < 5; i++){  
 System.*out*.printf("Marks of Student %d are %d\n",i+1,marksArray[i]);  
 }  
 scanner.close();  
 }  
}

**Output:**

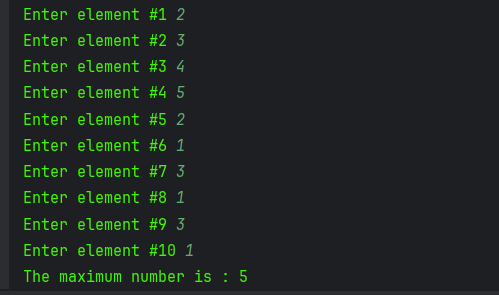
****

**9. Arrays and Loops – Find Maximum**

**Code:**

package Module1;  
import java.util.Scanner;  
public class PHA\_M1\_Q9 {  
  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 int[] arr = new int[10];  
 int max = 0 ;  
 for (int i = 0; i < 10 ; i++){  
 System.*out*.printf("Enter element #%d",i+1);  
 arr[i] = scanner.nextInt();  
 }  
 max = arr[0];  
 for (int num: arr){  
 System.*out*.println(num);  
 if (num > max){  
 max = num;}  
 }  
 System.*out*.println("The maximum number is : " + max);  
 scanner.close();  
 }  
  
}

**Output**

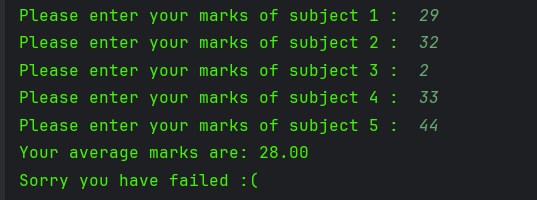


**10. Mini Project – Average and Grade**

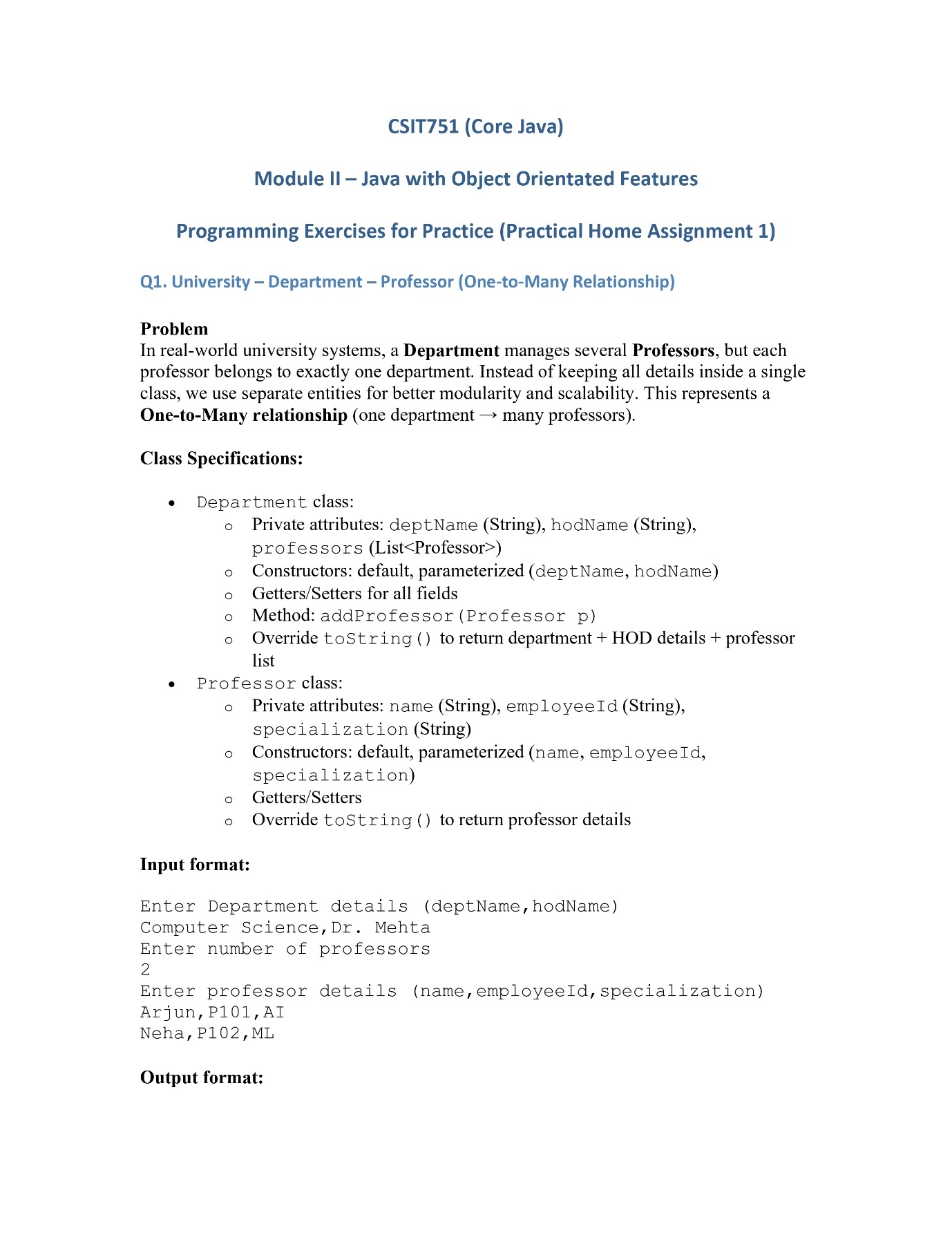
**Code:**

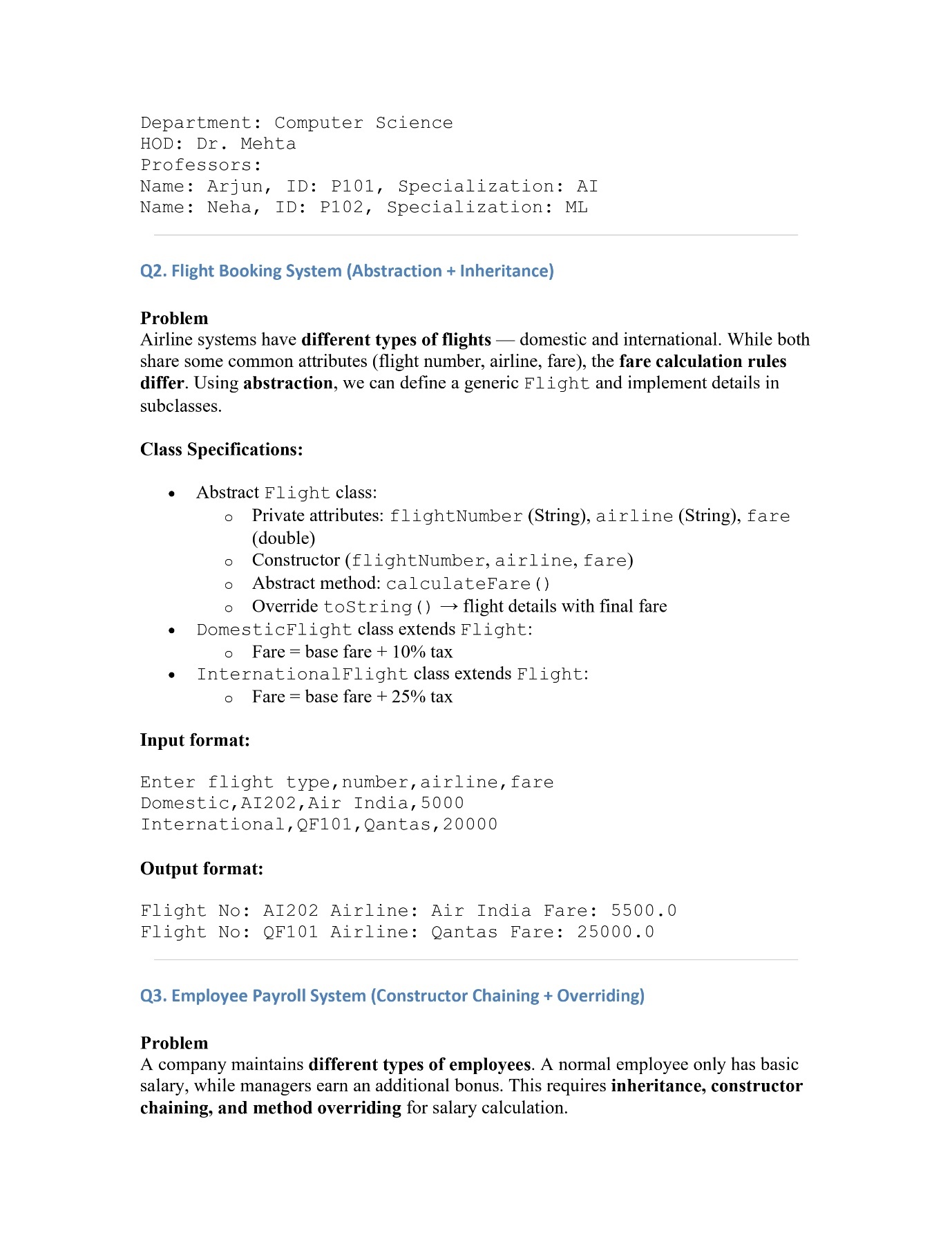
package Module1;  
import java.util.Scanner;  
public class PHA\_M1\_Q10 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 double[] marksArray = new double[5] ;  
 double avgMarks = 0;  
 char grade;  
 for (int i = 0; i < marksArray.length ; i++){  
 System.*out*.printf("Please enter your marks of subject %d : ", i+1);  
 marksArray[i] = scanner.nextDouble();  
 }  
 for (double marks :marksArray){  
 avgMarks += marks;  
 }  
 avgMarks /= marksArray.length;  
 System.*out*.printf("Your average marks are: %.2f\n",avgMarks);  
  
 if (avgMarks >= 90){  
 grade = 'A';  
 } else if (avgMarks >= 75) {  
 grade = 'B';  
 } else if (avgMarks >= 50) {  
 grade = 'C';  
 }  
 else {  
 grade = 'F';  
 }  
  
 if (grade != 'F'){  
 System.*out*.println("Congrats! You have passed, here is your grade: " + grade);  
 }  
 else {  
 System.*out*.println("Sorry you have failed :( ");  
 }  
 scanner.close();  
  
