

**Dr. D.Y. Patil Unitech Society's**  
**Dr. D.Y. Patil Arts, Commerce and Science College Pimpri, Pune-18**

**Department of Computer Science**

**Academic Year: 2025-2026**

**Practical Assignment – 2**

**Class:- T.Y.B.C.A.(Science)**

**Subject:- Programming in JAVA**

**Date:-04/07/2025**

---

- 1. Define a class area. Write calculate method to calculate area of Square, Rectangle, Triangle and Circle. (Use method overloading)**

```
class Area
{
void calculate(int s)
{
System.out.println("Area of Square: "+(s*s));
}
void calculate(double r)
{
System.out.println("Area of Circle: "+(3.14*r*r));
}
void calculate(int l,int b)
{
System.out.println("Area of Rectangle: "+(l*b));
}
```

```

void calculate(double b,double h)
{
System.out.println("Area of Triangle: "+(0.5*b*h));
}

public static void main(String args[])
{
Area a=new Area();
a.calculate(8);
a.calculate(9.5);
a.calculate(5.3,6.2);
a.calculate(8,4);
}
}

```

- 2. Define a “Point” class having members- x, y (coordinates). Define default constructor and parameterized constructor. Define two subclasses “ColorPoint” with member as color and sub class “Point3D” with member as Z(coordinate). Write display method to display the details of different types of Points.**

```

class Point
{
int x,y;

Point()
{
x=0;
y=0;
}
}

```

```
Point(int x,int y)
{
this.x=x;

this.y=y;
}

void display()
{
System.out.println("Point: "+x+","+y);
}
}

class ColorPoint extends Point
{
String color;

ColorPoint()
{
super();
color="";
}

ColorPoint(int x, int y, String color)
{
super(x,y);
this.color=color;
}

void display()
{
```

```
System.out.println("Point: "+super.x+", "+super.y+" Color: "+color);
```

```
}
```

```
}
```

```
class Point3D extends Point
```

```
{
```

```
int z;
```

```
Point3D()
```

```
{
```

```
super();
```

```
z=0;
```

```
}
```

```
Point3D(int x,int y,int z)
```

```
{
```

```
super(x,y);
```

```
this.z=z;
```

```
}
```

```
void display()
```

```
{
```

```
System.out.println("Point: "+super.x+", "+super.y+", "+z);
```

```
}
```

```
public static void main(String args[])
```

```
{
```

```
Point p=new Point(4,6);
```

```
ColorPoint cp=new ColorPoint(4,1,"Red");
```

```
Point3D pd=new Point3D(6,8,9);  
p.display();  
cp.display();  
pd.display();  
}  
}
```

**3. Define a class Bank (name and rate\_of\_interest). Derive two subclasses SBI and HDFC from Bank. Override method of Bank Class.**

```
class Bank  
{  
String name;  
float rate_of_interest;  
String getName()  
{  
return name;  
}  
float getInterest()  
{  
return 6.5f;  
}  
}  
  
class SBI extends Bank  
{  
SBI(String name)  
{
```

```
this.name=name;
```

```
}
```

```
String getName()
```

```
{
```

```
return name;
```

```
}
```

```
float getInterest()
```

```
{
```

```
return 7.6f;
```

```
}
```

```
}
```

```
class HDFC extends Bank
```

```
{
```

```
HDFC(String name)
```

```
{
```

```
this.name=name;
```

```
}
```

```
String getName()
```

```
{
```

```
return name;
```

```
}
```

```
float getInterest()
```

```
{
```

```
return 7.75f;
```

```

}

public static void main(String args[])

{

SBI s=new SBI("State Bank of India");

HDFC h=new HDFC("HDFC Bank");

System.out.println("Rate of Interest for "+s.getName()+" : "+s.getInterest());
System.out.println("Rate of Interest for "+h.getName()+" : "+h.getInterest());

}

}

```

- 4. Define an interface “Operation” which has methods area(),volume(). Define a constant PI having a value 3.142. Create a class circle (member – radius), cylinder (members – radius, height) which implements this interface. Calculate and display the area and volume.**

```

interface Operation {

    double PI = 3.142;

    double area();

    double volume();

}

class Circle implements Operation {

    private double radius;

    public Circle(double radius) {

        this.radius = radius;

    }

    public double area() {

        return PI * radius * radius;

    }

}

```

```
public double volume() {  
    return 0;  
}
```

```
public double getRadius() {  
    return radius;  
}
```

```
public void setRadius(double radius) {  
    this.radius = radius;  
}  
}
```

```
class Cylinder implements Operation {  
    private double radius;  
    private double height;
```

```
public Cylinder(double radius, double height) {  
    this.radius = radius;  
    this.height = height;  
}
```

```
public double area() {  
    return 2 * PI * radius * (radius + height);  
}
```



```

    public double volume() {
        return PI * radius * radius * height;
    }
}

class Main {
    public static void main(String[] args) {
        Circle c1 = new Circle(5);
        Cylinder c2 = new Cylinder(5, 10);
        System.out.println("Circle:");
        System.out.println("Area: " + c1.area());
        System.out.println("Volume: " + c1.volume());
        System.out.println("\nCylinder:");
        System.out.println("Surface Area: " + c2.area());
        System.out.println("Volume: " + c2.volume());
    }
}

