

**Roll NO:57**

Q.1 Write a Java program to create a class known as "BankAccount" with methods called deposit() and withdraw(). Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

```
Program: class BankAccount {
    protected double balance;
    public BankAccount(double initialBalance) {
        balance = initialBalance;
    }
    public void deposit(double amount) {
        balance = balance + amount;
        System.out.println("Deposited: " + amount);
    }
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance = balance - amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance!");
        }
    }
    public void displayBalance() {
        System.out.println("Current Balance: " + balance);
    }
}
class SavingsAccount extends BankAccount {
    public SavingsAccount(double initialBalance) {
        super(initialBalance);
    }
    public void withdraw(double amount) {
        if (balance - amount < 100) {
            System.out.println("Withdrawal denied! Minimum balance of $100 required.");
        } else {
            balance = balance - amount;
            System.out.println("Withdrawn: " + amount);
        }
    }
}
```

**Output:**

```
Current Balance: 500.0
Deposited: 200.0
Withdrawal denied! Minimum balance of $100
required.
Withdrawn: 50.0
Current Balance: 650.0
```

```
public class Bank {
    public static void main(String[] args) {
        SavingsAccount myAccount = new SavingsAccount(500);
        myAccount.displayBalance();
        myAccount.deposit(200);
        myAccount.withdraw(550);
        myAccount.withdraw(50);
        myAccount.displayBalance();
    }
}
```

Q. 2. Write a Java program that creates a class hierarchy for employees of a company. The base class should be Employee, with subclasses Manager, Developer, and Programmer. Each subclass should have properties such as name, address, salary, and job title. Implement methods for calculating bonuses, generating performance reports, and managing projects

Program:

```
import java.util.Scanner;
```

```
abstract class Employee {
    protected String name;
    protected String address;
    protected double salary;
    protected String jobTitle;

    public Employee(String name, String address, double salary, String jobTitle) {
        name = name;
        address = address;
```

```

        salary = salary;
        jobTitle = jobTitle;
    }

    public abstract double calculateBonus();
    public abstract String generatePerformanceReport();
}

class Manager extends Employee {
    private int managedProjects;

    public Manager(String name, String address, double salary,
int managedProjects) {
        super(name, address, salary, "Manager");
        managedProjects = managedProjects;
    }

    public double calculateBonus() {
        return salary * 0.2 + managedProjects * 500;
    }

    public String generatePerformanceReport() {
        return name + " is managing " + managedProjects + " projects with high efficiency.";
    }
}

class Developer extends Employee {
    private int completedTasks;

    public Developer(String name, String address, double salary, int completedTasks) {
        super(name, address, salary, "Developer");
        completedTasks = completedTasks;
    }

    public double calculateBonus() {
        return salary * 0.15 + completedTasks * 200;
    }

    public String generatePerformanceReport() {
        return name + " has successfully completed " + completedTasks + " tasks this quarter.";
    }
}

class Programmer extends Employee {
    private int codeLines;

    public Programmer(String name, String address, double salary, int codeLines) {
        super(name, address, salary, "Programmer");
        codeLines = codeLines;
    }

    public double calculateBonus() {
        return salary * 0.1 + codeLines * 0.1;
    }

    public String generatePerformanceReport() {
        return name + " has written " + codeLines + " lines of code with great efficiency.";
    }
}

public class Company {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

```

#### input:

Enter Manager Name:  
 snehal  
 Enter Manager Address:  
 vitha  
 Enter Manager Salary:  
 80000  
 Enter Number of  
 Managed Projects: 5  
  
 Enter Developer Name:  
 nilam  
 Enter Developer Address:  
 vadgaon  
 Enter Developer Salary:  
 70000  
 Enter Number of  
 Completed Tasks: 20  
  
 Enter Programmer  
 Name: Shruti  
 Enter Programmer  
 Address: uchgaon  
 Enter Programmer  
 Salary: 60000  
 Enter Number of Lines of  
 Code Written: 10000