  
 }  
}

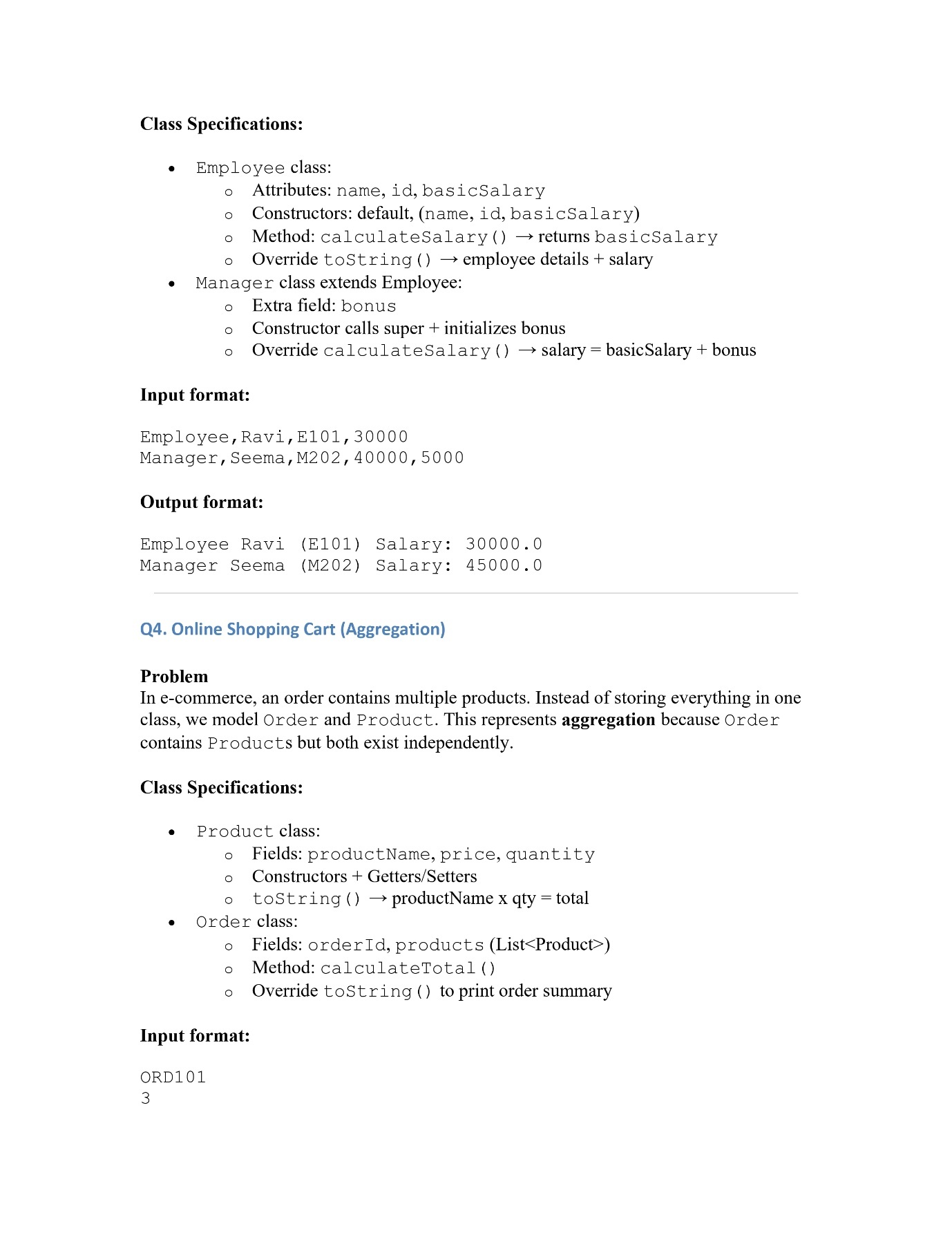
**Output:**

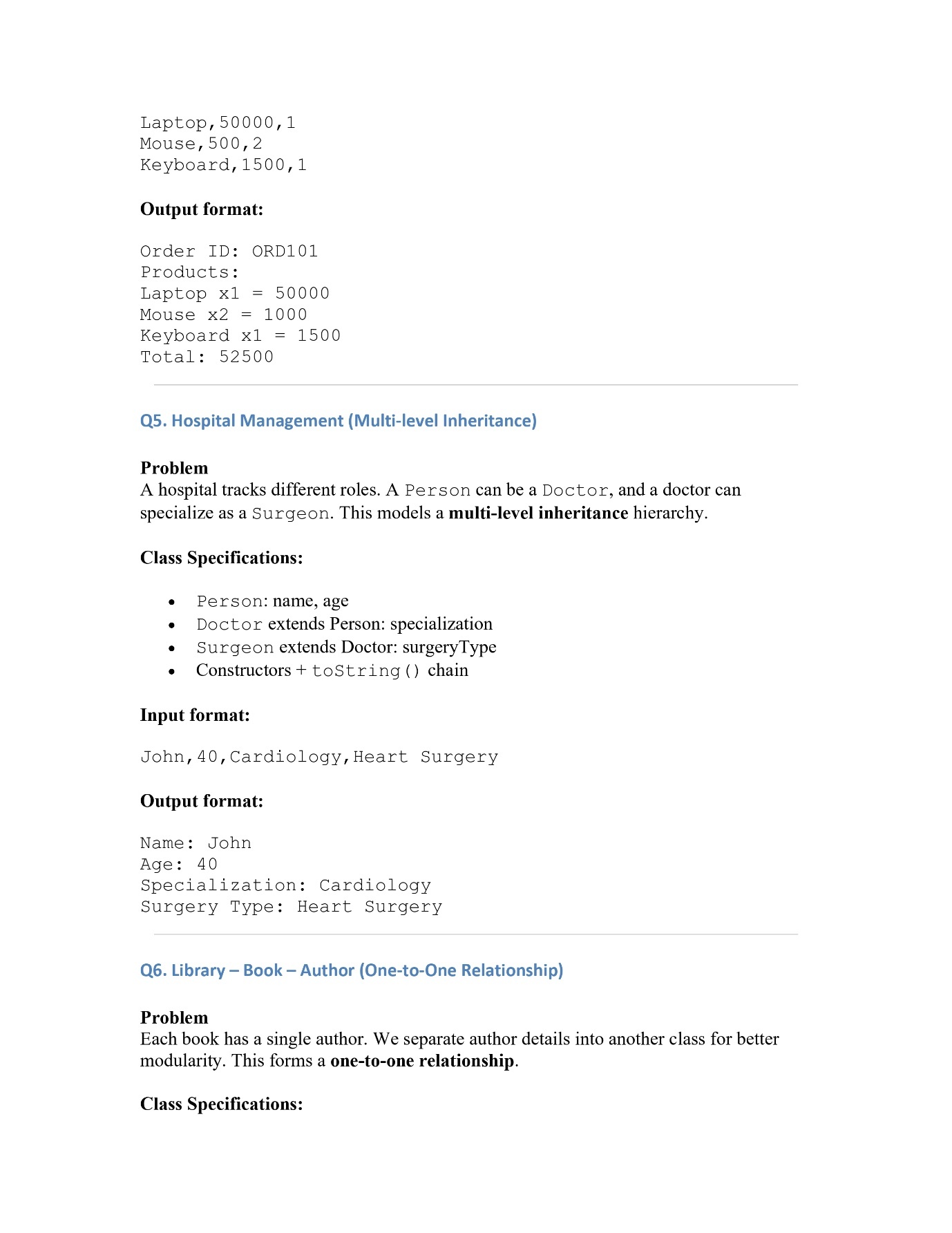
****

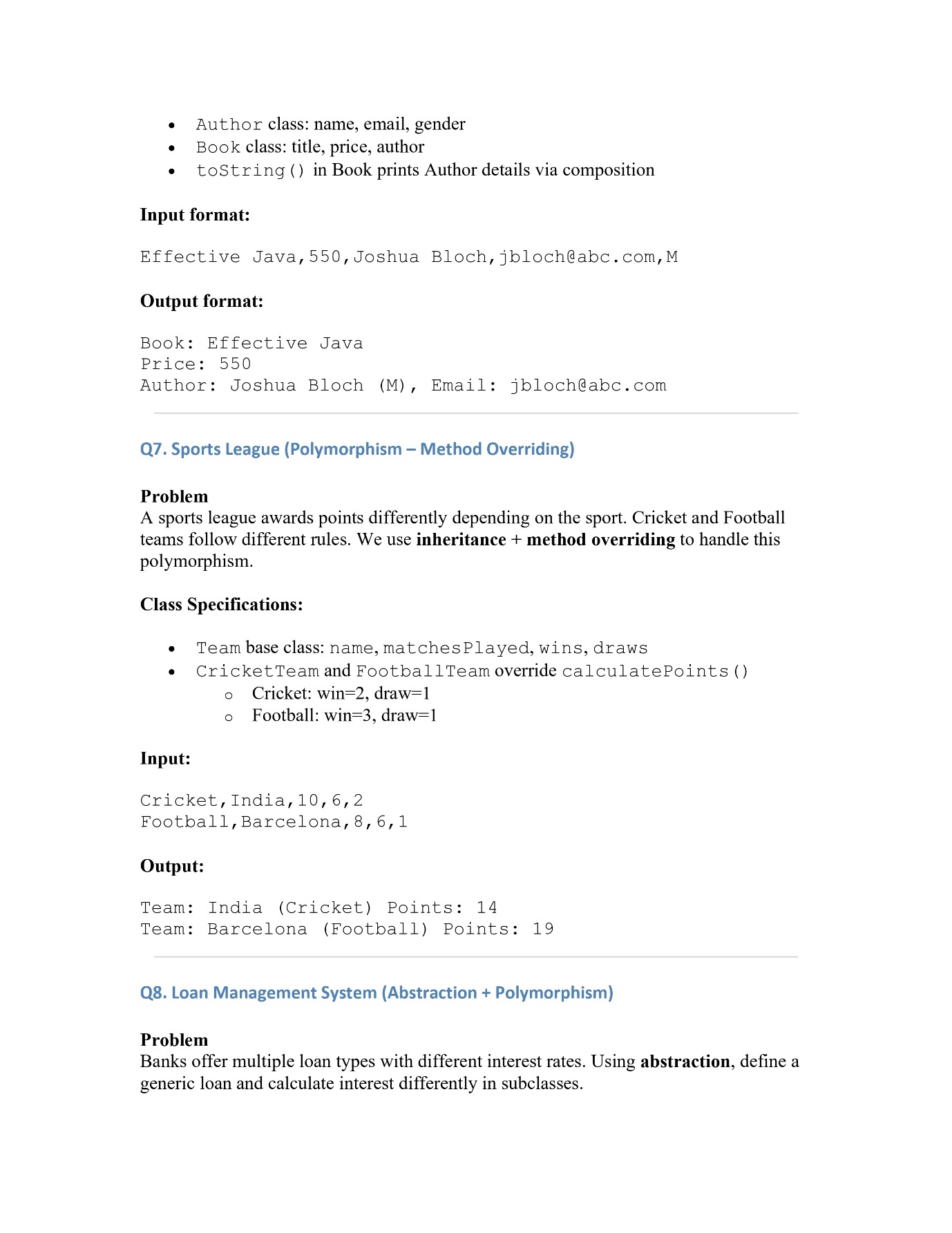
**Module 2**

****

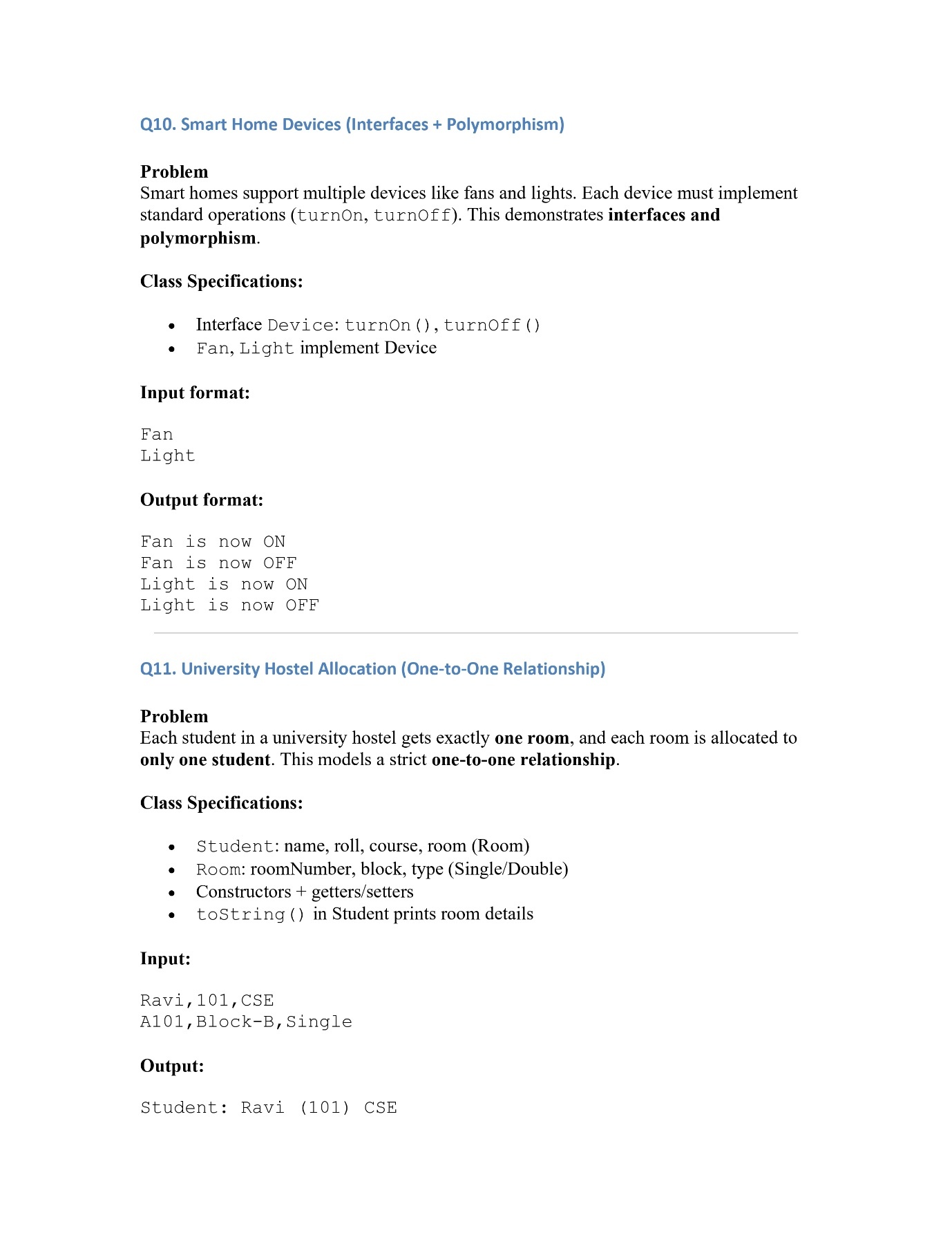
****

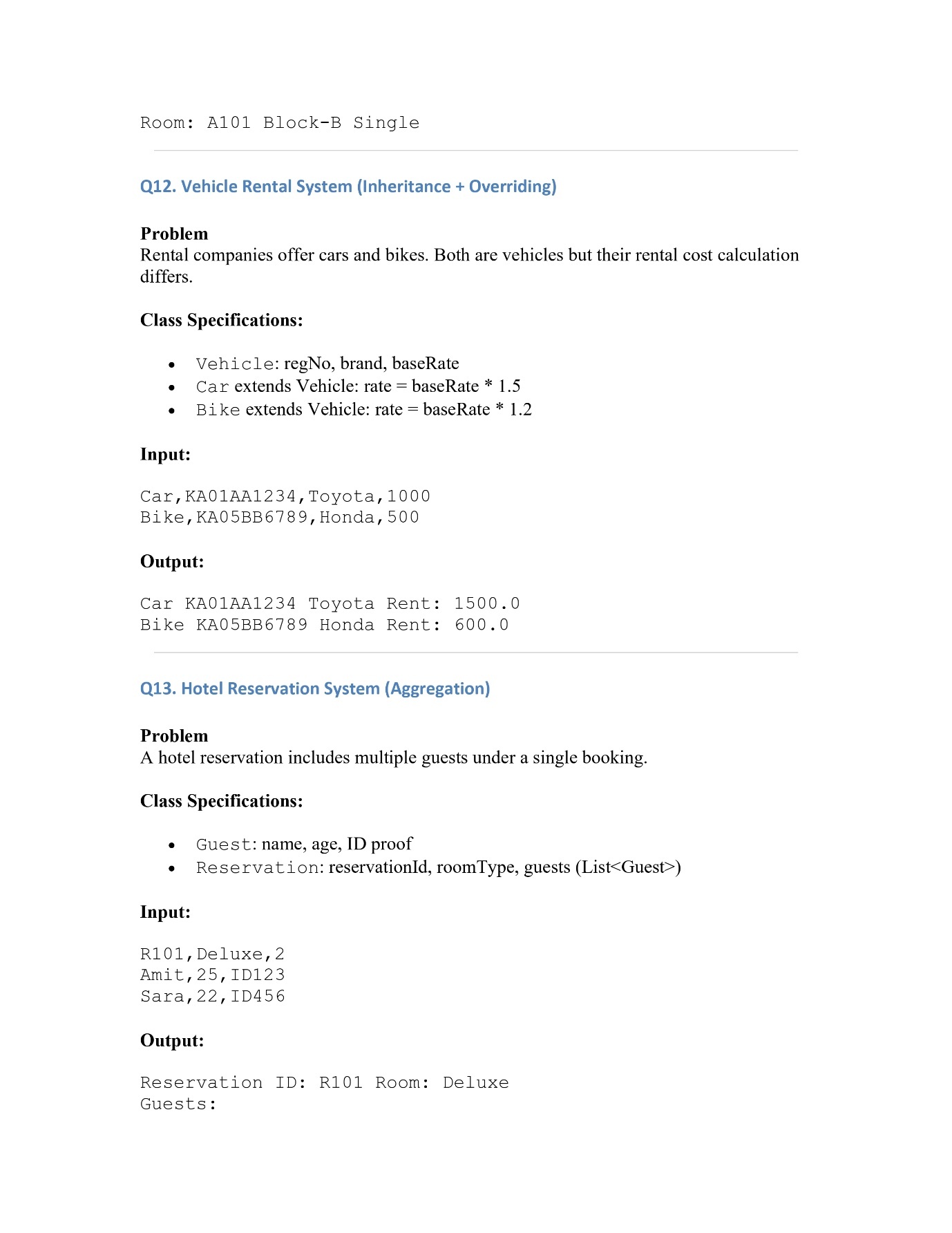
****

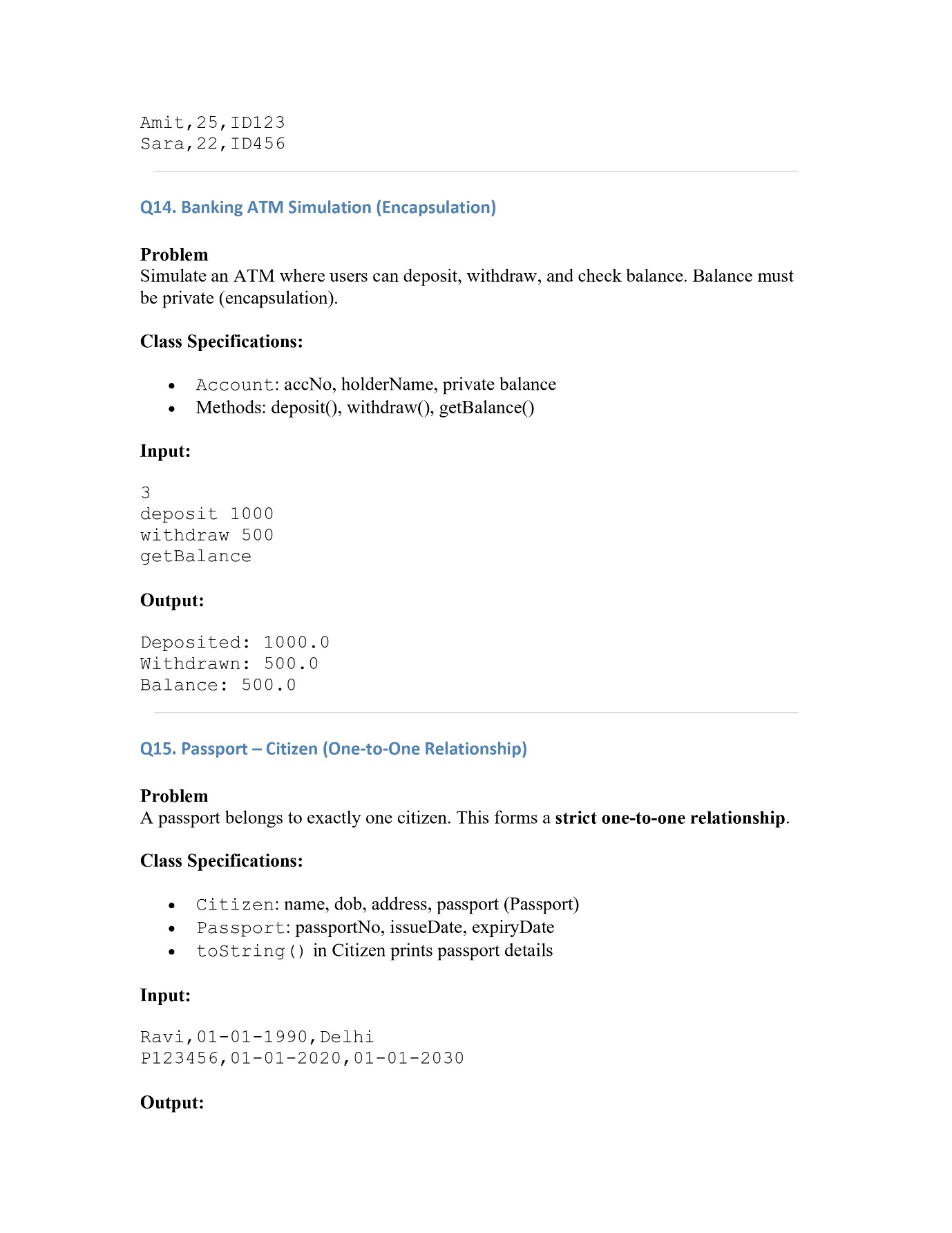
****

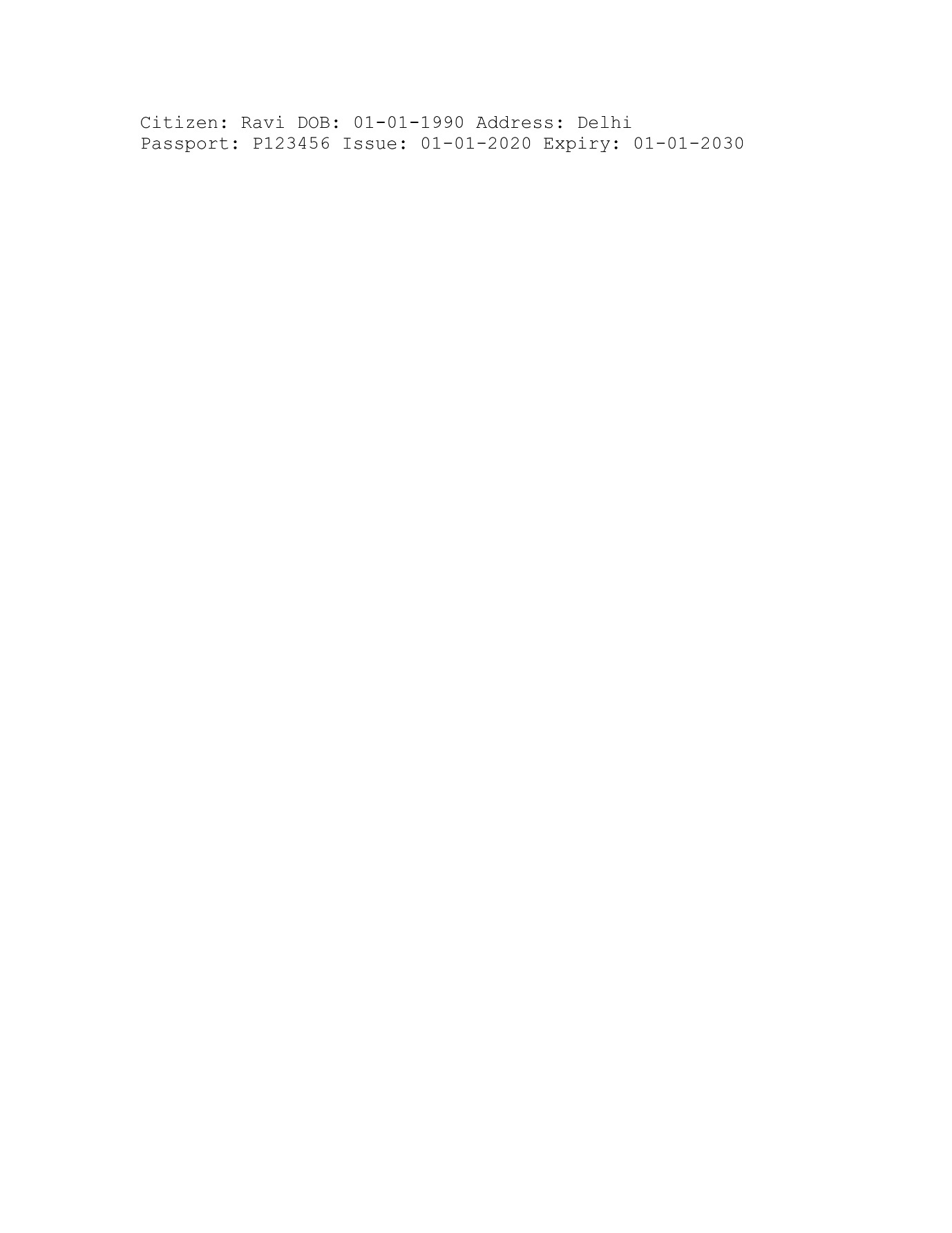
****

****

****

****

****

****

**Q1. University – Department – Professor (One-to-Many Relationship)**

**Code:**

**Professor Class:**

package Module2;  
public class Professor {  
 private String name;  
 private String employeeId;  
 private String specialization;  
  
 Professor(String name,String employeeId,String specialization){  
 this.name = name;  
 this.employeeId = employeeId;  
 this.specialization = specialization;  
 }  
  
 String getDetails(){  
 return "Name : " +this.name +", EmployeeID : " + this.employeeId + ", Specialization : " +  
 this.specialization;  
 }  
 @Override  
 public String toString(){  
 return this.getDetails();  
 }  
  
}

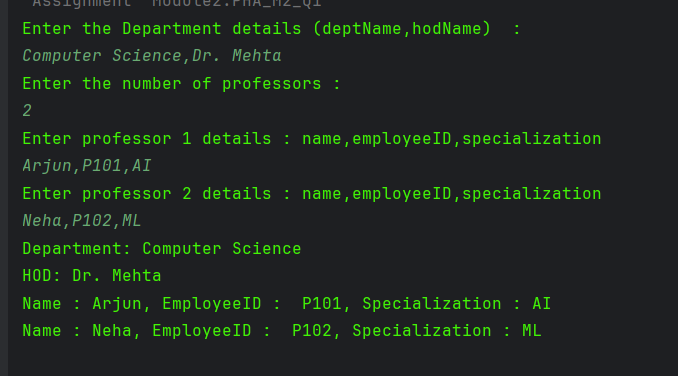
**Department Class:**

package Module2;  
  
public class Department {  
 private String deptName;  
 private String hodName;  
 private Professor[] professors = new Professor[10];  
 private int count = 0;  
  
 Department(String deptName, String hodName){  
 this.deptName = deptName;  
 this.hodName = hodName;  
 }  
  
 void addProfessor(Professor p){  
 this.professors[count] = p;  
 this.count += 1;  
 }  
  
 @Override  
 public String toString(){  
 String dept = "Department: " + this.deptName + "\n" ;  
 String hod = "HOD: " + this.hodName + "\n";  
 String professorDetails = "";  
  
 for (int i = 0; i < this.count; i++){  
 professorDetails += professors[i].getDetails() + "\n";  
 }  
 return dept + hod + professorDetails;  
 }  
}

**Main Class:**

package Module2;  
import java.util.Scanner;  
  
public class PHA\_M2\_Q1 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter the Department details (deptName,hodName) : ");  
 String[] dept\_Details = scanner.nextLine().split(",");  
  
 Department dept = new Department(dept\_Details[0],dept\_Details[1]); //department init  
 System.*out*.println("Enter the number of professors : ");  
 int num\_profs = scanner.nextInt();  
 scanner.nextLine();  
 for (int i = 0; i < num\_profs; i++){  
 System.*out*.println("Enter professor " + (i+1) + " details : name,employeeID,specialization");  
 String[] tempProfArr = scanner.nextLine().split(",");  
 dept.addProfessor(new Professor(tempProfArr[0],tempProfArr[1],tempProfArr[2]));}  
 System.*out*.println(dept);  
 }  
  
}

**Output:**

****

**Q2. Flight Booking System (Abstraction + Inheritance)**

**Code:**

**Flight Class**

package Module2;  
  
abstract class Flight {  
 private String flightNumber;  
 private String airline;  
 private double fare;  
  
 Flight(String flightNumber, String airline, double fare){  
 this.flightNumber = flightNumber;  
 this.airline = airline;  
 this.fare = fare;  
 }  
  
 String getFlightNumber(){  
 return this.flightNumber;}  
  
 String getAirline(){  
 return this.airline;}  
 Double getFare(){  
 return this.fare;}  
  
 @Override  
 public String toString(){  
 return "Flight No. : "+ this.flightNumber + " Airline : " + this.airline + " Fare : " +  
 this.calculateFare();  
 }  
 abstract double calculateFare();  
}