```

5. **Write a Java program to create a super class Employee (members – name, salary). Derive a sub-class as Developer (member – projectname). Derive a sub-class Programmer (member – proglanguage) from Developer. Create object of Developer and display the details of it. Implement this multilevel inheritance with appropriate constructor and methods.**

```

class Employee {
    private String name;

```

```
private int id;

public Employee(String name, int id) {
    this.name = name;
    this.id = id;
}

public void displayDetails() {
    System.out.println("Employee Name: " + name);
    System.out.println("Employee ID: " + id);
}
}

class Developer extends Employee {
    private String projectName;

    public Developer(String name, int id, String projectName) {
        super(name, id);
        this.projectName = projectName;
    }

    public void displayDetails() {
        super.displayDetails();
        System.out.println("Project Name: " + projectName);
    }
}

class Programmer extends Developer {
    private String programmingLanguage;

    public Programmer(String name, int id, String projectName, String programmingLanguage) {
```

```

        super(name, id, projectName);
        this.programmingLanguage = programmingLanguage;
    }
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Programming Language: " + programmingLanguage);
    }
}

public class Main1 {
    public static void main(String[] args) {
        Programmer p= new Programmer("Arti", 101, "Project ABC", "Java");
        p.displayDetails();
    }
}

```

**6. Define a class Employee having members – id, name, salary. Define default constructor.**

**Create a subclass called Manager with private member bonus. Define methods accept and display in both the classes. Create “n” objects of the Manager class and display the details of the worker having the maximum total salary (salary + bonus).**

**7. Create a package named “Math” having different classes to perform following operations:**

- **Addition**
- **Subtraction**
- **Multiplication**
- **Division**

**8. Write a java program to accept 2 numbers and perform above operations.**

**Create a package named “Area” having following classes :**

- **Circle**
- **Square**
- **Triangle**
- **Rectangle**

**Write a java program to calculate area and volume for above classes.**

**9. Create a package named “Series” having three different classes to print series:**

- Fibonacci series**
- Cube of numbers**
- Square of numbers**

**Write a java program to generate “n” terms of the above series. Accpet n from user.**

**FibonacciSeries.java**

```
package Series;
```

```
public class FibonacciSeries {  
    public void printSeries(int n) {  
        int a = 0, b = 1, c;  
        System.out.print("Fibonacci Series: ");  
        for (int i = 0; i < n; i++) {  
            if (i <= 1) {  
                c = i;  
            } else {  
                c = a + b;  
                a = b;  
                b = c;  
            }  
            System.out.print(c + " ");  
        }  
    }  
}
```

```
    }  
    System.out.print(c + " ");  
}  
System.out.println();  
}  
}
```

### **CubeSeries.java**

```
package Series;  
  
public class CubeSeries {  
    public void printSeries(int n) {  
        System.out.print("Cubes of Numbers: ");  
        for (int i = 1; i <= n; i++) {  
            System.out.print((i * i * i) + " ");  
        }  
        System.out.println();  
    }  
}
```

### **SquareSeries.java**

```
package Series;  
  
public class SquareSeries {  
    public void printSeries(int n) {
```

```
System.out.print("Squares of Numbers: ");  
for (int i = 1; i <= n; i++) {  
    System.out.print((i * i) + " ");  
}  
System.out.println();  
}  
}
```

### **SeriesMain.java**

```
import Series.FibonacciSeries;  
import Series.CubeSeries;  
import Series.SquareSeries;  
import java.util.Scanner;  
  
public class SeriesMain {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter the number of terms (n): ");  
        int n = sc.nextInt();  
  
        FibonacciSeries f = new FibonacciSeries();  
        CubeSeries c = new CubeSeries();  
        SquareSeries s = new SquareSeries();  
  
        f.printSeries(n);
```

```
        c.printSeries(n);
        s.printSeries(n);
    }
}
```

**10. Create a package “utility”. Define a class CapitalString under “utility” package which will contain a method to return String with first letter capital. Create a Person class (members– name, city) outside the package. Display the person name with first letter as capital by making use of CapitalString.**

**CapitalString.java**

```
package utility;

public class CapitalString {

    public String capitalizeFirstLetter(String input) {
        if (input == null || input.isEmpty()) {
            return input;
        }
        return input.substring(0, 1).toUpperCase() + input.substring(1).toLowerCase();
    }
}
```

**Person.java**

```
import utility.CapitalString;

public class Person {
    private String name;
    private String city;
```

```
public Person(String name, String city) {  
    this.name = name;  
    this.city = city;  
}
```

```
public void displayCapitalizedName() {  
    CapitalString capitalString = new CapitalString();  
    String capitalizedName = capitalString.capitalizeFirstLetter(name);  
    System.out.println("Person's Name: " + capitalizedName);  
}
```

```
public static void main(String[] args) {  
    Person p= new Person("john doe", "new york");  
    p.displayCapitalizedName();  
}  
}
```

- 11. Define an abstract class Staff with members name and address. Define two subclasses of this class – FullTimeStaff (members - department, salary, hra - 8% of salary, da – 5% of salary) and PartTimeStaff (members - number-of-hours, rate-per-hour). Define appropriate constructors. Write abstract method as calculateSalary() in Staff class. Implement this method in subclasses. Create n objects which could be of either FullTimeStaff or PartTimeStaff class by asking the user's choice. Display details of all FullTimeStaff objects and all PartTimeStaff objects along with their salary.**