1.

a)

public class OOPExercises {

public static void main(String[] args) {

A objA = new A();

System.out.println("objA.a = "+objA.a); // Error here

objA.a = 222;

}

}

Error is Direct access to private variable 'a'. Using getter method objA.getA() can prevent that

b)

public class OOPExercises {

public static void main(String[] args) {

System.out.println("objA.a = "+getA() ); // Error here

setA(123);

}

}

Error is Trying to call instance methods statically and it can fixed using Create instance of class A first

A objA = new A();

System.out.println("objA.a = " + objA.getA());

objA.setA(123);

c)

double result = 55;

objA.setA(result);

Error is loss of precision converting double to int and it can Fix using Cast the double value or use int

int result = 55;

d)

final int a = 100;

public void setA(int value) {

a = value; // Error here

}

Error is Attempting to modify final variable and we can fix it Remove final keyword or don't modify the variable

2.

public class **Point** {

private int x, y;

public Point(int a, int b) {

x = a;

y = b;

}

public String displayPoint() {

return "(" + x + ", " + y + ")";

}

public double distance() {

return Math.sqrt(x \* x + y \* y);

}

public double distance(Point other) {

int dx = x - other.x;

int dy = y - other.y;

return Math.sqrt(dx \* dx + dy \* dy);

}

}

public class **PointClient** {

public static void main(String[] args) {

Point p1 = new Point(20, 2);

Point p2 = new Point(3, 6);

System.out.println("Point 1: " + p1.displayPoint());

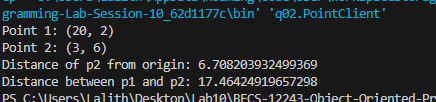
System.out.println("Point 2: " + p2.displayPoint());

System.out.println("Distance of p2 from origin: " + p2.distance());

System.out.println("Distance between p1 and p2: " + p2.distance(p1));

}

}



3.

public class **Employee** {

private String name;

private int idNumber;

private double payRate;

private static int employeeCounter = 0;

public Employee(String name, double payRate) {

this.name = name;

this.payRate = payRate;

employeeCounter++;

this.idNumber = employeeCounter + 1000;

}

public double calcPay(double hours) {

return hours \* payRate;

}

public static int numberOfEmployees() {

return employeeCounter;

}

public String getName() { return name; }

public int getIdNumber() { return idNumber; }

public double getPayRate() { return payRate; }

}

public class **EmployeeTest** {

public static void main(String[] args) {

Employee emp1 = new Employee("Saman", 1000.0);

Employee emp2 = new Employee("Amal", 2500.0);

Employee emp3 = new Employee("Kamala", 1250.0);

System.out.println("Name: " + emp1.getName() + ", ID: " + emp1.getIdNumber() + ", Pay Rate: Rs." + emp1.getPayRate());

System.out.println("Name: " + emp2.getName() + ", ID: " + emp2.getIdNumber() + ", Pay Rate: Rs." + emp2.getPayRate());

System.out.println("Name: " + emp3.getName() + ", ID: " + emp3.getIdNumber() + ", Pay Rate: Rs." + emp3.getPayRate());

System.out.println(emp1.getName() + "'s Pay for 40 hours: Rs." + emp1.calcPay(40));

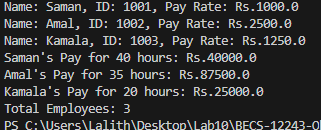
System.out.println(emp2.getName() + "'s Pay for 35 hours: Rs." + emp2.calcPay(35));

System.out.println(emp3.getName() + "'s Pay for 20 hours: Rs." + emp3.calcPay(20));

System.out.println("Total Employees: " + Employee.numberOfEmployees());

}

}



4.

public class **Movie** {

private String title;

private int duration;

public Movie(String title, int duration) {

this.title = title;

this.duration = duration;

}

public String getTitle() { return title; }

public int getDuration() { return duration; }

}

public class **Ticket** {

private Movie movie;

private String seatNumber;

private double price;

public Ticket(Movie movie, String seatNumber, double price) {

this.movie = movie;

this.seatNumber = seatNumber;

this.price = price;

}

public String toString() {

return "Movie: " + movie.getTitle() + ", Seat: " + seatNumber + ", Price: Rs." + price;

}

}

public class **Cinema** {

private Ticket[] tickets;

private int ticketCount;

public Cinema(int capacity) {

tickets = new Ticket[capacity];

ticketCount = 0;

}

public void bookTicket(Ticket ticket) {

if (ticketCount < tickets.length) {

tickets[ticketCount++] = ticket;

System.out.println("Ticket booked successfully");

} else {

System.out.println("Cinema is full");

}

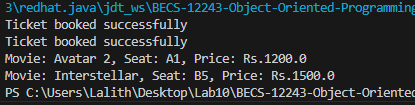
}

public void displayAllTickets() {

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

System.out.println(tickets[i]);

}

 }

}

5.

public class **Patient** {

private String patientID;

private String name;

private String disease;

public Patient(String patientID, String name, String disease) {

this.patientID = patientID;

this.name = name;

this.disease = disease;

}

public void displayDetails() {

System.out.println("Patient ID: " + patientID + ", Name: " + name + ", Disease: " + disease);

}

public String getName() { return name; }

public String getDisease() { return disease; }

}

public class **Doctor** {

private String doctorID;

private String name;

private String specialization;

public Doctor(String doctorID, String name, String specialization) {

this.doctorID = doctorID;

this.name = name;

this.specialization = specialization;

}

public void treatPatient(Patient patient) {

System.out.println("Dr. " + name + " is treating " + patient.getName() + " for " + patient.getDisease());

}

}

public class **HospitalSystem** {

public static void main(String[] args) {

Patient p1 = new Patient("P001", "John", "Fever");

Patient p2 = new Patient("P002", "Mary", "Flu");

p1.displayDetails();

p2.displayDetails();

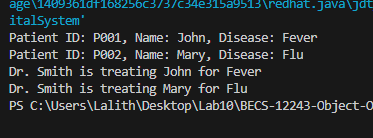
Doctor doc = new Doctor("D001", "Smith", "General Medicine");

doc.treatPatient(p1);

doc.treatPatient(p2);

}

}



|  |
| --- |
| 6. Restaurant Management System |

public class **Dish** {

private String name;

private double price;

public Dish(String name, double price) {

this.name = name;

this.price = price;

}

public void displayDetails() {

System.out.println("Dish: " + name + ", Price: Rs." + price);

}

public double getPrice() { return price; }

public String getName() { return name; }

}

public class **Order** {

private Dish dish;

private int quantity;

public Order(Dish dish, int quantity) {

this.dish = dish;

this.quantity = quantity;

}

public double calculateTotal() {

return dish.getPrice() \* quantity;

}

public void displayOrder() {

System.out.println(quantity + "x " + dish.getName() + " - Rs." + calculateTotal());

}

}

public class **Restaurant** {

private Order[] orders;

private int orderCount;

public Restaurant(int capacity) {

orders = new Order[capacity];

orderCount = 0;

}

public void addOrder(Order order) {

if (orderCount < orders.length) {

orders[orderCount++] = order;

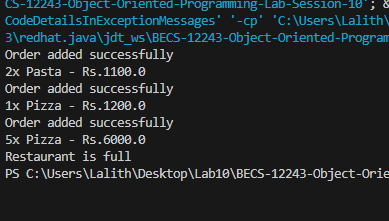
System.out.println("Order added successfully");

order.displayOrder();

} else {

System.out.println("Restaurant is full");

}

 }

}