**Domestic Flight**

package Module2;  
  
public class DomesticFlight extends Flight{  
 DomesticFlight(String flightNumber, String flight, double fare){  
 super(flightNumber,flight,fare);  
 }  
  
  
  
 @Override  
 double calculateFare(){  
 double fare = getFare();  
 return fare + 0.10\*fare;}  
}

**International Flight**

package Module2;  
  
public class InternationalFlight extends Flight{  
 InternationalFlight(String flightNumber, String flight, double fare){  
 super(flightNumber,flight,fare);  
 }  
  
  
  
 @Override  
 double calculateFare(){  
 double fare = getFare();  
 return fare + 0.25\*fare;}  
}

**Main Class**

// if multiple is asked here, we can always just put this in a loop  
package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q2 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 String choice;  
 do {  
 Flight flight;  
 System.*out*.println("Enter flight type,number,airline,fare");  
 String[] details = scanner.nextLine().split(",");  
  
 if (details[0].equalsIgnoreCase("Domestic")){  
 flight = new DomesticFlight(details[1],details[2],Double.*parseDouble*(details[3]));  
 System.*out*.println(flight);  
 } else if (details[0].equalsIgnoreCase("International")) {  
 flight = new InternationalFlight(details[1],details[2],Double.*parseDouble*(details[3]));  
 System.*out*.println(flight);  
 }  
 else {  
 System.*out*.println("Wrong flight type entered, code will not work");  
  
 }  
 System.*out*.print("Do you want to continue : Yes/No ");  
 choice = scanner.nextLine();  
 }while (choice.toUpperCase().charAt(0) != 'N');  
 System.*out*.println("Thanks for using our program!");  
  
 }  
}

**Output**

****

**Q3. Employee Payroll System (Constructor Chaining + Overriding)**

**Code:**

**Employee Class:**

package Module2;  
  
public class Employee {  
 String name;  
 String id;  
 double basicSalary;  
 double bonus;  
 Employee(String name,String id, double basicSalary){  
 this.name = name;  
 this.basicSalary = basicSalary;  
 this.id = id;  
 }  
 Employee(String name,String id, double basicSalary,double bonus){  
 this.name = name;  
 this.basicSalary = basicSalary;  
 this.id = id;  
 this.bonus = bonus;  
 }  
  
 double calculateSalary(){  
 return this.basicSalary;  
 }  
  
 public String toString(){  
 if (this.id.charAt(0) == 'M'){  
 return "Manager " + this.name + "(" + this.id + ")" + " Salary: " + calculateSalary();  
 }  
 else {  
 return "Employee " + this.name + "(" + this.id + ")" + " Salary: " + calculateSalary();  
 }  
 }  
}

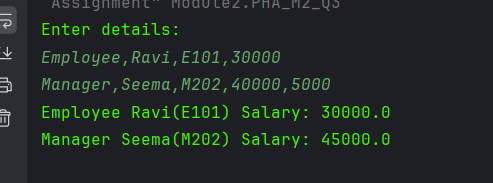
**Manager Class:**

package Module2;  
  
public class Manager extends Employee {  
 Manager(String name, String id,double salary,double bonus){  
 super(name,id,salary,bonus);  
 }  
 @Override  
 double calculateSalary(){  
 return this.basicSalary + this.bonus;  
 }  
}

**Main Class**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q3 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter details: ");  
 Employee[] employees = new Employee[2];  
  
 int i = 0;  
 while (i<2){  
 Employee employee;  
  
 String[] details = scanner.nextLine().split(",");  
  
 if (details.length == 4){  
 employee = new Employee(details[1],details[2],Double.*parseDouble*(details[3]));  
 employees[i] = employee;  
 } else if (details.length == 5) {  
 employee = new Manager(details[1],details[2],Double.*parseDouble*(details[3]),Double.*parseDouble*(details[4]));  
 employees[i] = employee;  
 }  
 i++;  
 }  
 for (Employee emp : employees){  
 System.*out*.println(emp);  
 }}  
}

**Output:**

****

**Q4. Online Shopping Cart (Aggregation)**

**Code:**

**Product Class:**

package Module2;  
public class Product {  
 String productName;  
 double price;  
 int quantity;  
  
 Product(String productName, double price, int quantity){  
 this.productName = productName;  
 this.price = price;  
 this.quantity = quantity;  
 }  
  
 @Override  
 public String toString(){  
 return productName + " x " + quantity + " = " + "₹" + (price\*quantity);  
  
 }  
}

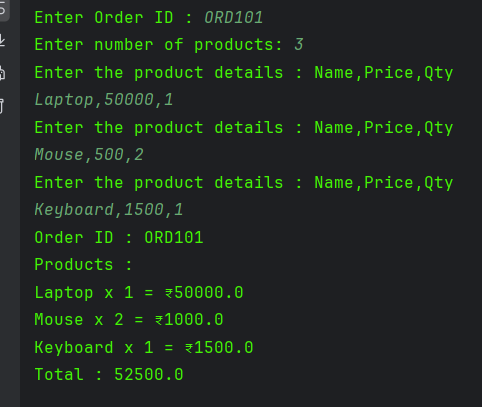
**Order Class:**

package Module2;  
public class Order {  
 String orderId ;  
 Product[] products;  
  
 Order(String orderId,Product[] products){  
 this.orderId = orderId;  
 this.products = products;  
 }  
  
 public double calculateTotal(){  
 double total = 0;  
 for (Product prod : products ){  
 total += prod.price \* prod.quantity ;  
 }  
 return total;}  
  
 @Override  
 public String toString(){  
 String temp = "";  
 for (Product prod : products){  
 temp += prod.toString()+"\n";  
 }  
 return "Order ID :" + this.orderId + "\n" + "Products : \n" + temp + "Total : " + calculateTotal();  
}}

**Main Class:**

package Module2;  
import java.util.Scanner;  
  
public class PHA\_M2\_Q4 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter Order ID : ");  
 String ordId = scanner.nextLine();  
 System.*out*.print("Enter number of products: ");  
 int numProd = scanner.nextInt();  
 scanner.nextLine();  
  
  
 Product[] prods = new Product[numProd];  
  
 for (int i = 0; i < numProd; i++){  
 System.*out*.println("Enter the product details : Name,Price,Qty ");  
 String[] prodDetails = scanner.nextLine().split(",");  
 prods[i] = new Product(prodDetails[0],Double.*parseDouble*(prodDetails[1]),Integer.*parseInt*(prodDetails[2]));  
 }  
  
 Order order = new Order(ordId,prods);  
 System.*out*.println(order);  
  
  
 }  
}

**Output:**

****

**Q5. Hospital Management (Multi-level Inheritance)**

**Code:**

**Person Class:**

package Module2;  
  
public class Person {  
 String name;  
 int age;  
  
 Person(String name,int age){  
 this.name = name;  
 this.age = age;  
 }  
  
 @Override  
 public String toString(){  
 return "Name: " + this.name + "\nAge : " + this.age;  
 }  
}

**Doctor Class:**

package Module2;  
  
public class Doctor extends Person{  
 String specialization;  
 Doctor(String name, int age, String specialization){  
 super(name,age);  
 this.specialization = specialization;  
 }  
  
 @Override  
 public String toString(){  
 return "Name: " + this.name + "\nAge : " + this.age + "\nSpecialization :" + this.specialization;  
 }  
}

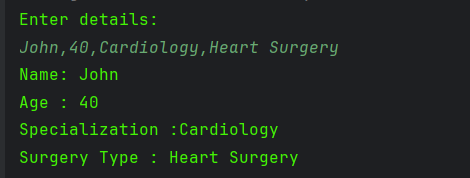
**Surgeon Class:**

package Module2;  
  
public class Surgeon extends Doctor{  
 String surgeryType;  
  
 Surgeon(String name,int age, String specialization, String surgeryType){  
 super(name,age,specialization);  
 this.surgeryType = surgeryType;  
 }  
  
 @Override  
 public String toString(){  
 return "Name: " + this.name + "\nAge : " + this.age + "\nSpecialization :" + this.specialization +  
 "\nSurgery Type : " + this.surgeryType;  
 }  
}

**Main Class:**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q5 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 Person person;  
 System.*out*.println("Enter details: ");  
 String[] details = scanner.nextLine().split(",");  
  
 if (details.length == 4){  
 person = new Surgeon(details[0],Integer.*parseInt*(details[1]),details[2],details[3]);  
 System.*out*.println(person);  
 }  
 else if (details.length == 3) {  
 person = new Doctor(details[0],Integer.*parseInt*(details[1]),details[2]);  
 System.*out*.println(person);  
  
 } else if (details.length == 2) {  
 person = new Person(details[0],Integer.*parseInt*(details[1]));  
 System.*out*.println(person);}  
 else {  
 System.*out*.println("Details format doesnt match");  
 }  
 }  
  
}

**Output:**

****

**Q6. Library – Book – Author (One-to-One Relationship)**

**Code:**

**Author Class:**

package Module2;  
  
public class Author {  
 String name;  
 String email;  
 char gender;  
  
 Author(String name, String email, char gender){  
 this.email = email;  
 this.name = name;  
 this.gender = gender;  
  
 }  
 String getDetails(){  
 return this.name + " (" + this.gender + ")" + "Email: " + this.email;  
 }  
}

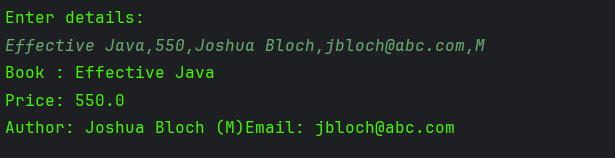
**Book Class:**

package Module2;  
  
public class Book {  
 String title;  
 Double price;  
 Author author;  
  
 Book(String title, Double price,String name,String email, char gender){  
 this.title = title;  
 this.price = price;  
 this.author = new Author(name,email,gender);  
 }  
  
 @Override  
 public String toString(){  
 return "Book : " + title + "\nPrice: " + price + "\nAuthor: " + this.author.getDetails();  
 }  
}

**Main Class:**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q6 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter details: ");  
 String[] details = scanner.nextLine().split(",");  
 Book book = new Book(details[0],Double.*parseDouble*(details[1]) ,details[2],details[3],details[4].charAt(0));  
 System.*out*.println(book);}  
}

**Output:**

****

**Q7. Sports League (Polymorphism – Method Overriding)**

**Code:**

**Team Class:**

package Module2;  
  
public class Team {  
 String name;  
 int matchesPlayed,wins,draws;  
  
 Team(String name,int matchesPlayed,int wins,int draws){  
 this.name = name;  
 this.matchesPlayed = matchesPlayed;  
 this.wins = wins;  
 this.draws = draws;  
 }  
  
 int calculatePoints(){return 0;};  
}

**Cricket Team Class:**

package Module2;  
  
public class CricketTeam extends Team{  
 CricketTeam(String name,int matchesPlayed,int wins,int draws){  
 super(name,matchesPlayed,wins,draws);  
 }  
 @Override  
 int calculatePoints(){  
 int win = 2;  
 int draw = 1;  
 return win\*this.wins + draw\*this.draws;  
 }  
}

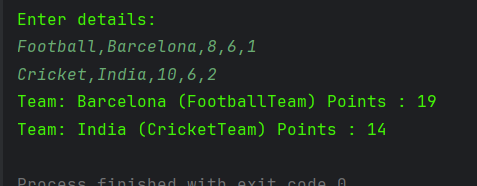
**Football Team Class:**

package Module2;  
  
public class FootballTeam extends Team{  
 FootballTeam(String name,int matchesPlayed,int wins,int draws){  
 super(name,matchesPlayed,wins,draws);  
 }  
 @Override  
 int calculatePoints(){  
 int win = 3;  
 int draw = 1;  
 return win\*this.wins + draw\*this.draws;  
 }  
}

**Main Class:**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q7 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 Team[] teams = new Team[2];  
 Team team;  
 int i = 0;  
 System.*out*.println("Enter details: ");  
 while (i<2){ //limiting to 2, can be asked from user  
 String[] details = scanner.nextLine().split(",");  
 if (details[0].equalsIgnoreCase("Cricket")){  
 team = new CricketTeam(details[1],Integer.*parseInt*(details[2]),Integer.*parseInt*(details[3]), Integer.*parseInt*(details[4]));  
 teams[i] = team;  
 } else if (details[0].equalsIgnoreCase("Football")) {  
 team = new FootballTeam(details[1],Integer.*parseInt*(details[2]),Integer.*parseInt*(details[3]), Integer.*parseInt*(details[4]));  
 teams[i] = team;}  
 i++;}  
 for (Team t : teams){  
 System.*out*.println("Team: " + t.name + " (" + t.getClass().getSimpleName() + ") " + "Points : " + t.calculatePoints());}  
 }  
 }

**Output:**

****

**Q8. Loan Management System (Abstraction + Polymorphism)**

**Code:**

**Loan Class:**

package Module2;  
  
abstract class Loan {  
 double principal;  
 double rate;  
 int time;  
  
 Loan(double principal, double rate, int time){  
 this.principal = principal;  
 this.rate = rate;  
 this.time = time;  
 }  
 abstract double calculateInterest();  
}

**Home Loan Class:**

package Module2;  
public class HomeLoan extends Loan{  
 static double *home\_rate* = 8;  
 HomeLoan(double principal, int time){  
 super(principal,*home\_rate*,time);  
 }  
  
 @Override  
 double calculateInterest() {  
 return ((this.principal \* this.rate \* this.time)/100);  
 }  
}

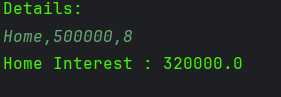
**Car Loan Class:**

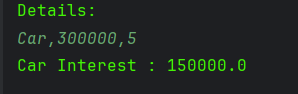
package Module2;  
  
public class CarLoan extends Loan{  
 static double *car\_rate* = 10;  
  
 CarLoan(double principal, int time){  
 super(principal,*car\_rate*,time);  
 }  
  
 @Override  
 double calculateInterest(){  
 return ((this.principal \* this.rate \* this.time)/100);  
 }  
}

**Main Class:**

package Module2;  
import java.util.Scanner;  
  
public class PHA\_M2\_Q8 {public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Details: ");  
 String[] detailArr = scanner.nextLine().split(",");  
 Loan loan;  
  
 if (detailArr[0].equalsIgnoreCase("Home")){  
 loan = new HomeLoan( Double.*parseDouble*(detailArr[1]) , Integer.*parseInt*(detailArr[2]));  
 System.*out*.println(detailArr[0]+ " Interest : " + loan.calculateInterest());  
 }  
 else if (detailArr[0].equalsIgnoreCase("Car")){  
 loan = new CarLoan( Double.*parseDouble*(detailArr[1]) , Integer.*parseInt*(detailArr[2]));  
 System.*out*.println(detailArr[0]+ " Interest : " + loan.calculateInterest());  
 }  
 else{  
 System.*out*.println("Wrong I/P");  
 }}  
}

**Output:**

****

****

**Q9. Online Course Platform (Association + Inheritance)**

**Code:**

**Course Class:**

package Module2;  
  
public class Course {  
 String courseName;  
 String duration;  
  
 Course(String courseName, String duration){  
 this.courseName = courseName;  
 this.duration = duration;  
 }  
  
 @Override  
 public String toString(){  
 return "Course: " + this.courseName + "(" + duration+")";  
 }  
}

**Student Class:**

package Module2;  
  
public class Student2{  
 String name;  
 Course enrolledCourse;  
  
 Student2(String name,Course course){  
 this.name = name;  
 this.enrolledCourse = course;}  
  
 @Override  
 public String toString(){  
 return "Student : " + this.name + " " + enrolledCourse.toString();  
 }  
}

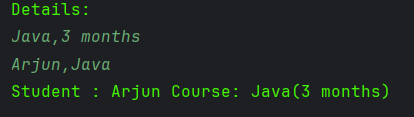
**Premium Student Class:**

package Module2;  
  
public class PremiumStudent extends Student2{  
 double discount;  
 PremiumStudent(String name,Course enrolledCourse,double discount){  
 super(name,enrolledCourse);  
 this.discount = discount;  
 }  
  
 public String toString(){  
 return "Premium " +super.toString() + " Discount" + ": " + ((int) this.discount) + "%";  
 }  
}

**Main Class:**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q9 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.println("Details: ");  
  
 String[] courseDetails = scanner.nextLine().split(",");  
 Course course = new Course(courseDetails[0],courseDetails[1]);  
 String[] studentDetails = scanner.nextLine().split(",");  
 Student2 student;  
  
 if (studentDetails.length == 2 && courseDetails[0].equalsIgnoreCase(studentDetails[1])){  
 student = new Student2(studentDetails[0],course);  
 System.*out*.println(student);  
 } else if (studentDetails.length == 3 && courseDetails[0].equalsIgnoreCase(studentDetails[1])) {  
 student = new PremiumStudent(studentDetails[0],course, Double.*parseDouble*(studentDetails[2]));  
 System.*out*.println(student);  
 }  
 else {  
 System.*out*.println("Wrong output format!");  
 }  
  
 }  
}

**Output:**

****

****

**Q10. Smart Home Devices (Interfaces + Polymorphism)**

**Code:**

**Device Interface:**

package Module2;  
  
public interface Device {  
  
 void turnOn();  
 void turnOff();  
}

**Fan Class:**

package Module2;  
  
public class Fan implements Device{  
  
 @Override  
 public void turnOn(){  
 System.*out*.println("Fan is now ON");  
 }  
  
 @Override  
 public void turnOff(){  
 System.*out*.println("Fan is now OFF");  
 }  
}

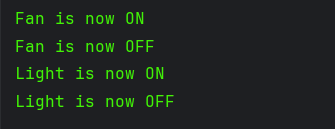
**Light Class:**

package Module2;  
  
public class Light implements Device{  
  
 @Override  
 public void turnOn(){  
 System.*out*.println("Light is now ON");  
 }  
  
 public void turnOff(){  
 System.*out*.println("Light is now OFF");  
 }  
}

**Main Class:**

package Module2;  
  
public class PHA\_M2\_Q10 {  
 public static void main(String[] args) {  
 Device[] devices = {new Fan(), new Light()};  
  
 for (Device device : devices){  
 device.turnOn();  
 device.turnOff();  
 }  
 }  
}

**Output:**

****

**Q11. University Hostel Allocation (One-to-One Relationship)**

**Code:**

**Student Class:**

package Module2;  
  
public class Student {  
 String name;  
 String course;  
 int rollNo;  
 Room room;  
  
 Student(String name,String course,int rollNo,Room room){  
 this.name = name;  
 this.course = course;  
 this.rollNo = rollNo;  
 this.room = room;  
 }  
  
 @Override  
 public String toString(){  
 return "Student: " + this.name + " (" + this.rollNo + ") " + course + "\n" + room.getDetails();  
 }  
}

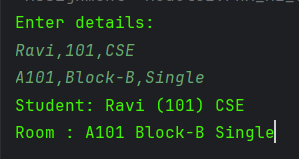
**Room Class:**

package Module2;  
  
public class Room {  
 String roomNumber;  
 String block;  
 String type;  
  
 Room(String roomNumber,String block, String type){  
 this.roomNumber = roomNumber;  
 this.block = block;  
 this.type = type;  
 }  
  
 String getDetails(){  
 return "Room : " + this.roomNumber + " " + this.block + " " + this.type;  
 }  
}

**Main Class:**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q11 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter details: ");  
 String[] studentDetails = scanner.nextLine().split(",");  
 String[] roomDetails = scanner.nextLine().split(",");  
  
 Student student = new Student(studentDetails[0],studentDetails[2],Integer.*parseInt*(studentDetails[1]),  
 new Room(roomDetails[0],roomDetails[1],roomDetails[2]));  
 System.*out*.println(student);  
 }  
}

**Output:**



**Q12. Vehicle Rental System (Inheritance + Overriding)**

**Code:**

**Vehicle Class:**

package Module2;  
  
public class Vehicle {  
 String regNo;  
 String brand;  
 Double baseRate;  
  
 Vehicle(String regNo,String brand, double baseRate){  
 this.regNo = regNo;  
 this.brand = brand;  
 this.baseRate = baseRate;  
 }  
}