#### Output:

Snehal is managing 5  
 projects with high  
 efficiency.  
 Bonus: 90000.0  
  
 Nilam has successfully  
 completed 20 tasks this  
 quarter.  
 Bonus: 74000.0  
  
 shruti has written 10000  
 lines of code with great  
 efficiency.  
 Bonus: 61000.0

```

System.out.print("Enter Manager Name: ");
String managerName = scanner.nextLine();
System.out.print("Enter Manager Address: ");
String managerAddress = scanner.nextLine();
System.out.print("Enter Manager Salary: ");
double managerSalary = scanner.nextDouble();
System.out.print("Enter Number of Managed Projects: ");
int managedProjects = scanner.nextInt();
scanner.nextLine();
Manager manager = new Manager(managerName, managerAddress, managerSalary, managedProjects);

System.out.print("Enter Developer Name: ");
String devName = scanner.nextLine();
System.out.print("Enter Developer Address: ");
String devAddress = scanner.nextLine();
System.out.print("Enter Developer Salary: ");
double devSalary = scanner.nextDouble();
System.out.print("Enter Number of Completed Tasks: ");
int completedTasks = scanner.nextInt();
scanner.nextLine();
Developer developer = new Developer(devName, devAddress, devSalary, completedTasks);

System.out.print("Enter Programmer Name: ");
String progName = scanner.nextLine();
System.out.print("Enter Programmer Address: ");
String progAddress = scanner.nextLine();
System.out.print("Enter Programmer Salary: ");
double progSalary = scanner.nextDouble();
System.out.print("Enter Number of Lines of Code Written: ");
int codeLines = scanner.nextInt();
Programmer programmer = new Programmer(progName, progAddress, progSalary, codeLines);

System.out.println(manager.generatePerformanceReport());
System.out.println("Bonus: " + manager.calculateBonus());

System.out.println(developer.generatePerformanceReport());
System.out.println("Bonus: " + developer.calculateBonus());

System.out.println(programmer.generatePerformanceReport());
System.out.println("Bonus: " + programmer.calculateBonus());
}
}

```

Q.3 Implement Following: a. Create abstract class shapes with dim1, dim2 variables and abstract area() method. Class b. rectangle and triangle inherits shape class. Calculate area of rectangle and triangle.  
Program:

```

import java.util.Scanner;

abstract class Shape {
    double dim1, dim2;
    Shape(double d1, double d2) {
        dim1 = d1;
        dim2 = d2;
    }
    abstract double area();
}

class Rectangle extends Shape {
    Rectangle(double d1, double d2) {
        super(d1, d2);
    }

    double area() {
        return dim1 * dim2;
    }
}

class Triangle extends Shape {
    Triangle(double d1, double d2) {
        super(d1, d2);
    }
}

```

**Input:**

Enter length and breadth of rectangle:38 22  
Enter base and height of triangle:10 22

**Output:**

Area of rectangle is: 748.0  
Area of triangle is: 40.0

```

    }
    double area(){
        return 0.5 * dim1 * dim2;
    }
}

public class ShapeDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter length and breadth of rectangle: ");
        double l = scanner.nextDouble();
        double b = scanner.nextDouble();
        Rectangle rect = new Rectangle(l, b);
        System.out.println("Area of Rectangle: " + rect.area());

        System.out.print("Enter base and height of triangle: ");
        double base = scanner.nextDouble();
        double height = scanner.nextDouble();
        Triangle tri = new Triangle(base, height);
        System.out.println("Area of Triangle: " + tri.area());

        scanner.close();
    }
}

```

**Q.4 Write a program to perform Multilevel Inheritance**

**Program:**

```

import java.util.*;
class Student{
    int roll_no;
    Scanner sc=new Scanner(System.in);
    Student(){
        System.out.println("Enter roll no.");
        roll_no=sc.nextInt();
    }
}
class Test extends Student{
    float sub1;
    float sub2;
    Test(){
        System.out.println("Enter marks out of 50:");
        System.out.println("Subject 1");
        sub1=sc.nextFloat();
        System.out.println("Subject 2");
        sub2=sc.nextFloat();
    }
}
class Result extends Test{
    double result;
    double Display_Result(){
        return ((sub1+sub2)/100)*100;
    }
}

public class Multilevel {
    public static void main(String []args){
        Result r=new Result();
        System.out.println("Result"+ r.Display_Result()+"%"); }}

```

**Inputs:**

```

Enter roll no.
57
Enter marks out of 50:
Subject 1
45
Subject 2
40

```

**Output:**

```

Result 85.0%

```