

Name : Sanket Banate
Roll No : 64
PRN : 202301040044
Batch : A4

PART 1 (1):

Write a Java program to create a class known as Person with methods called getFirstName() and getLastName(). Create a subclass called Employee that adds a new method named getEmployeeId() and overrides the getLastName() method to include the employee's job title.

```
class Person {
    private String firstName;
    private String lastName;

    // Constructor
    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public static void main(String[] args) {
        // Create a Person
        Person person = new Person("Ravi", "Kumar");

        // Display person's information
        System.out.println("Person:");
        System.out.println("First Name: " + person.getFirstName());
        System.out.println("Last Name: " + person.getLastName());
        System.out.println();

        // Create an Employee
        Employee employee = new Employee("Sanket", "Banate", 2022010, "Software Engineer");

        // Display employee's information
        System.out.println("Employee:");
        System.out.println("First Name: " + employee.getFirstName());
        System.out.println("Last Name: " + employee.getLastName());
        System.out.println("Employee ID: " + employee.getEmployeeId());
    }
}

// Employee class (subclass of Person)
class Employee extends Person {
    private int employeeId;
    private String jobTitle;

    // Constructor
    public Employee(String firstName, String lastName, int employeeId, String jobTitle) {
        super(firstName, lastName);
    }
}
```

```

        this.employeeId = employeeId;
        this.jobTitle = jobTitle;
    }

    // New method for Employee class
    public int getEmployeeId() {
        return employeeId;
    }

    // Override getLastName() method to include job title Override
    public String getLastName() {
        // Include job title in the last name for employees
        return super.getLastName() + " (" + jobTitle + ")";
    }
}

```

OUTPUT:

```

PROBLEMS 1 PORTS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 1 (1)/" && javac Person.java && java Person
sanketbanate@Sankets-MacBook-Air ~ % cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 1 (1)/" && javac Person.java && java Person
Person:
First Name: Ravi
Last Name: Kumar

Employee:
First Name: Sanket
Last Name: Banate (Software Engineer)
Employee ID: 2022010
sanketbanate@Sankets-MacBook-Air Part 1 (1) %

Ln 63, Col 2 Spaces: 4 UTF-8 CRLF {} Java Go Live Prettier

```

PART 1 (2):

Create a class named 'Member' having the following members:

Data members

1 - Name

2 - Age

3 - Phone number

4 - Address

5 - Salary

It also has a method named 'printSalary' which prints the salary of the members.

Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

```
class Member {
    // Data members
    protected String name;
    protected int age;
    protected String phoneNumber;
    protected String address;
    protected double salary;

    // Constructor
    public Member(String name, int age, String phoneNumber, String address, double salary) {
        this.name = name;
        this.age = age;
        this.phoneNumber = phoneNumber;
        this.address = address;
        this.salary = salary;
    }

    // Method to print salary
    public void printSalary() {
        System.out.println("Salary: " + salary);
    }

    public static void main(String[] args) {
        // Create an Employee
        Employee employee = new Employee("Sanket Banate", 30, "123-456-7890", "123 Main St",
        50000, "Software Engineer");

        // Print employee information
        System.out.println("Employee Information:");
        System.out.println("Name: " + employee.name);
        System.out.println("Age: " + employee.age);
        System.out.println("Phone Number: " + employee.phoneNumber);
        System.out.println("Address: " + employee.address);
        System.out.println("Specialization: " + employee.getSpecialization());
        employee.printSalary();
        System.out.println();

        // Create a Manager
        Manager manager = new Manager("Mukesh Naik", 35, "987-654-3210", "456 Oak St",
        80000, "IT");

        // Print manager information
```

```

        System.out.println("Manager Information:");
        System.out.println("Name: " + manager.name);
        System.out.println("Age: " + manager.age);
        System.out.println("Phone Number: " + manager.phoneNumber);
        System.out.println("Address: " + manager.address);
        System.out.println("Department: " + manager.getDepartment());
        manager.printSalary();
    }
}

class Employee extends Member {
    private String specialization;

    // Constructor
    public Employee(String name, int age, String phoneNumber, String address, double salary,
String specialization) {
        super(name, age, phoneNumber, address, salary);
        this.specialization = specialization;
    }

    public String getSpecialization() {
        return specialization;
    }
}

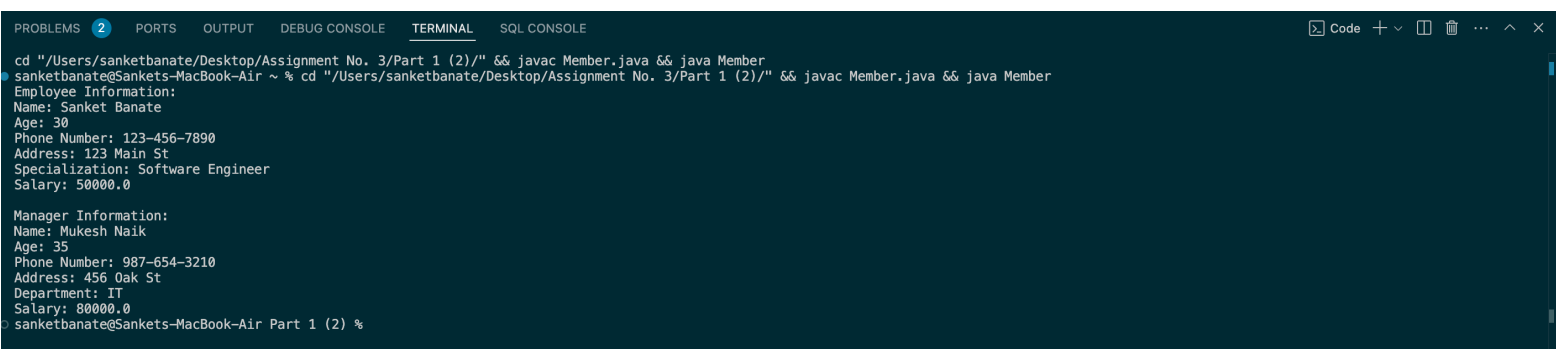
class Manager extends Member {
    private String department;

    // Constructor
    public Manager(String name, int age, String phoneNumber, String address, double salary,
String department) {
        super(name, age, phoneNumber, address, salary);
        this.department = department;
    }

    public String getDepartment() {
        return department;
    }
}

```

OUTPUT:



```

PROBLEMS 2 PORTS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 1 (2)/" && javac Member.java && java Member
sanketbanate@Sankets-MacBook-Air ~ % cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 1 (2)/" && javac Member.java && java Member
Employee Information:
Name: Sanket Banate
Age: 30
Phone Number: 123-456-7890
Address: 123 Main St
Specialization: Software Engineer
Salary: 50000.0

Manager Information:
Name: Mukesh Naik
Age: 35
Phone Number: 987-654-3210
Address: 456 Oak St
Department: IT
Salary: 80000.0
sanketbanate@Sankets-MacBook-Air Part 1 (2) %

```

PART 2 (2):

Write a Java program to create a class called Shape with a method called getArea(). Create a subclass called Rectangle that overrides the getArea() method to calculate the area of a rectangle.

```
class Shape {  
  
    public double getArea() {  
        return 0.0;  
    }  
    public static void main(String[] args) {  
  
        Rectangle rectangle = new Rectangle(5.0, 10.0);  
  
        System.out.println("Area of the Rectangle: " + rectangle.getArea());  
    }  
}  
  
class Rectangle extends Shape {  
    private double length;  
    private double width;  
  
    public Rectangle(double length, double width) {  
        this.length = length;  
        this.width = width;  
    }  
  
    public double getArea() {  
        return length * width;  
    }  
}
```

OUTPUT:

```
cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 2 (2)/" && javac Shape.java && java Shape  
sanketbanate@Sankets-MacBook-Air ~ % cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 2 (2)/" && javac Shape.java && java Shape  
Area of the Rectangle: 50.0  
sanketbanate@Sankets-MacBook-Air Part 2 (2) %  
  
Ln 30, Col 2  Spaces: 4  UTF-8  CRLF  {} Java  Go Live
```

PART 3 (2):

Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.

```
class Vehicle {

    protected String make;
    protected String model;
    protected int year;
    protected String fuelType;

    public Vehicle(String make, String model, int year, String fuelType) {
        this.make = make;
        this.model = model;
        this.year = year;
        this.fuelType = fuelType;
    }

    public double calculateFuelEfficiency() {
        return 0.0;
    }

    public double calculateDistance(double fuelEfficiency, double fuelAmount) {
        return fuelEfficiency * fuelAmount;
    }

    public int getMaxSpeed() {
        return 0;
    }

    public static void main(String[] args) {

        Truck myTruck = new Truck("Ford", "F-150", 2022, "Gasoline", 1500.0);
        Car myCar = new Car("Toyota", "Camry", 2022, "Gasoline", false);
        Motorcycle myMotorcycle = new Motorcycle("Harley-Davidson", "Sportster", 2022,
"Gasoline", true);

        displayVehicleInformation(myTruck);
        displayVehicleInformation(myCar);
        displayVehicleInformation(myMotorcycle);
    }
}
```

```

private static void displayVehicleInformation(Vehicle vehicle) {
    System.out.println("Vehicle Information:");
    System.out.println("Make: " + vehicle.make);
    System.out.println("Model: " + vehicle.model);
    System.out.println("Year: " + vehicle.year);
    System.out.println("Fuel Type: " + vehicle.fuelType);

    double fuelEfficiency = vehicle.calculateFuelEfficiency();
    System.out.println("Fuel Efficiency: " + fuelEfficiency + " miles per gallon");

    double distanceTraveled = vehicle.calculateDistance(fuelEfficiency, 20.0); // Assuming 20
gallons of fuel
    System.out.println("Distance Traveled: " + distanceTraveled + " miles");

    int maxSpeed = vehicle.getMaxSpeed();
    System.out.println("Maximum Speed: " + maxSpeed + " miles per hour");

    System.out.println();
}
}

// Subclass: Truck
class Truck extends Vehicle {
    private double cargoCapacity;

    // Constructor
    public Truck(String make, String model, int year, String fuelType, double cargoCapacity) {
        super(make, model, year, fuelType);
        this.cargoCapacity = cargoCapacity;
    }

    public double calculateFuelEfficiency() {

        return 10.0;
    }

    public int getMaxSpeed() {

        return 70;
    }
}

class Car extends Vehicle {
    private boolean isConvertible;

    public Car(String make, String model, int year, String fuelType, boolean isConvertible) {
        super(make, model, year, fuelType);
        this.isConvertible = isConvertible;
    }

    public double calculateFuelEfficiency() {

        return 25.0;
    }
}

```

```

    }

    public int getMaxSpeed() {

        return 120;
    }
}

class Motorcycle extends Vehicle {
    private boolean hasFairing;

    public Motorcycle(String make, String model, int year, String fuelType, boolean hasFairing) {
        super(make, model, year, fuelType);
        this.hasFairing = hasFairing;
    }

    public double calculateFuelEfficiency() {

        return 50.0;
    }

    public int getMaxSpeed() {

        return 150;
    }
}

```

OUTPUT:

```

cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 3 (2)/" && javac Vehicle.java && java Vehicle
sanketbanate@Sankets-MacBook-Air ~ % cd "/Users/sanketbanate/Desktop/Assignment No. 3/Part 3 (2)/" && javac Vehicle.java && java Vehicle
Vehicle Information:
Make: Ford
Model: F-150
Year: 2022
Fuel Type: Gasoline
Fuel Efficiency: 10.0 miles per gallon
Distance Traveled: 200.0 miles
Maximum Speed: 70 miles per hour

Vehicle Information:
Make: Toyota
Model: Camry
Year: 2022
Fuel Type: Gasoline
Fuel Efficiency: 25.0 miles per gallon
Distance Traveled: 500.0 miles
Maximum Speed: 120 miles per hour

Vehicle Information:
Make: Harley-Davidson
Model: Sportster
Year: 2022
Fuel Type: Gasoline
Fuel Efficiency: 50.0 miles per gallon
Distance Traveled: 1000.0 miles
Maximum Speed: 150 miles per hour
sanketbanate@Sankets-MacBook-Air Part 3 (2) %

```