**Car Class:**

package Module2;  
  
public class Car extends Vehicle {  
 Car(String regNo,String brand,double baseRate){  
 super(regNo,brand,baseRate);  
 super.baseRate = baseRate\*1.5;  
 }  
  
 @Override  
 public String toString(){  
 return "Car " + this.regNo + " " + this.brand + " Rent: " +  
 this.baseRate;  
 }  
}

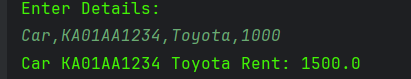
**Bike Class:**

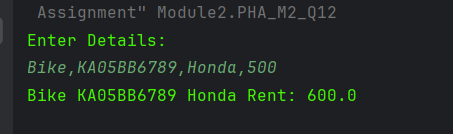
package Module2;  
  
public class Bike extends Vehicle{  
 Bike(String regNo,String brand,double baseRate){  
 super(regNo,brand,baseRate);  
 super.baseRate = baseRate\*1.2;  
 }  
  
 @Override  
 public String toString(){  
 return "Bike " + this.regNo + " " + this.brand + " Rent: " + this.baseRate;  
 }  
}

**Main Class:**

package Module2;  
  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q12 {  
 static Scanner *scanner* = new Scanner(System.*in*);  
 public static void main(String[] args) {  
 System.*out*.println("Enter Details: ");  
 String[] detail\_arr = *scanner*.nextLine().split(",");  
 Vehicle vehicle;  
  
 if (detail\_arr[0].equalsIgnoreCase("Car")){  
 vehicle = new Car(detail\_arr[1],detail\_arr[2],Double.*parseDouble*(detail\_arr[3]));  
 System.*out*.println(vehicle);  
 }  
 else if (detail\_arr[0].equalsIgnoreCase("Bike")){  
 vehicle = new Bike(detail\_arr[1],detail\_arr[2],Double.*parseDouble*(detail\_arr[3]));  
 System.*out*.println(vehicle);  
 }  
 else{  
 System.*out*.println("Wrong Input Format");  
 }  
 }  
}

**Output:**

****

****

**Q13. Hotel Reservation System (Aggregation)**

**Code:**

**Guest Class:**

package Module2;  
  
public class Guest {  
 String name;  
 int age;  
 String idProof;  
  
 Guest(String name,int age,String idProof){  
 this.name = name;  
 this.age = age;  
 this.idProof = idProof;  
 }  
  
 String getDetails(){  
 return this.name + "," + this.age + "," + this.idProof;  
 }  
}

**Reservation Class:**

package Module2;  
  
public class Reservation {  
 String reservationId;  
 String roomType;  
 Guest[] guests;  
  
 Reservation(String reservationId,String roomType, Guest[] guests){  
 this.reservationId = reservationId;  
 this.roomType = roomType;  
 this.guests = guests;  
 }  
  
 @Override  
 public String toString(){  
 String rDetails = "Reservation ID : " + this.reservationId + " Room : " + roomType;  
 String gDetails = "\nGuests: \n" ;  
 for (Guest g : guests){  
 gDetails += g.getDetails() + "\n";  
 }  
 return rDetails + gDetails;  
 }  
}

**Main Class:**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q13 {  
 public static void main(String[] args){  
 int n;  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter Reservation details: ");  
 String[] reservationDetails = scanner.nextLine().split(",");  
  
 n = Integer.*parseInt*(reservationDetails[2]);  
  
 Guest[] guests = new Guest[n];  
  
 for (int i = 0;i < n; i++ ){  
 System.*out*.printf("Enter Guest %d details: ",i+1);  
 String[] guestDetails = scanner.nextLine().split(",");  
 guests[i] = new Guest(guestDetails[0],Integer.*parseInt*(guestDetails[1]),guestDetails[2]);  
 }  
  
 Reservation reservation= new Reservation(reservationDetails[0],reservationDetails[1],guests);  
 System.*out*.println(reservation);  
  
 }  
}

**Output:**

****

**Q14. Banking ATM Simulation (Encapsulation)**

**Code:**

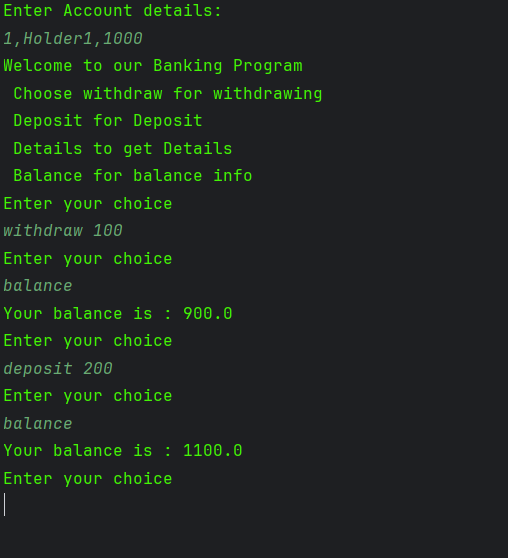
**Account Class**

package Module2;  
  
public class Account {  
 int accNo;  
 String holderName;  
 private double balance;  
  
 Account(int accNo,String holderName,double balance){  
 this.accNo = accNo;  
 this.holderName = holderName;  
 this.balance = balance;  
 }  
  
 String getAccountDetails(){  
 return "Account Number: " + accNo + "\nHolder Name: " + holderName + "\nBalance : " + balance;  
 }  
  
 void deposit(double amt){  
 if (amt < 0){  
 System.*out*.println("Wrong amount entered");  
 }  
 else {  
 this.balance += amt;  
 }  
 }  
  
 void withdraw(double amt){  
 if (amt > this.balance){  
 System.*out*.println("You cannot withdraw more than your balance");  
 }  
 else {  
 this.balance -= amt;  
 }  
 }  
  
  
 void showBalance(){  
 System.*out*.println("Your balance is : " + balance);  
 }  
}

**Main Class:**

package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q14 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter Account details: ");  
 String[] bankingDetails = scanner.nextLine().split(",");  
 Account account = new Account(Integer.*parseInt*(bankingDetails[0]),bankingDetails[1],  
 Double.*parseDouble*(bankingDetails[2]));  
 String choice;  
 System.*out*.println("Welcome to our Banking Program\n Choose withdraw for withdrawing\n" +  
 " Deposit for Deposit \n Details to get Details\n Balance for balance info");  
 do{  
 System.*out*.println("Enter your choice");  
 choice = scanner.nextLine();  
 String[] choiceArr = choice.split(" ");  
 if (choiceArr[0].equalsIgnoreCase("deposit")){  
 account.deposit(Double.*parseDouble*(choiceArr[1]));  
 } else if (choiceArr[0].equalsIgnoreCase("withdraw")) {  
 account.withdraw(Double.*parseDouble*(choiceArr[1]));  
 }  
 else if (choiceArr[0].equalsIgnoreCase("balance")){  
 account.showBalance();  
 } else if (choiceArr[0].equalsIgnoreCase("details")) {  
 System.*out*.println(account.getAccountDetails());  
 } else if (choiceArr[0].equalsIgnoreCase("quit")) {  
 System.*out*.println("Thanks for using");  
  
 }  
 else {  
 System.*out*.println("invalid choice");  
 }  
  
  
 }while (!choice.equalsIgnoreCase("quit"));  
 }  
}

**Output:**

****

**Q15. Passport – Citizen (One-to-One Relationship)**

**Code:**

**Citizen Class:**

package Module2;  
  
public class Citizen {  
 String name;  
 String dob;  
 String address;  
 Passport passport;  
  
 Citizen(String name,String dob, String address, String passportNo, String issueDate,String expiryDate){  
 this.name = name;  
 this.dob = dob;  
 this.address = address;  
 this.passport = new Passport(passportNo,issueDate,expiryDate);  
 }  
  
 @Override  
 public String toString(){  
 return "Citizen: " + this.name+ " DOB: "+ this.dob + " Address : " + this.address + "\n" + this.passport.getDetails();  
 }  
}

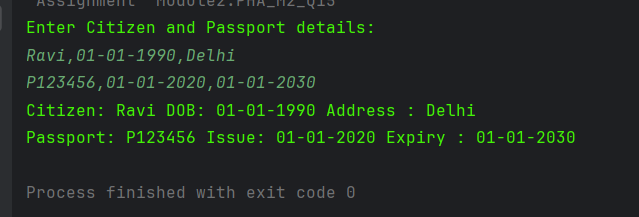
**Passport Class:**

package Module2;  
  
public class Passport {  
 String passportNo;  
 String issueDate;  
 String expiryDate;  
  
 Passport(String passportNo,String issueDate,String expiryDate){  
 this.expiryDate = expiryDate;  
 this.passportNo = passportNo;  
 this.issueDate = issueDate;  
 }  
  
 String getDetails(){  
 return "Passport: " + this.passportNo+" Issue: " + this.issueDate + " Expiry : " +this.expiryDate;  
 }  
}

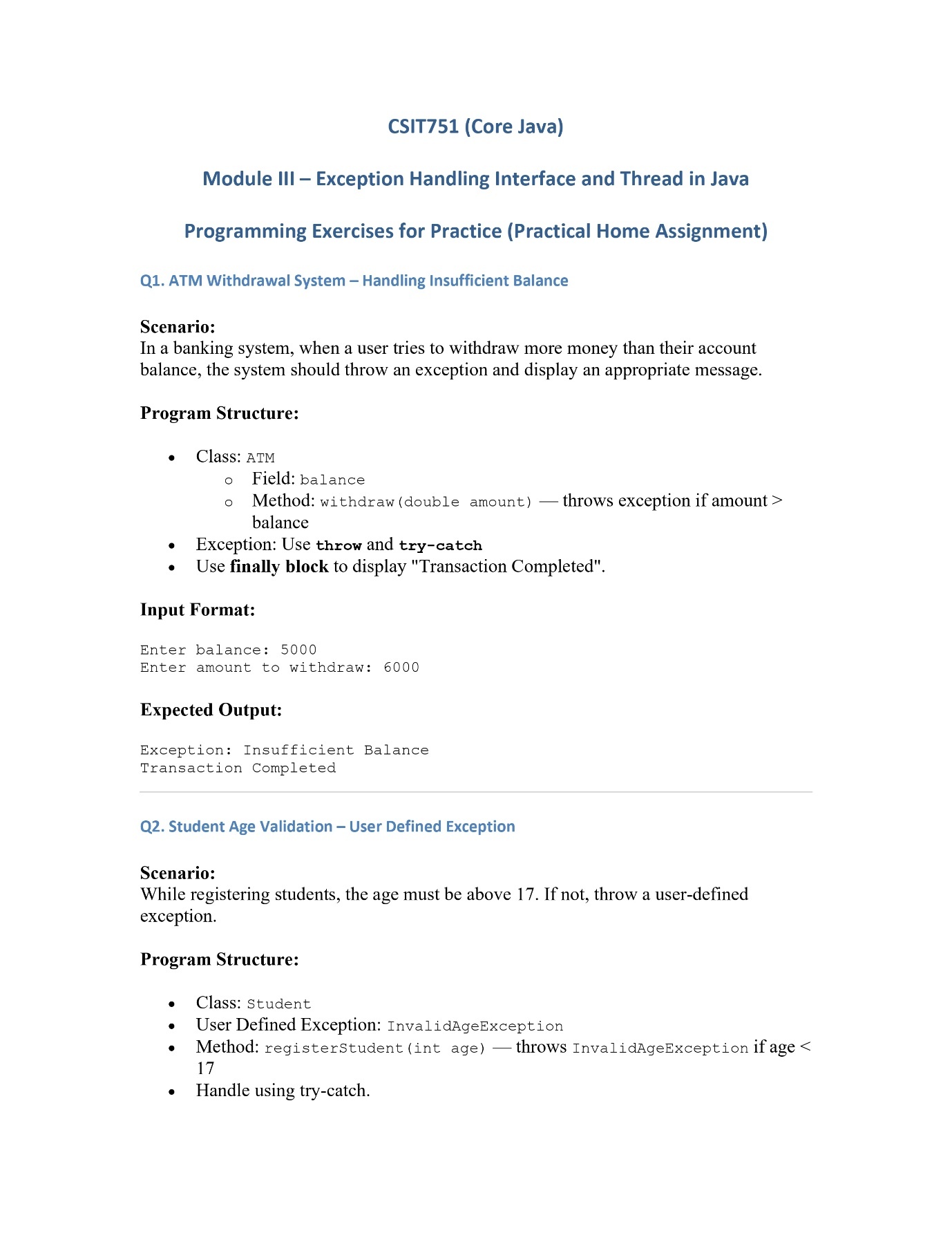
**Main Class:**

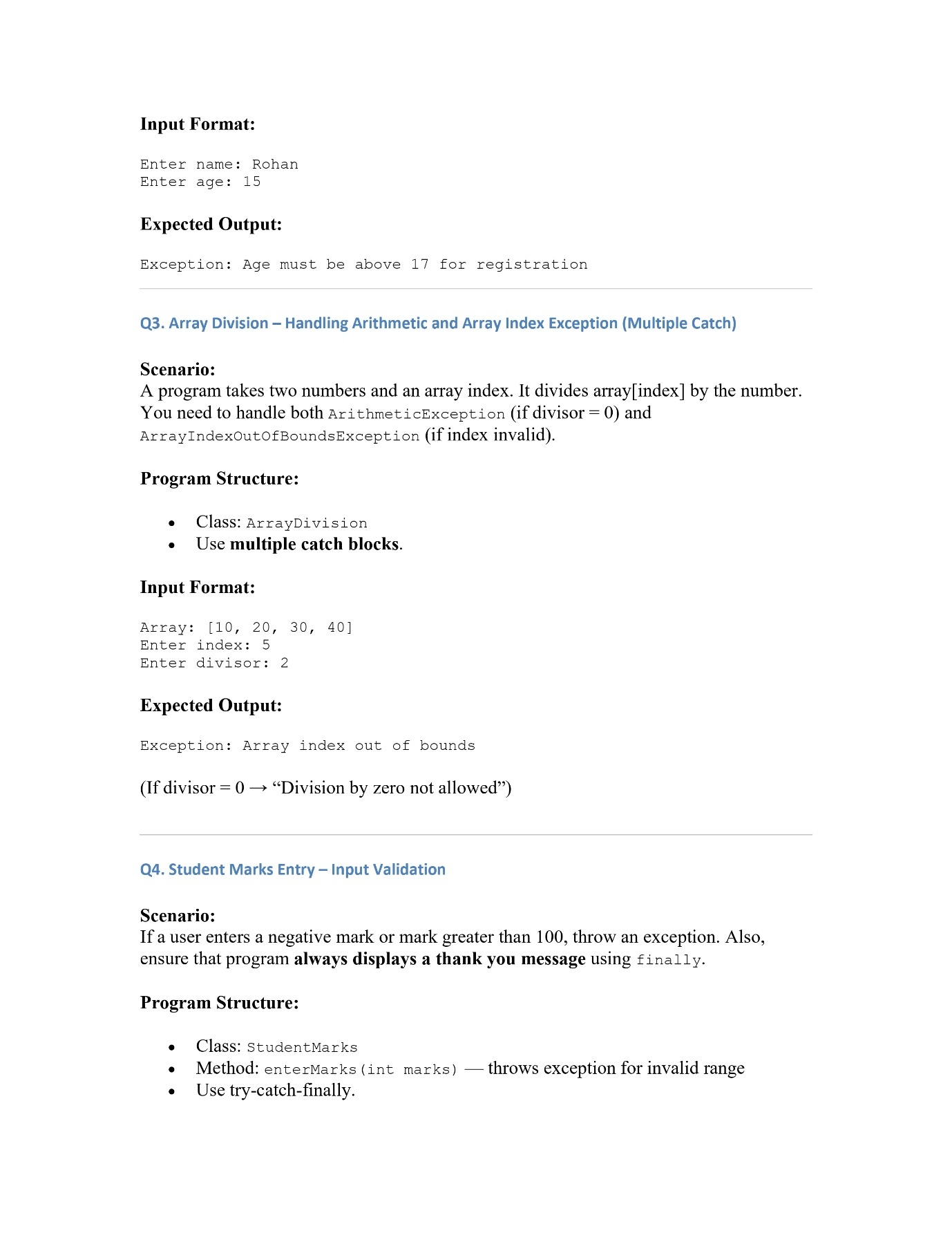
package Module2;  
  
import java.util.Scanner;  
  
public class PHA\_M2\_Q15 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter Citizen and Passport details: ");  
 String[] citizenDetails = scanner.nextLine().split(",");  
 String[] passportDetails = scanner.nextLine().split(",");  
  
 Citizen citizen = new Citizen(citizenDetails[0],citizenDetails[1],  
 citizenDetails[2],passportDetails[0],passportDetails[1],passportDetails[2]);  
 System.*out*.println(citizen);  
 }  
}

