

# JAVA

## Task 3 :

Write a Java program that demonstrates various OOP concepts including class design, inheritance, abstraction, polymorphism, encapsulation, method overriding, and method overloading.

## CODE :

```
// Abstract class
abstract class Employee {
    private String firstName;
    private String lastName;
    private double salary;

    public Employee(String firstName, String lastName, double salary) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.salary = salary;
    }

    // Abstract method
    public abstract void work();

    // Encapsulation
    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public double getSalary() {
        return salary;
    }
}
```

```

    public void setSalary(double salary) {
        this.salary = salary;
    }

    //Method overriding
    @Override
    public String toString() {
        return "Employee: " + firstName + " " + lastName + ", Salary: " +
salary;
    }
}

//inheritance
class Developer extends Employee {
    private String programmingLanguage;

    public Developer(String firstName, String lastName, double salary,
String programmingLanguage) {
        super(firstName, lastName, salary);
        this.programmingLanguage = programmingLanguage;
    }

    // Method Overriding
    @Override
    public void work() {
        System.out.println(getFirstName() + " is coding in " +
programmingLanguage);
    }

    // Method Overloading
    public void work(String projectName) {
        System.out.println(getFirstName() + " is coding in " +
programmingLanguage + " on project " + projectName);
    }
}

public class Main {
    public static void main(String[] args) {
        Developer dev1 = new Developer("Alice", "Thomas", 80000, "Java");
    }
}

```

```

        //encapsulation
        dev1.setSalary(90000);
        System.out.println(dev1.getFirstName() + " " + dev1.getLastName());

        //Method overriding
        System.out.println(dev1);
        dev1.work();

        // Method overloading
        dev1.work("Project X");

        // Polymorphism
        Employee emp1 = new Developer("Charlie", "Brown", 90000, "Python");
        emp1.work();
    }
}

```

OUTPUT :

Alice Thomas

Employee: Alice Thomas, Salary: 90000.0

Alice is coding in Java

Alice is coding in Java on project Project X

Charlie is coding in Python