**Output:**

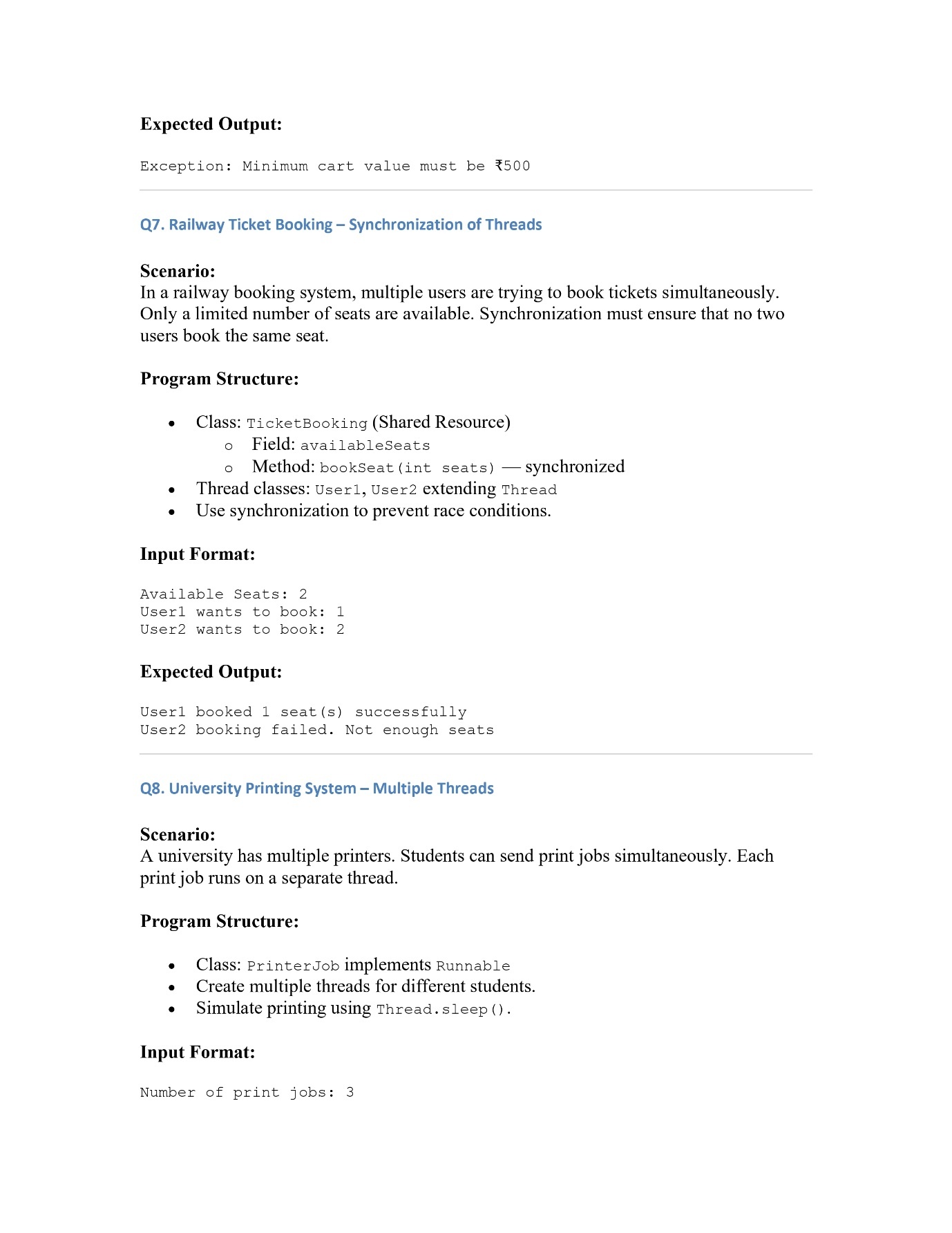
****

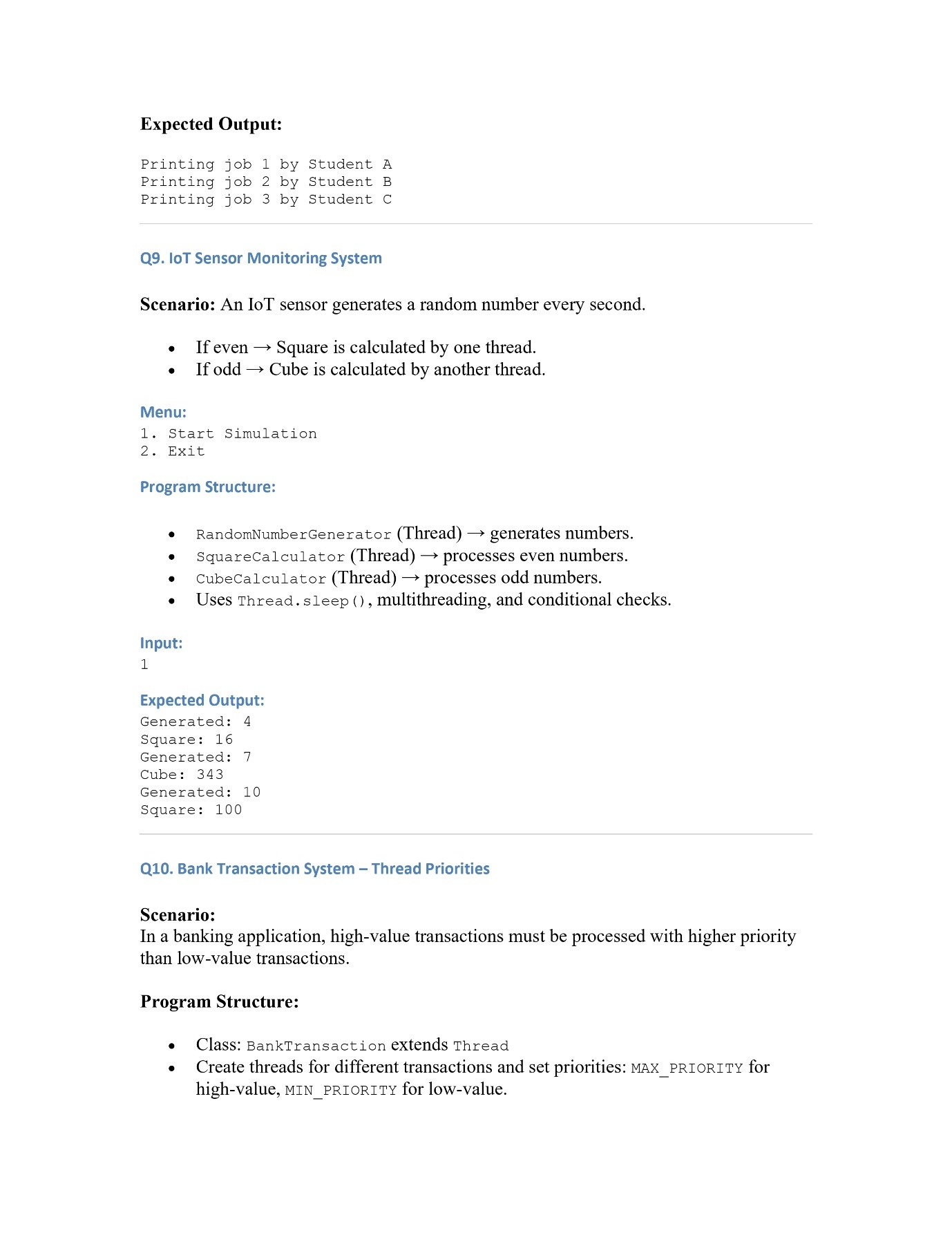
**Module 3**

****

****

****

****

****

****

**Q1. ATM Withdrawal System – Handling Insufficient Balance**

**Code:**

**ATM Class:**

package Module3;  
  
public class ATM {  
 double balance;  
 ATM(double balance){  
 this.balance = balance;  
 }  
  
 public void withdraw(double amt) throws WithdrawGreaterThanBalanceException {  
 if (this.balance < amt){  
 throw new WithdrawGreaterThanBalanceException();  
 }  
 this.balance -= amt;  
 }  
}

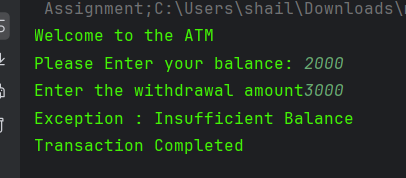
**WithdrawGreaterThanBalance Exception:**

package Module3;  
  
public class WithdrawGreaterThanBalanceException extends Exception{  
 WithdrawGreaterThanBalanceException(){  
 super("Exception : Insufficient Balance");  
 }  
}

**Main Class**

package Module3;  
  
import java.util.Scanner;  
  
public class PHA\_M3\_Q1 {  
 public static void main(String[] args){  
 System.*out*.println("Welcome to the ATM");  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Please Enter your balance: ");  
 double balance = scanner.nextDouble();  
 ATM atm = new ATM(balance);  
  
 try{  
 System.*out*.print("Enter the withdrawal amount");  
 double withdraw = scanner.nextDouble();  
 atm.withdraw(withdraw);  
 System.*out*.println("Your balance after withdrawal is : " + atm.balance );  
 }  
 catch (WithdrawGreaterThanBalanceException e){  
 System.*out*.println(e.getMessage());  
 }  
 finally {  
 System.*out*.println("Transaction Completed");  
 }  
 }  
  
}

**Output:**

****

**Q2. Student Age Validation – User Defined Exception**

**Code:**

**Student Class:**

package Module3;  
  
public class Student {  
 String name;  
 int age;  
  
 public void registerStudent(String name, int age) throws InvalidAgeException{  
 if (age <17){  
 throw new InvalidAgeException();  
 }  
 else{  
 this.name = name;  
 this.age = age;  
 System.out.println("Registration Successful");  
 }  
 }  
  
}

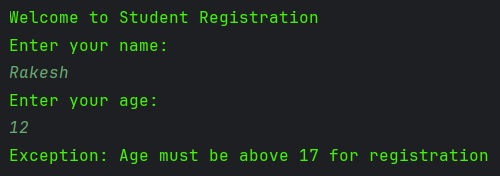
**InvalidAgeException**

package Module3;  
  
public class InvalidAgeException extends Exception{  
 InvalidAgeException(){  
 super("Exception: Age must be above 17 for registration");  
 }  
}

**Main Class**

package Module3;  
  
import java.util.Scanner;  
  
public class PHA\_M3\_Q2 {  
 public static void main(String[] args){  
 System.*out*.println("Welcome to Student Registration");  
 try(Scanner scanner = new Scanner(System.*in*)){  
 System.*out*.println("Enter your name: ");  
 String name = scanner.nextLine();  
 System.*out*.println("Enter your age: ");  
 int age = scanner.nextInt();  
 Student student = new Student();  
 student.registerStudent(name,age);  
 }  
 catch (InvalidAgeException e){  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**Output:**

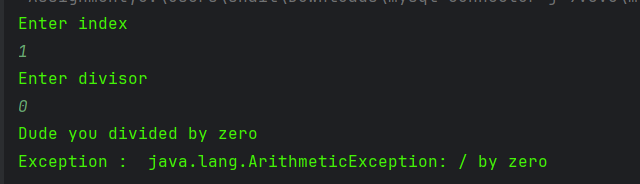
****

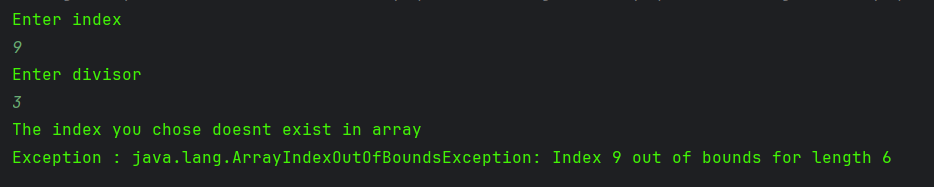
**Q3. Array Division – Handling Arithmetic and Array Index Exception (Multiple Catch)**

**Code:**

package Module3;  
  
import java.util.Scanner;  
  
public class PHA\_M3\_Q3 {  
 public static void main(String[] args){  
 int[] arr1 = {1,2,3,4,5,0};  
 try(Scanner scanner = new Scanner(System.*in*)){  
 System.*out*.println("Enter index");  
 int index = scanner.nextInt();  
 System.*out*.println("Enter divisor");  
 int divisor = scanner.nextInt();  
 int result = arr1[index]/divisor;  
 System.*out*.println("The result is + " + result);  
 }  
 catch (ArithmeticException e){  
 System.*out*.println("Dude you divided by zero\nException : " + e);  
 }  
 catch (ArrayIndexOutOfBoundsException e){  
 System.*out*.println("The index you chose doesnt exist in array \nException : " + e);  
 }  
 }  
}

**Output:**

****

****

**Q4. Student Marks Entry – Input Validation**

**Code:**

**StudentMarks Class**

package Module3;  
  
public class StudentMarks {  
  
 public void enterMarks(int marks) throws InvalidMarksException{  
 if (marks < 0 || marks > 100){  
 throw new InvalidMarksException();  
 }  
 else {  
 System.*out*.println("Entered Successfully");  
 }  
 }  
}

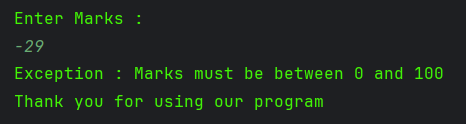
**InvalidMarksException**

package Module3;  
  
public class InvalidMarksException extends Exception{  
 InvalidMarksException(){  
 super("Exception : Marks must be between 0 and 100");  
 }  
}

**Main Class**

package Module3;  
  
import java.util.Scanner;  
  
public class PHA\_M3\_Q4 {  
 public static void main(String[] args){  
 try(Scanner scanner = new Scanner(System.*in*)){  
 StudentMarks studentMarks = new StudentMarks();  
 System.*out*.println("Enter Marks : ");  
 studentMarks.enterMarks(scanner.nextInt());  
 }  
 catch (InvalidMarksException e){  
 System.*out*.println(e.getMessage());  
 }  
 finally {  
 System.*out*.println("Thank you for using our program");  
 }  
 }  
}

**Output:**

****

**Q5. University Login System – Null Pointer Handling**

**Code:**

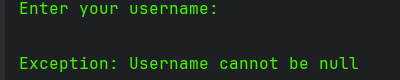
**University Login Class**

package Module3;  
  
public class UniversityLogin {  
 String username;  
 public void login(String username) throws NullPointerException{  
 if(username.isEmpty()){  
 this.username = null;  
 }  
 }  
}

**Main Class**

package Module3;  
  
import java.util.Scanner;  
  
public class PHA\_M3\_Q5 {  
 public static void main(String[] args){  
 UniversityLogin universityLogin = new UniversityLogin();  
 try(Scanner scanner = new Scanner(System.*in*)){  
 System.*out*.println("Enter your username: ");  
 universityLogin.login(scanner.nextLine());  
 System.*out*.println("Login Successful");  
 }  
 catch (NullPointerException e){  
 System.*out*.println("Exception: Username cannot be null");  
 }  
}}

**Output:**

****

**Q6. Online Shopping – Minimum Purchase Amount Exception**

**Code:**

**Online Shopping Class**

class OnlineShopping{  
 int amount;  
 public void placeOrder(int amount) throws MinimumAmountException{  
 if (amount < 500){  
 throw new MinimumAmountException();  
 }  
 else{  
 System.*out*.println("Order Placed");  
 this.amount = amount;  
 }  
 }  
}

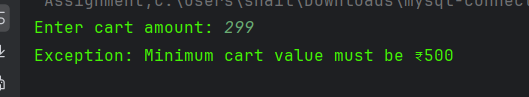
**MinimumAmountException**

class MinimumAmountException extends Exception{  
 MinimumAmountException(){  
 super("Exception: Minimum cart value must be ₹500");  
 }  
}

**Main Class**

public class PHA\_M3\_Q6 {  
 public static void main(String[] args){  
 OnlineShopping onlineShopping = new OnlineShopping();  
 try(Scanner scanner = new Scanner(System.in)){  
 System.out.print("Enter cart amount: ");  
 onlineShopping.placeOrder(scanner.nextInt());  
 }  
 catch (MinimumAmountException e){  
 System.out.println(e.getMessage());  
 }  
}}

**Output:**

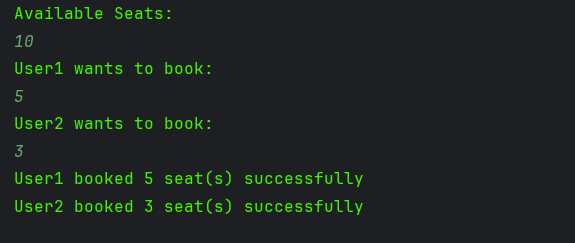
****

**Q7. Railway Ticket Booking – Synchronization of Threads**

**Code:**

package Module3;  
  
import javax.annotation.processing.SupportedSourceVersion;  
import java.util.Scanner;  
  
public class PHA\_M3\_Q7 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Available Seats: ");  
 TicketBooking t= new TicketBooking(scanner.nextInt());  
 System.*out*.println("User1 wants to book: ");  
 User1 user1 = new User1(t,scanner.nextInt());  
 System.*out*.println("User2 wants to book: ");  
 User2 user2 = new User2(t,scanner.nextInt());  
 user1.start();  
 user2.start();  
  
 }  
}

**Output:**

****

**Q8. University Printing System – Multiple Threads**

**Code:**

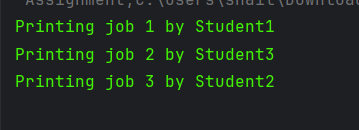
**Printer Job Class:**

package Module3;public class PrinterJob implements Runnable{  
 Thread t;  
 static int *counter* = 0;  
  
 PrinterJob(String name){  
 t = new Thread(this,name);  
 }  
  
 static synchronized void print(String name){  
 System.*out*.println("Printing job " + (*counter*+1) + " by " + name);  
 try{Thread.*sleep*(2000);}  
 catch(InterruptedException e){  
 System.*out*.println("Interrupt");  
 };  
 }  
  
 @Override  
 public void run(){  
 PrinterJob.*print*(this.t.getName());  
 *counter*++;  
 }  
}

**Main Class:**

package Module3;  
  
public class PHA\_M3\_Q8 {  
 public static void main(String[] args){  
 PrinterJob t1 = new PrinterJob("Student1");  
 PrinterJob t2 = new PrinterJob("Student2");  
 PrinterJob t3 = new PrinterJob("Student3");  
 t1.t.start();  
 t2.t.start();  
 t3.t.start();  
 }  
}

**Output:**

****

**Q9. IoT Sensor Monitoring System**

**Code:**

**RandGen class**

package Module3;  
  
public class RandGen implements Runnable{  
 Thread t;  
 IoTSensor iot;  
 RandGen(IoTSensor iot){  
 this.iot = iot;  
 t = new Thread(this);  
 }  
  
 @Override  
 public void run(){  
 try {  
 while (true){  
 Thread.*sleep*(1000);  
 iot.generator();  
 }  
 } catch (InterruptedException e){  
 System.*out*.println("Caught exception");  
 }  
 }  
}

**OddCalc Class**

package Module3;  
  
public class OddCalc implements Runnable {  
 IoTSensor iot;  
 Thread t;  
  
 OddCalc(IoTSensor iot){  
 this.iot = iot;  
 t = new Thread(this);  
  
 }  
  
 @Override  
 public void run(){  
 while (true) iot.oddCalculator();  
 }  
}

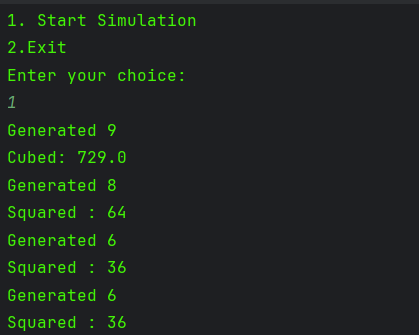
**EvenCalc Class**

package Module3;  
  
public class EvenCalc implements Runnable {  
 IoTSensor iot;  
 Thread t;  
  
 EvenCalc(IoTSensor iot){  
 this.iot = iot;  
 t = new Thread(this);  
  
 }  
  
 @Override  
 public void run(){  
 while (true) iot.evenCalculator();  
 }  
}

**Main Class:**

package Module3;  
  
import java.util.Scanner;  
  
public class PHA\_M3\_Q9 {  
 public static void main(String[] args){  
  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("1. Start Simulation\n2.Exit");  
 System.*out*.println("Enter your choice: ");  
 int choice = scanner.nextInt();  
 if (choice == 1){  
 IoTSensor iot = new IoTSensor();  
 RandGen randGen = new RandGen(iot);  
 OddCalc oddCalc = new OddCalc(iot);  
 EvenCalc evenCalc = new EvenCalc(iot);  
  
 randGen.t.start();  
 evenCalc.t.start();  
 oddCalc.t.start();  
 }  
 else {  
 System.*out*.println("You exited successfully");  
 }  
  
 }}

**Output:**

****

**Q10. Bank Transaction System – Thread Priorities**

**Code:**

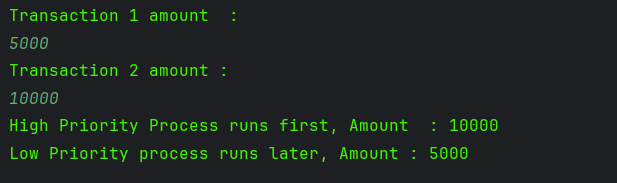
**Bank Transaction Class**

package Module3;  
  
public class BankTransaction extends Thread{  
 int amt;  
 BankTransaction(int amt){  
 this.amt = amt;  
 if (amt > 5000){  
 this.setPriority(*MAX\_PRIORITY*);  
 }  
 }  
  
 @Override  
 public void run(){  
 if (this.getPriority() == 10){  
 System.*out*.println("High Priority Process runs first, Amount : " + this.amt);  
 }  
 else {  
 System.*out*.println("Low Priority process runs later, Amount : " + this.amt);  
 }  
 }  
}

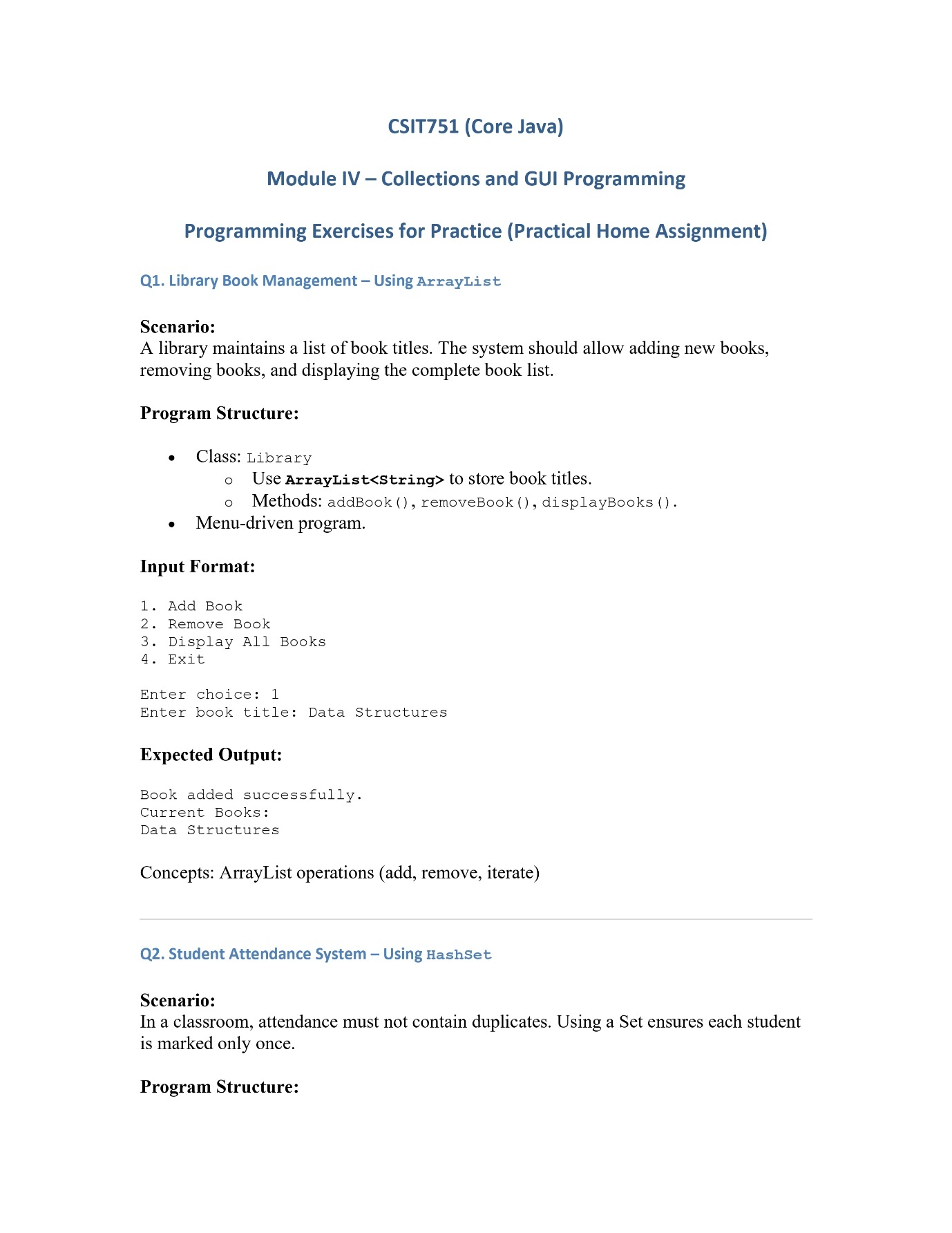
**Main Class**

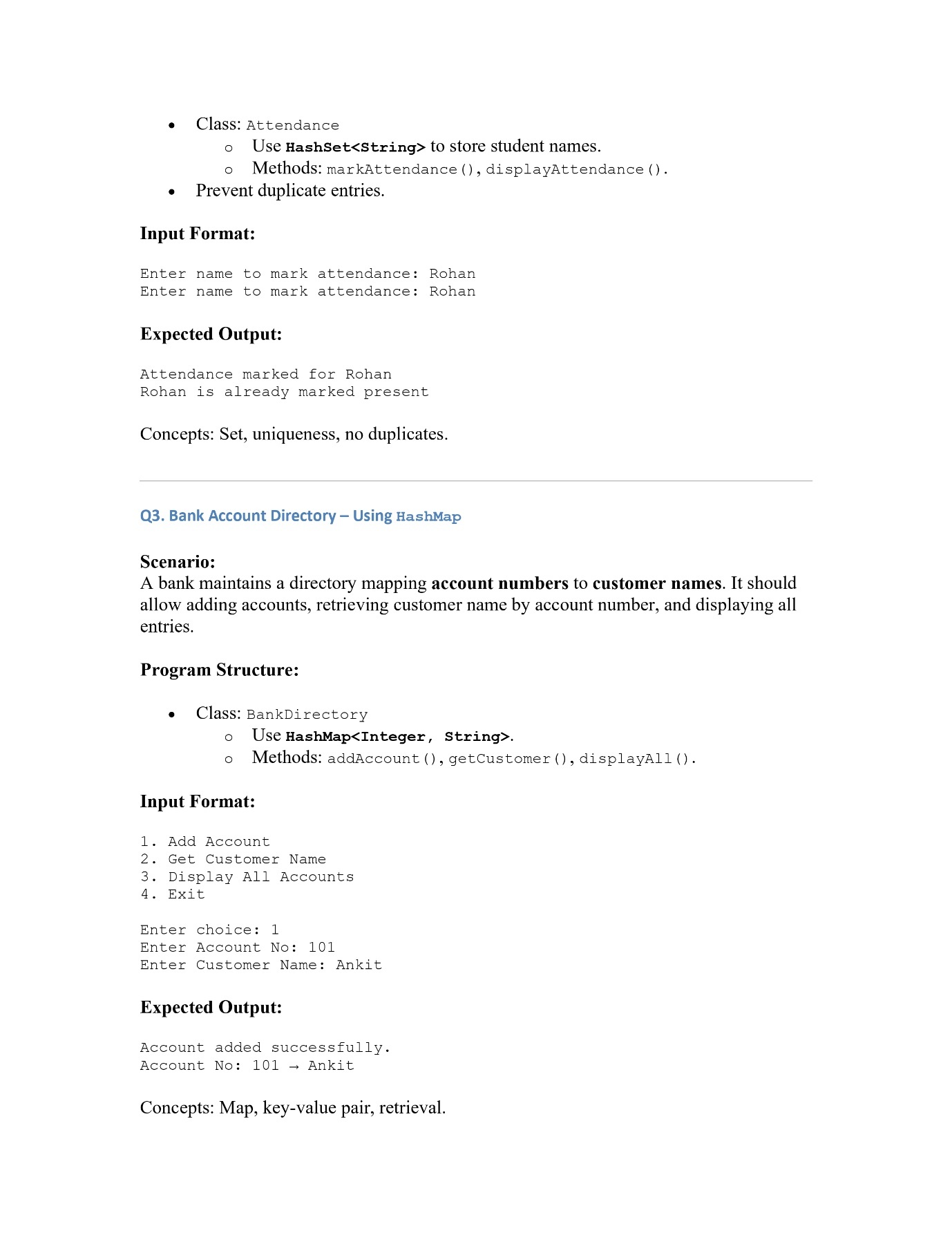
package Module3;  
  
import java.util.Scanner;  
  
public class PHA\_M3\_Q10 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Transaction 1 amount : ");  
 int amount1 = scanner.nextInt();  
 System.*out*.println("Transaction 2 amount : ");  
 int amount2 = scanner.nextInt();  
 Thread t1 = new BankTransaction(amount1);  
 Thread t2 = new BankTransaction(amount2);  
 t1.start();  
 t2.start();  
 }  
}

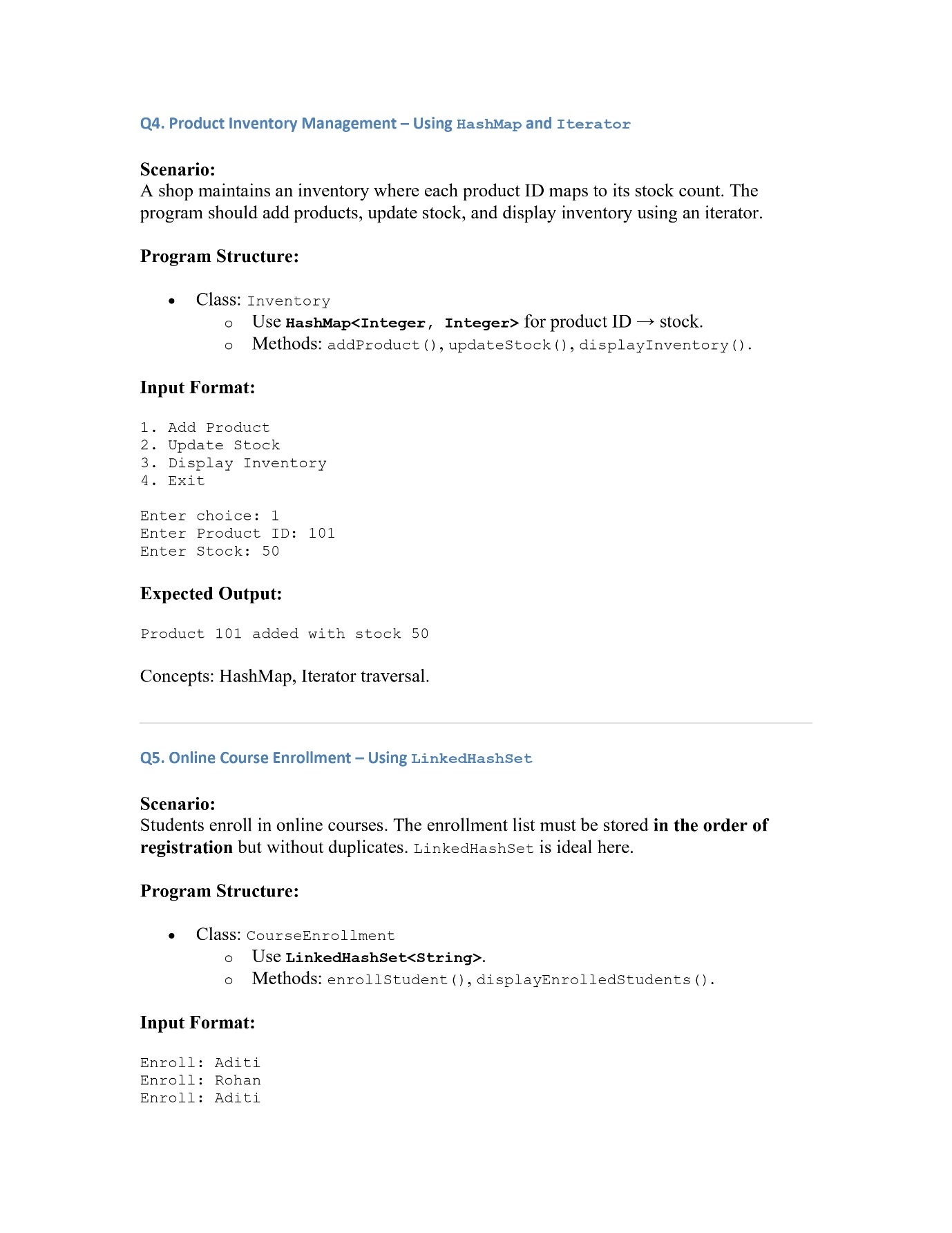
**Output:**

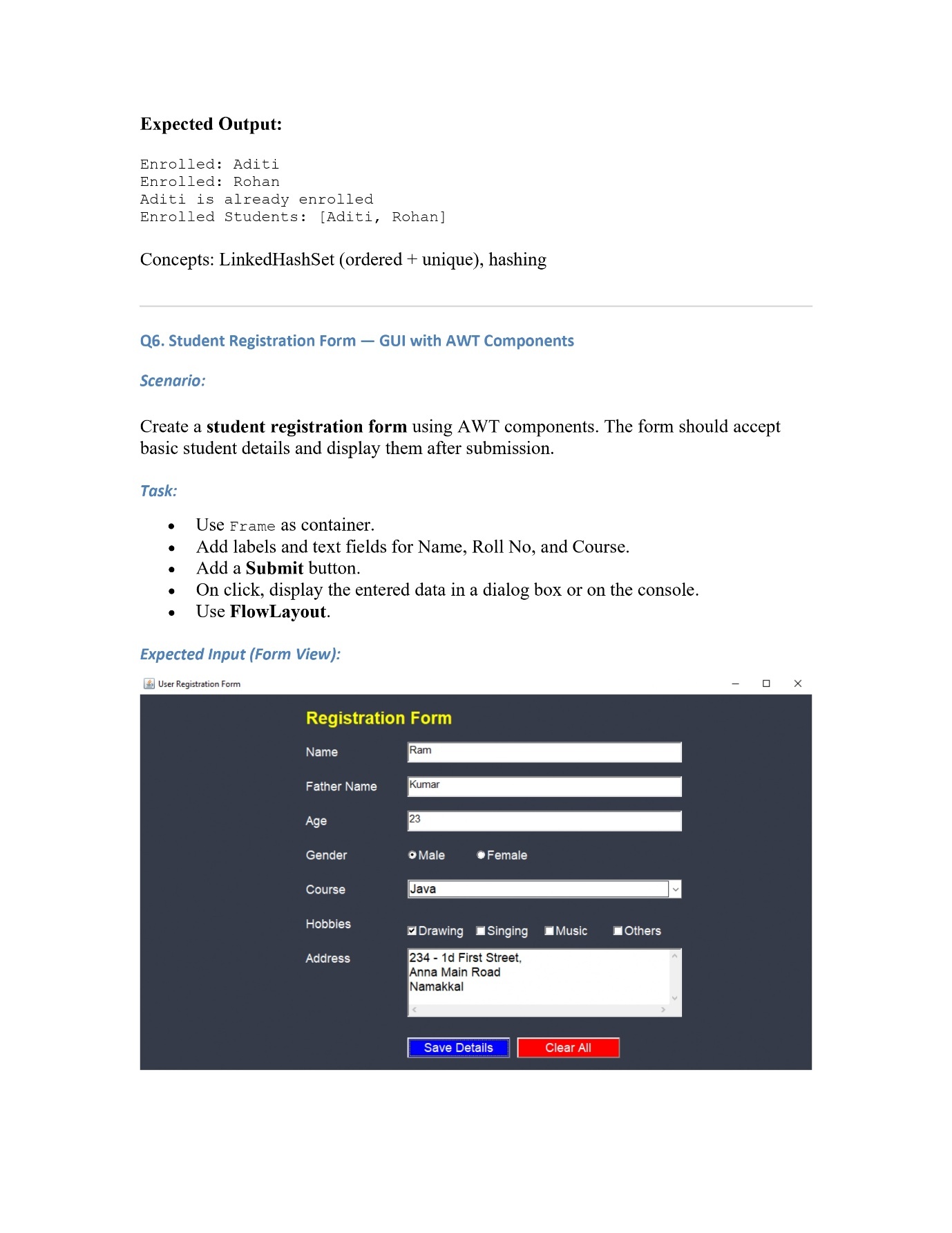
****

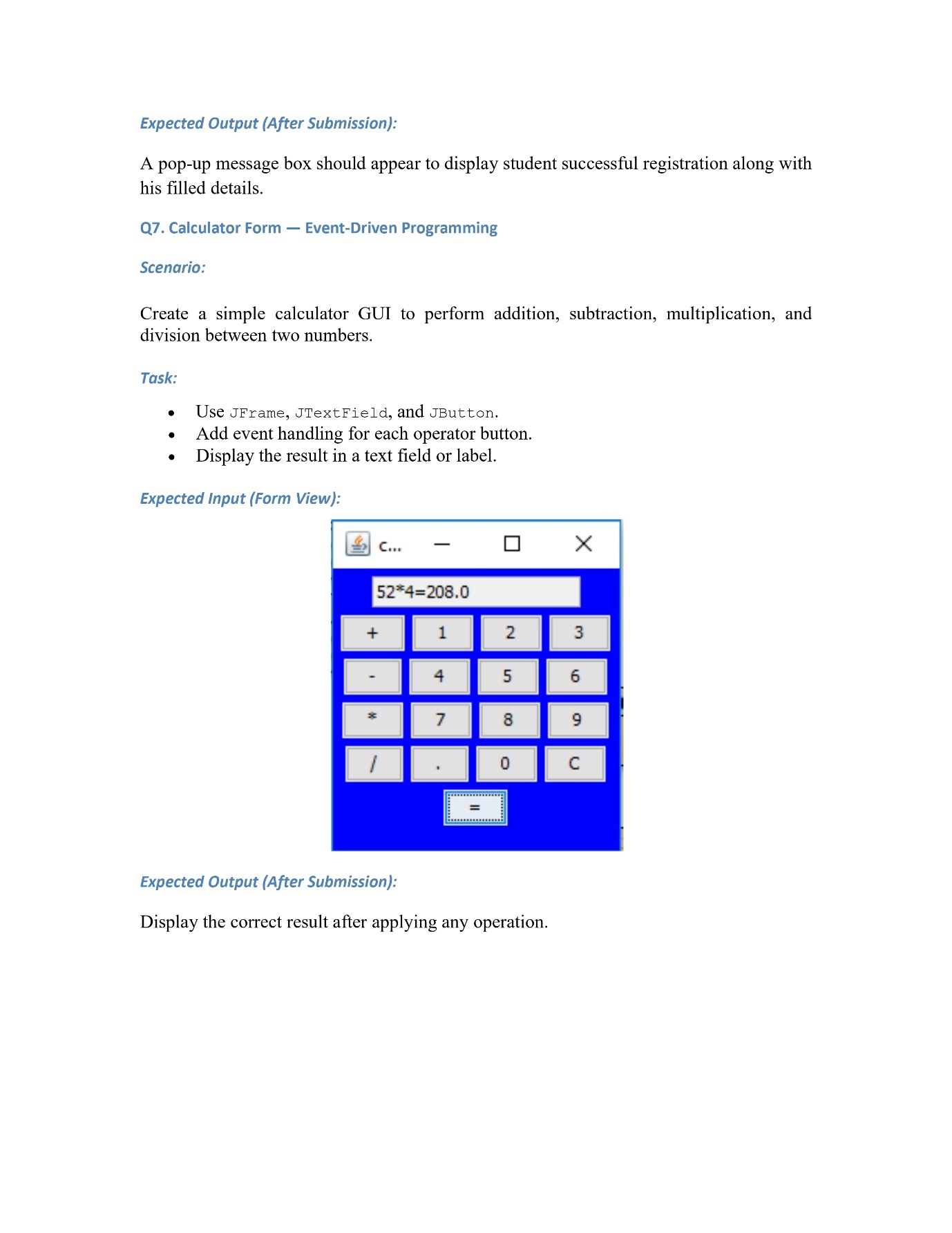
**Module 4**

****

****

****

****

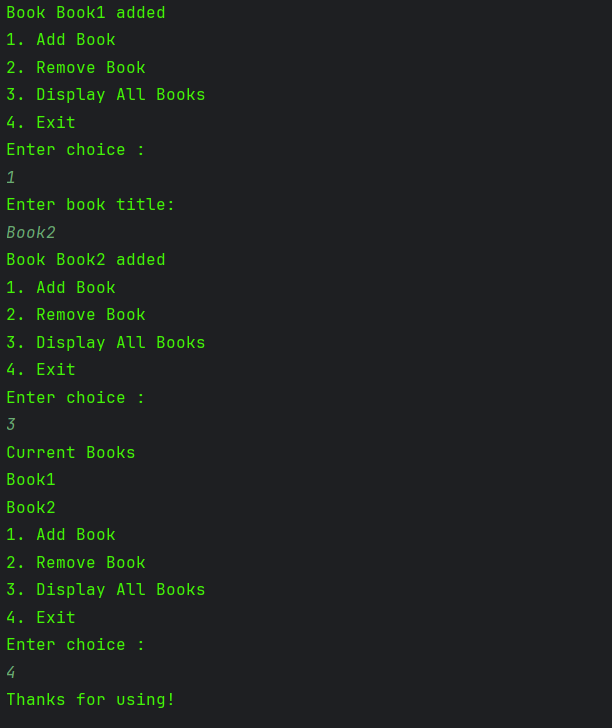
****

**Q1. Library Book Management – Using ArrayList**

**Code:**

package Module4;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class PHA\_M4\_Q1 {  
 static ArrayList<String> *books* = new ArrayList<>();  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Welcome to Library");  
 int choice;  
 do {  
 System.*out*.println("1. Add Book\n" +  
 "2. Remove Book\n" +  
 "3. Display All Books\n" +  
 "4. Exit");  
 System.*out*.println("Enter choice : ");  
 choice = scanner.nextInt();  
 scanner.nextLine();  
 switch (choice){  
 case 1 ->{  
 System.*out*.println("Enter book title: ");  
 String book = scanner.nextLine();  
 *addBook*(book);  
 }  
 case 2 ->{  
 System.*out*.println("Enter book title removed");  
 String book = scanner.nextLine();  
 *removeBook*(book);  
 }  
  
 case 3 -> *displayBooks*();  
 }  
 } while (choice!=4);  
 System.*out*.println("Thanks for using!");  
 }  
 static void addBook(String book){  
 *books*.add(book);  
 System.*out*.println("Book " + book + " added");  
 }  
  
 static void removeBook(String book){  
 *books*.remove(book);  
 System.*out*.println("Book " + book + " removed");  
 }  
  
 static void displayBooks(){  
 System.*out*.println("Current Books");  
 for (String book : *books*){  
 System.*out*.println(book);  
 }  
 }  
}

**Output:**

****

**Q2. Student Attendance System – Using HashSet**

**Code:**

package Module4;  
import java.util.Scanner;  
import java.util.HashMap;  
public class PHA\_M4\_Q2 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 HashMap<String,Boolean> attendanceSheet= new HashMap<>();  
 while (true){  
 System.*out*.println("Enter name to mark attendance: ");  
 String name = scanner.nextLine();  
 if (attendanceSheet.containsKey(name)){  
 System.*out*.println("Already marked");  
 break;  
 }  
 else {  
 attendanceSheet.put(name,true);  
 System.*out*.println("Attendance marked for " + name);  
 }  
 }  
 }  
}

**Output:**

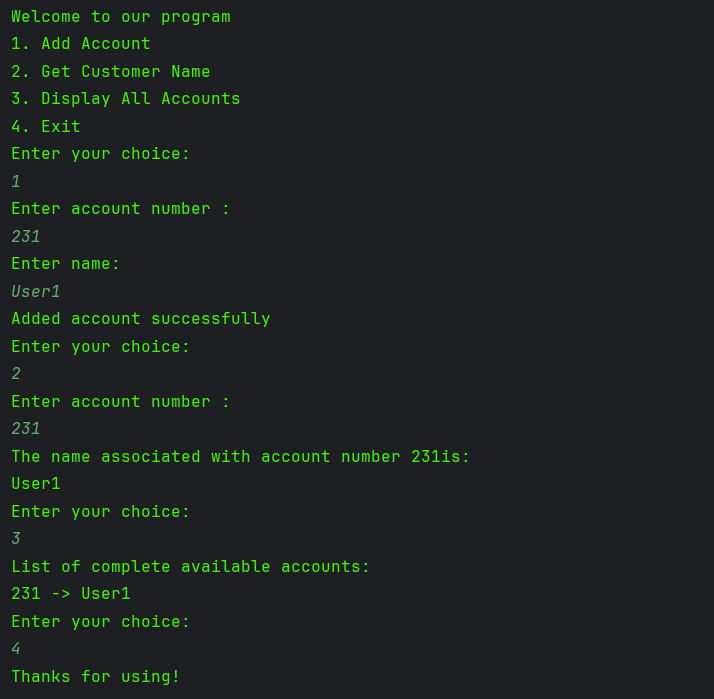
****

**Q3. Bank Account Directory – Using HashMap**

**Code:**

package Module4;  
  
import java.util.Scanner;  
import java.util.HashMap;  
  
  
public class PHA\_M4\_Q3 {  
 static Scanner *scanner* = new Scanner(System.*in*);  
 static HashMap<Integer,String> *accounts* = new HashMap<>();  
 public static void main(String[] args){  
 int choice;  
 System.*out*.println("Welcome to our program");  
 System.*out*.println("1. Add Account\n" +  
 "2. Get Customer Name\n" +  
 "3. Display All Accounts\n" +  
 "4. Exit");  
 do{  
 System.*out*.println("Enter your choice: ");  
 choice = *scanner*.nextInt();  
 *scanner*.nextLine();  
  
 switch (choice){  
 case 1 -> {  
 System.*out*.println("Enter account number :");  
 int accno = *scanner*.nextInt();  
 *scanner*.nextLine();  
 System.*out*.println("Enter name: ");  
 String name = *scanner*.nextLine();  
 *accounts*.put(accno,name);  
 System.*out*.println("Added account successfully");}  
 case 2 -> {  
 System.*out*.println("Enter account number : ");  
 int accno = *scanner*.nextInt();  
 System.*out*.println("The name associated with account number " + accno + "is: ");  
 System.*out*.println(*accounts*.get(accno));  
 }  
 case 3 -> {  
 System.*out*.println("List of complete available accounts: ");  
 for (int accno : *accounts*.keySet()){  
 System.*out*.println(accno + " -> " + *accounts*.get(accno));  
 }  
 }  
 }  
 } while (choice !=4);  
 System.*out*.println("Thanks for using!");  
 }  
}

**Output:**

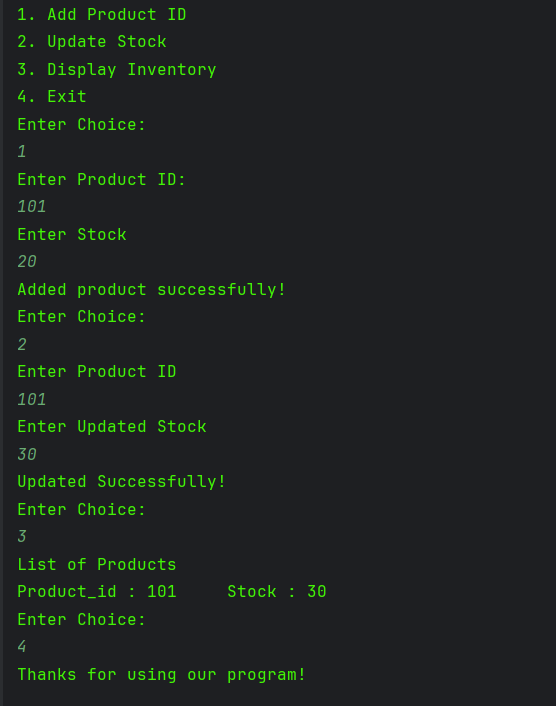
****

**Q4. Product Inventory Management – Using HashMap and Iterator**

**Code:**

package Module4;  
import java.util.HashMap;  
import java.util.Iterator;  
import java.util.Map;  
import java.util.Scanner;  
public class PHA\_M4\_Q4 {  
 static HashMap<Integer,Integer> *products* = new HashMap<>();  
 public static void main(String[] args){  
 int choice;  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("1. Add Product\n" +  
 "2. Update Stock\n" +  
 "3. Display Inventory\n" +  
 "4. Exit");  
 do{  
 System.*out*.println("Enter Choice: ");  
 choice = scanner.nextInt();  
  
 switch (choice){  
 case 1 -> {  
 System.*out*.println("Enter Product: ");  
 int prod = scanner.nextInt();  
 System.*out*.println("Enter Stock");  
 int stock = scanner.nextInt();  
 *products*.put(prod,stock);  
 System.*out*.println("Added product successfully!");  
 }  
 case 2 -> {  
 System.*out*.println("Enter Product");  
 int prod = scanner.nextInt();  
 if (*products*.containsKey(prod)) {  
 System.*out*.println("Enter Updated Stock");  
 int stock = scanner.nextInt();  
 *products*.replace(prod, stock);  
 System.*out*.println("Updated Successfully!");  
 }  
 else {  
 System.*out*.println("There's no matching product!");  
 }  
 }  
  
 case 3 ->{  
 System.*out*.println("List of Products");  
 Iterator<Map.Entry<Integer,Integer>> iterator = *products*.entrySet().iterator();  
  
 while (iterator.hasNext()){  
 Map.Entry<Integer,Integer> entry = iterator.next();  
 System.*out*.println("Product\_name : " + entry.getKey() +  
 " Stock : " + entry.getValue());  
 }  
 }  
 case 4 -> {  
 System.*out*.println("Thanks for using our program!");  
 }  
 default -> {  
 System.*out*.println("You have entered the wrong value!");  
 }  
 }  
 }while (choice!=4);  
 }}

**Output:**

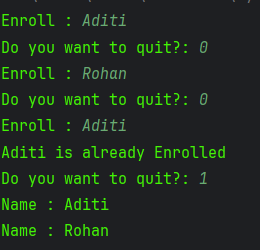
****

**Q5. Online Course Enrollment – Using LinkedHashSet**

**Code:**

package Module4;  
  
import java.util.LinkedHashSet;  
import java.util.Scanner;  
  
public class PHA\_M4\_Q5 {  
 static LinkedHashSet<String> *enrolled* = new LinkedHashSet<>();  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 int choice = 0;  
 while (choice!=2){  
 System.*out*.print("Enroll : ");  
 String name = scanner.nextLine();  
 System.*out*.print("Choice: ");  
 choice = scanner.nextInt();  
 scanner.nextLine();  
 *addEnrolledStudent*(name);  
 }  
 *displayEnrolledStudent*();  
 }  
  
 static void addEnrolledStudent(String name){  
 *enrolled*.add(name);  
 }  
  
 static void displayEnrolledStudent(){  
 for(String name : *enrolled* ){  
 System.*out*.println("Name : " + name);  
 }  
 }  
}

**Output:**

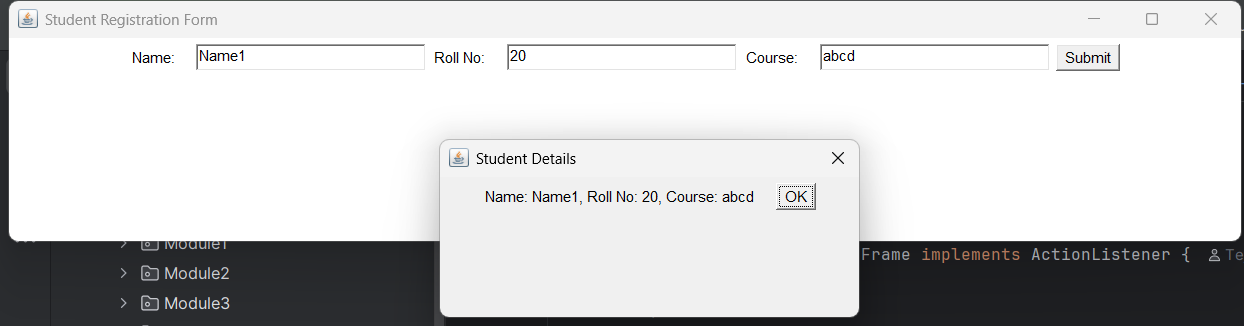
****

**Q6. Student Registration Form — GUI with AWT Components**

**Code:**

package Module4;  
  
import java.awt.\*;  
import java.awt.event.\*;  
  
public class PHA\_M4\_Q6 extends Frame implements ActionListener {  
  
 // Components  
 Label nameLabel, rollLabel, courseLabel;  
 TextField nameField, rollField, courseField;  
 Button submitButton;  
  
 // Constructor  
 public PHA\_M4\_Q6() {  
 // Frame setup  
 setTitle("Student Registration Form");  
 setSize(1000, 200);  
 setLayout(new FlowLayout());  
  
 // Create Labels  
 nameLabel = new Label("Name:");  
 rollLabel = new Label("Roll No:");  
 courseLabel = new Label("Course:");  
  
 // Create TextFields  
 nameField = new TextField(20);  
 rollField = new TextField(20);  
 courseField = new TextField(20);  
  
 // Create Button  
 submitButton = new Button("Submit");  
 submitButton.addActionListener(this);  
  
 // Add components to frame  
 add(nameLabel);  
 add(nameField);  
 add(rollLabel);  
 add(rollField);  
 add(courseLabel);  
 add(courseField);  
 add(submitButton);  
  
 // Window closing  
 addWindowListener(new WindowAdapter() {  
 public void windowClosing(WindowEvent we) {  
 dispose();  
 }  
 });  
  
 setVisible(true);  
 }  
  
 // Handle button click  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == submitButton) {  
 String name = nameField.getText();  
 String roll = rollField.getText();  
 String course = courseField.getText();  
  
 // Display in dialog box  
 Dialog d = new Dialog(this, "Student Details", true);  
 d.setLayout(new FlowLayout());  
 Label details = new Label("Name: " + name + ", Roll No: " + roll + ", Course: " + course);  
 Button ok = new Button("OK");  
 ok.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 d.setVisible(false);  
 }  
 });  
  
 d.add(details);  
 d.add(ok);  
 d.setSize(350, 150);  
 d.setVisible(true);  
  
 // Optional: Print to console  
 System.*out*.println("Student Details Submitted:");  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Roll No: " + roll);  
 System.*out*.println("Course: " + course);  
 }  
 }  
  
 // Main method  
 public static void main(String[] args) {  
 new PHA\_M4\_Q6();  
 }  
}

**Output:**

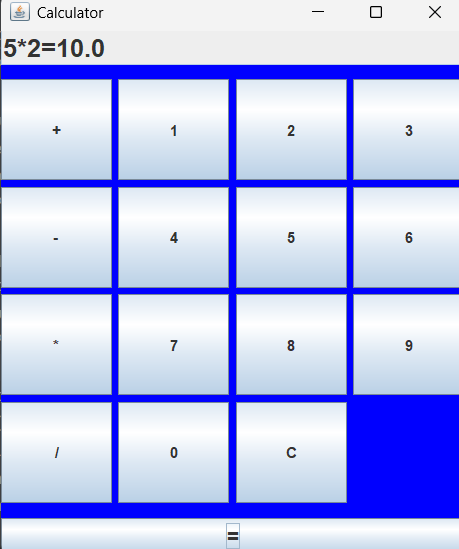
****

**Q7. Calculator Form — Event-Driven Programming**

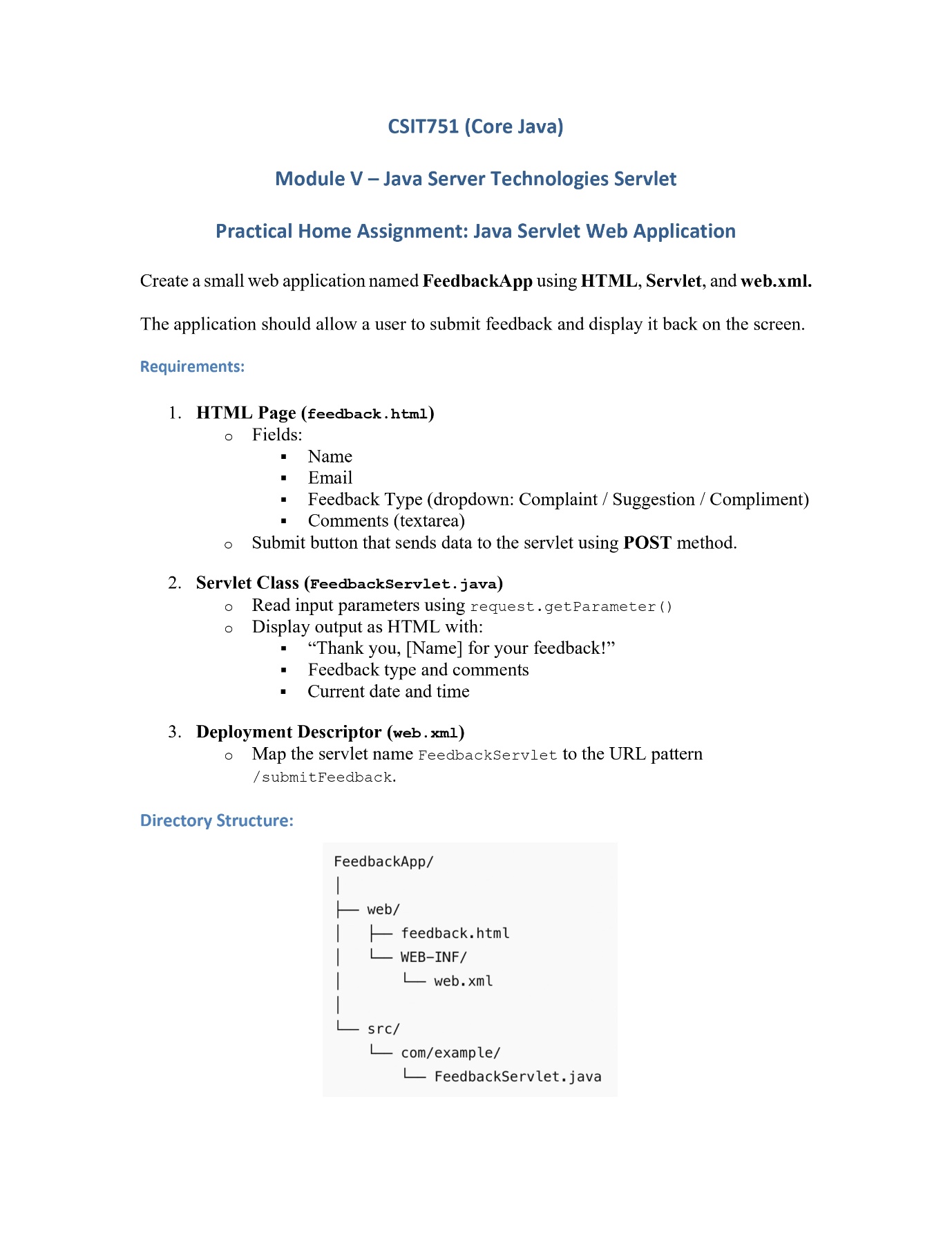
**Code:**

package Module4;  
import javax.swing.\*;  
import java.awt.\*;  
  
public class PHA\_M4\_Q7 {  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("Calculator");  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setSize(300, 400);  
  
 // main background  
 JPanel mainPanel = new JPanel(new BorderLayout(10, 10));  
 mainPanel.setBackground(Color.*BLUE*);  
 frame.setContentPane(mainPanel);  
  
 // text field on top  
 JTextField textField = new JTextField();  
 textField.setEditable(false);  
 textField.setFont(new Font("Arial", Font.*BOLD*, 20));  
 mainPanel.add(textField, BorderLayout.*NORTH*);  
  
 // button grid in center  
 JPanel buttonPanel = new JPanel(new GridLayout(4, 4, 5, 5));  
 buttonPanel.setBackground(Color.*BLUE*);  
 mainPanel.add(buttonPanel, BorderLayout.*CENTER*);  
  
 // equal button at bottom  
 JButton equButton = new JButton("=");  
 equButton.setFont(new Font("Arial", Font.*BOLD*, 16));  
 mainPanel.add(equButton, BorderLayout.*SOUTH*);  
  
 JButton[] numberButtons = new JButton[10];  
 for (int i = 0; i < 10; i++) {  
 int num = i;  
 numberButtons[i] = new JButton(String.*valueOf*(num));  
 numberButtons[i].addActionListener(e -> textField.setText(textField.getText() + num));  
 }  
  
 JButton addButton = new JButton("+");  
 JButton subButton = new JButton("-");  
 JButton mulButton = new JButton("\*");  
 JButton divButton = new JButton("/");  
 JButton clrButton = new JButton("C");  
  
 buttonPanel.add(addButton);  
 buttonPanel.add(numberButtons[1]);  
 buttonPanel.add(numberButtons[2]);  
 buttonPanel.add(numberButtons[3]);  
 buttonPanel.add(subButton);  
 buttonPanel.add(numberButtons[4]);  
 buttonPanel.add(numberButtons[5]);  
 buttonPanel.add(numberButtons[6]);  
 buttonPanel.add(mulButton);  
 buttonPanel.add(numberButtons[7]);  
 buttonPanel.add(numberButtons[8]);  
 buttonPanel.add(numberButtons[9]);  
 buttonPanel.add(divButton);  
 buttonPanel.add(numberButtons[0]);  
 buttonPanel.add(clrButton);  
  
 // logic  
 final double[] num1 = new double[1];  
 final char[] operator = new char[1];  
  
 addButton.addActionListener(e -> {  
 num1[0] = Double.*parseDouble*(textField.getText());  
 operator[0] = '+';  
 textField.setText(textField.getText() + "+");  
 });  
 subButton.addActionListener(e -> {  
 num1[0] = Double.*parseDouble*(textField.getText());  
 operator[0] = '-';  
 textField.setText(textField.getText() + "-");  
 });  
 mulButton.addActionListener(e -> {  
 num1[0] = Double.*parseDouble*(textField.getText());  
 operator[0] = '\*';  
 textField.setText(textField.getText() + "\*");  
 });  
 divButton.addActionListener(e -> {  
 num1[0] = Double.*parseDouble*(textField.getText());  
 operator[0] = '/';  
 textField.setText(textField.getText() + "/");  
 });  
  
 clrButton.addActionListener(e -> textField.setText(""));  
  
 equButton.addActionListener(e -> {  
 try {  
 String text = textField.getText();  
 String[] parts = text.split("[+\\-\*/]");  
 if (parts.length < 2) return;  
 double num2 = Double.*parseDouble*(parts[1]);  
 double result = switch (operator[0]) {  
 case '+' -> num1[0] + num2;  
 case '-' -> num1[0] - num2;  
 case '\*' -> num1[0] \* num2;  
 case '/' -> num1[0] / num2;  
 default -> 0;  
 };  
 textField.setText(text + "=" + result);  
 } catch (Exception ex) {  
 textField.setText("Error");  
 }  
 });  
  
 frame.setVisible(true);  
 }  
}

**Output:**

****

**Module 5**

****

**Q1 Java Servlet Web Application(Feedback App)**

**Code:**

**Servlet Code(FeedbackServlet class)**

package com.example;

import java.io.IOException;

import java.io.PrintWriter;

import java.time.LocalDateTime;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class FeedbackServlet extends HttpServlet {

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String name = request.getParameter("name");

String email = request.getParameter("email");

String type = request.getParameter("type");

String comments = request.getParameter("comments");

LocalDateTime now = LocalDateTime.now();

out.println("<html><body>");

out.println("<h2>Thank you, " + name + " for your feedback!</h2>");

out.println("<p><strong>Email:</strong> " + email + "</p>");

out.println("<p><strong>Feedback Type:</strong> " + type + "</p>");

out.println("<p><strong>Comments:</strong> " + comments + "</p>");

out.println("<p><strong>Submitted On:</strong> " + now + "</p>");

out.println("</body></html>");

}

}

**HTML File**

<!DOCTYPE html>

<html>

<head>

<title>Feedback Form</title>

</head>

<body>

<h2>Feedback Form</h2>

<form action="submitFeedback" method="post">

<label>Name:</label><br>

<input type="text" name="name" required><br><br>

<label>Email:</label><br>

<input type="email" name="email" required><br><br>

<label>Feedback Type:</label><br>

<select name="type" required>

<option value="Complaint">Complaint</option>

<option value="Suggestion">Suggestion</option>

<option value="Compliment">Compliment</option>

</select><br><br>

<label>Comments:</label><br>

<textarea name="comments" rows="5" cols="40" required></textarea><br><br>

<input type="submit" value="Submit Feedback">

</form>

</body>

</html>

**XML File**

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

version="3.1">

<servlet>

<servlet-name>FeedbackServlet</servlet-name>

<servlet-class>com.example.FeedbackServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>FeedbackServlet</servlet-name>

<url-pattern>/submitFeedback</url-pattern>

</servlet

**File Structure in Tomcat**

**C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\FeedbackApp\**

**│**

**├── feedback.html**

**│**

**└── WEB-INF/**

**├── web.xml**

**└── classes/**

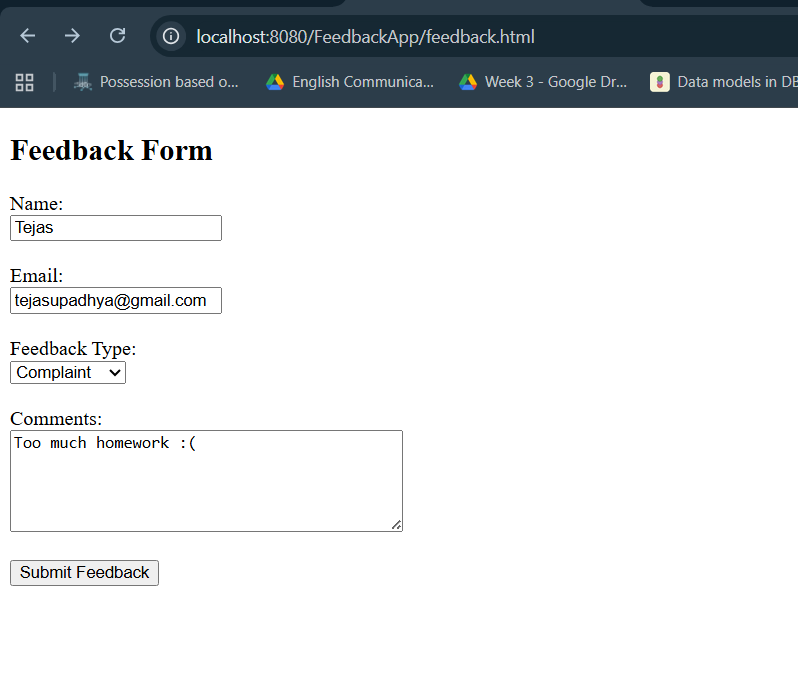
**└── com/**

**└── example/**

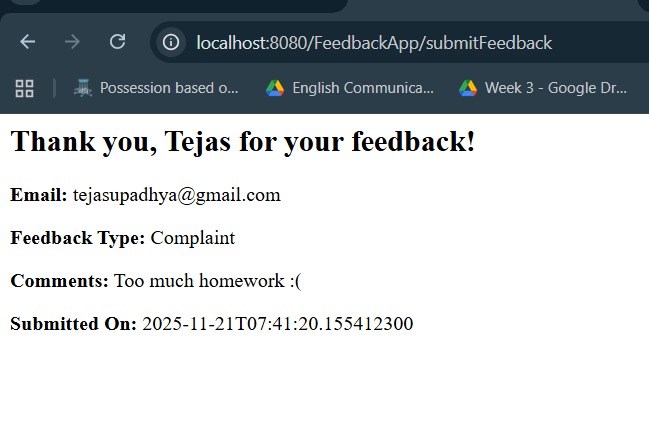
**└── FeedbackServlet.class**

**Output:**

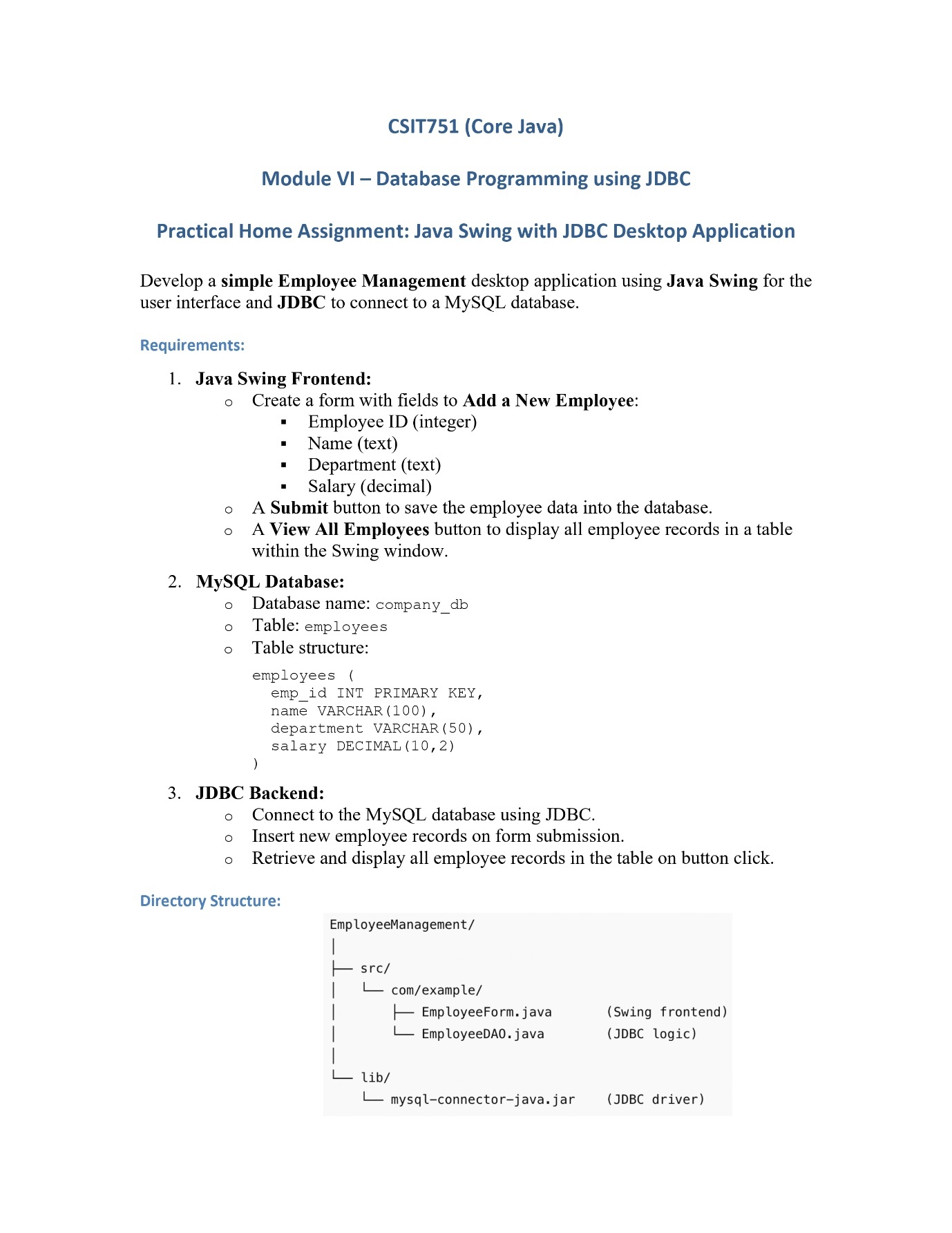
**HTML Form**



**Servlet Response**

****

**Module 6**

****

**Q1 Java Swing with JDBC Desktop Application**

**Code:**

**Swing Interface(EmployeeForm)**

package Module6;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.util.ArrayList;  
  
public class EmployeeForm extends JFrame {  
  
 private JTextField idField, nameField, deptField, salaryField;  
 private JTable table;  
 private DefaultTableModel tableModel;  
  
 public EmployeeForm() {  
 setTitle("Employee Management System");  
 setSize(700, 500);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(new BorderLayout());  
  
 // ======= FORM =======  
 JPanel form = new JPanel(new GridLayout(5, 2, 10, 10));  
  
 form.add(new JLabel("Employee ID:"));  
 idField = new JTextField();  
 form.add(idField);  
  
 form.add(new JLabel("Name:"));  
 nameField = new JTextField();  
 form.add(nameField);  
  
 form.add(new JLabel("Department:"));  
 deptField = new JTextField();  
 form.add(deptField);  
  
 form.add(new JLabel("Salary:"));  
 salaryField = new JTextField();  
 form.add(salaryField);  
  
 JButton addBtn = new JButton("Add Employee");  
 JButton viewBtn = new JButton("View All Employees");  
  
 form.add(addBtn);  
 form.add(viewBtn);  
  
 add(form, BorderLayout.*NORTH*);  
  
 // ======= TABLE =======  
 tableModel = new DefaultTableModel(new String[]{"ID", "Name", "Department", "Salary"}, 0);  
 table = new JTable(tableModel);  
 add(new JScrollPane(table), BorderLayout.*CENTER*);  
  
 // ======= BUTTON ACTIONS =======  
 addBtn.addActionListener(e -> addEmployee());  
 viewBtn.addActionListener(e -> loadEmployees());  
 }  
  
 private void addEmployee() {  
 try {  
 int id = Integer.*parseInt*(idField.getText());  
 String name = nameField.getText();  
 String dept = deptField.getText();  
 double salary = Double.*parseDouble*(salaryField.getText());  
  
 EmployeeDAO.*insertEmployee*(id, name, dept, salary);  
 JOptionPane.*showMessageDialog*(this, "Employee added!");  
  
 } catch (Exception ex) {  
 JOptionPane.*showMessageDialog*(this, "Error: " + ex.getMessage());  
 }  
 }  
  
 private void loadEmployees() {  
 try {  
 ArrayList<Object[]> employees = EmployeeDAO.*getAllEmployees*();  
 tableModel.setRowCount(0);  
  
 for (Object[] row : employees) {  
 tableModel.addRow(row);  
 }  
  
 } catch (Exception ex) {  
 JOptionPane.*showMessageDialog*(this, "Error: " + ex.getMessage());  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(() -> new EmployeeForm().setVisible(true));  
 }  
}

**JDBC Connection and Logic(EmployeeDAO)**

package Module6;  
  
import java.sql.\*;  
import java.util.ArrayList;  
  
public class EmployeeDAO {  
  
 private static final String *URL* = "jdbc:mysql://localhost:3306/company\_db";  
 private static final String *USER* = "root";  
 private static final String *PASSWORD* = "12345";  
  
 // Insert employee  
 public static void insertEmployee(int id, String name, String dept, double salary) throws Exception {  
 Connection con = DriverManager.*getConnection*(*URL*, *USER*, *PASSWORD*);  
  
 String query = "INSERT INTO employees(emp\_id, name, department, salary) VALUES(?, ?, ?, ?)";  
 PreparedStatement pst = con.prepareStatement(query);  
  
 pst.setInt(1, id);  
 pst.setString(2, name);  
 pst.setString(3, dept);  
 pst.setDouble(4, salary);  
  
 pst.executeUpdate();  
  
 pst.close();  
 con.close();  
 }  
  
 // Fetch all employees  
 public static ArrayList<Object[]> getAllEmployees() throws Exception {  
 ArrayList<Object[]> data = new ArrayList<>();  
  
 Connection con = DriverManager.*getConnection*(*URL*, *USER*, *PASSWORD*);  
 String query = "SELECT \* FROM employees";  
 Statement st = con.createStatement();  
 ResultSet rs = st.executeQuery(query);  
  
 while (rs.next()) {  
 Object[] row = new Object[4];  
 row[0] = rs.getInt("emp\_id");  
 row[1] = rs.getString("name");  
 row[2] = rs.getString("department");  
 row[3] = rs.getDouble("salary");  
 data.add(row);  
 }  
  
 rs.close();  
 st.close();  
 con.close();  
  
 return data;  
 }  
}

**MYSQL Code:**

CREATE DATABASE company\_db;

USE company\_db;

CREATE TABLE employees (

emp\_id INT PRIMARY KEY,

name VARCHAR(100),

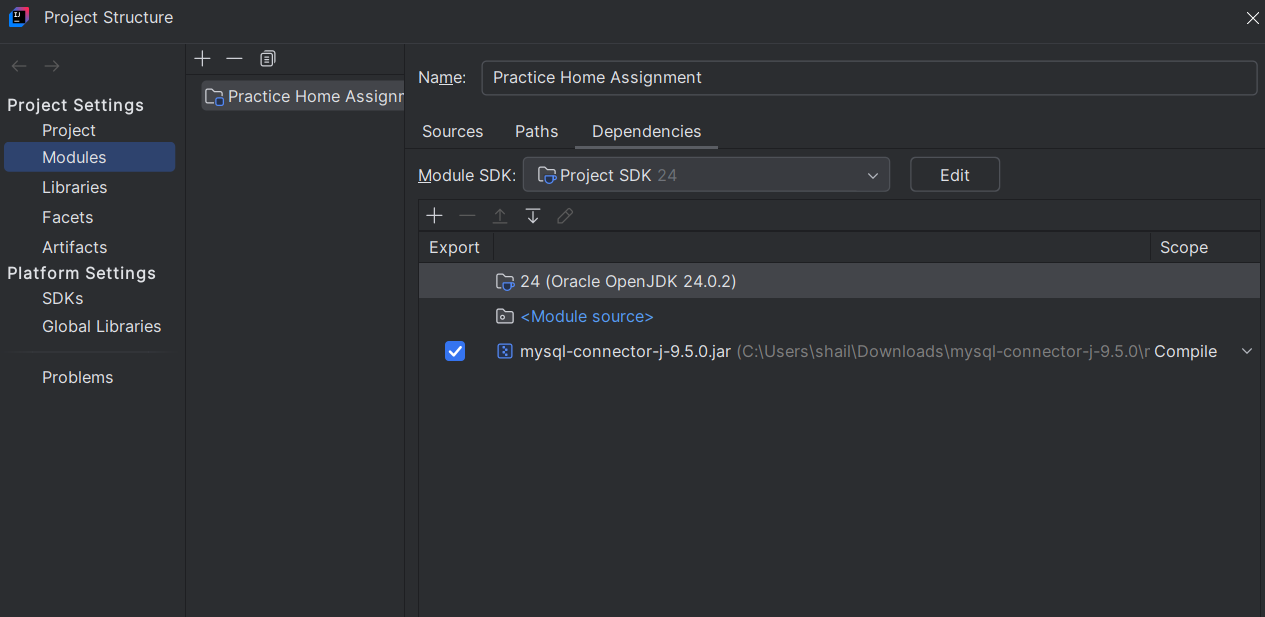
department VARCHAR(50),

salary DECIMAL(10,2)

);

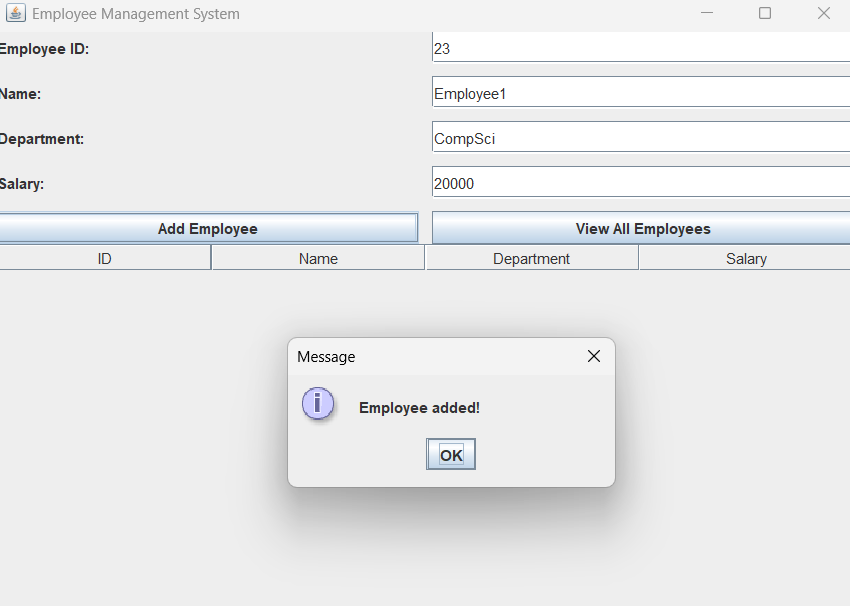
select \* from employees;

**MySQL connector**

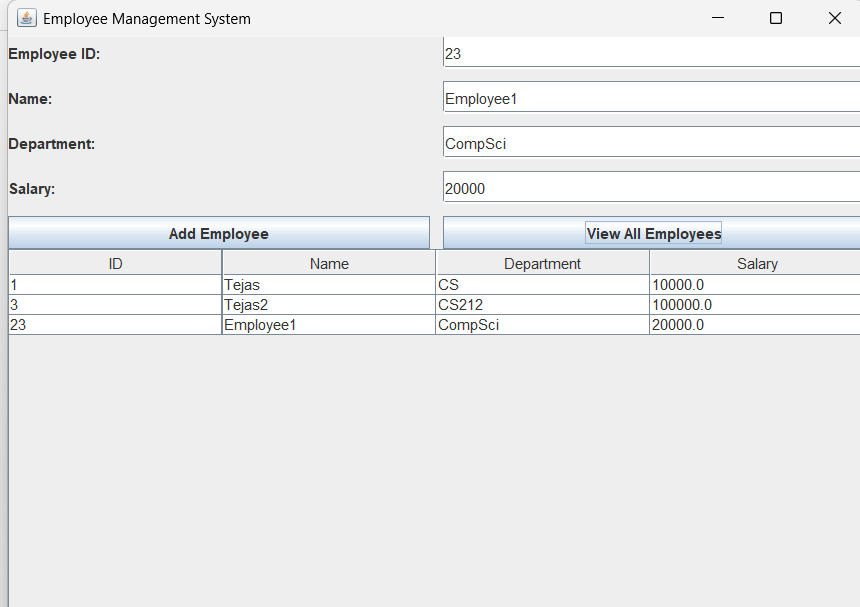


**Output:**

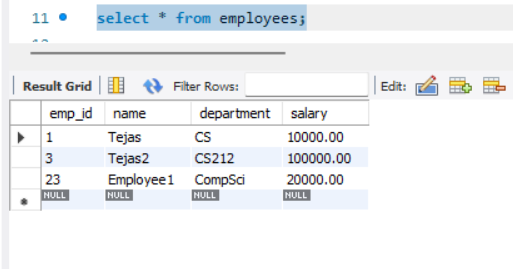
**Adding an Employee**

****

**Viewing all Employees**

****

**SQL Table**

****