Programming Assignment Solutions

Question 1: Frequency of Elements in an Array (Java)

```java
import java.util.*;

public class frequency {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = sc.nextInt();

        int[] arr = new int[n];
        System.out.println("Enter " + n + " integers:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        Arrays.sort(arr);

        System.out.println("\nElement Frequency:");
        int count = 1;
        for (int i = 1; i < n; i++) {
            if (arr[i] == arr[i - 1]) {
                count++;
            } else {
                System.out.println(arr[i - 1] + " appeared " + count + " times");
                count = 1;
            }
        }
        System.out.println(arr[n - 1] + " appeared " + count + " times");

        sc.close();
    }
}
```

Question 2: Factorial Calculation (Java)

```java
import java.util.Scanner;

class Factorial {
    long calculate(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        return n * calculate(n - 1);
    }
}

public class factorial {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number to find its factorial: ");
        int num = sc.nextInt();

        Factorial fact = new Factorial();
        long result = fact.calculate(num);
        System.out.println("Factorial of " + num + " is: " + result);

        sc.close();
    }
}
```

Question 3: Simple Calculator (Java)

```
class Calculator {
    public double operate(int a, int b, char operator) {
        switch (operator) {
            case '+':
                return a + b;
            case '-':
                return a - b;
            case '*':
                return a * b;
            case '/':
                if (b == 0) {
                    System.out.println("Error: Division by zero is not allowed.");
                    return 0;
                }
                return (double) a / b;
            default:
                System.out.println("Error: Invalid operator. Use +, -, *, or /.");
                return 0;
        }
    }
}

public class calc {
    public static void main(String[] args) {
        if (args.length != 3) {
            System.out.println("Enter numbers and operation");
            return;
        }
        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[1]);
        char operator = args[2].charAt(0);

        Calculator calculator = new Calculator();
        double result = calculator.operate(num1, num2, operator);

        if (!Double.isNaN(result)) {
            System.out.println("Result: " + result);
        }
    }
}
```

Question 4: Abstract Class and Inheritance (Java)

```
abstract class Shape {
    abstract double area();
}

class Rectangle extends Shape {
    double length;
    double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    @Override
    double area() {
        return length * width;
    }
}

class Circle extends Shape {
    double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    double area() {
        return Math.PI * radius * radius;
    }
}

public class Abstract {
    public static void main(String[] args) {
        Shape rect = new Rectangle(4, 4);
        Shape circle = new Circle(3);

        System.out.println("Area of Rectangle: " + rect.area());
        System.out.println("Area of Circle: " + circle.area());
    }
}
```

Question 5: Interfaces (Java)

```
interface Printable {
    void print();
}

interface Showable {
    void show();
}

class Document implements Printable, Showable {
    @Override
    public void print() {
        System.out.println("Print the document");
    }

    @Override
    public void show() {
        System.out.println("Show the document");
    }
}

public class Interface {
    public static void main(String[] args) {
        Document doc = new Document();
        doc.print();
        doc.show();
    }
}
```

Question 6: Student Class (Java)

```
import java.util.Scanner;

class Student {
    int rollNumber;
    String name;
    int mathMarks;
    int phyMarks;
    int chemMarks;

    public void getRequiredDetails(Scanner sc) {
        System.out.print("Enter Roll Number: ");
        rollNumber = sc.nextInt();
        sc.nextLine();

        System.out.print("Enter Name: ");
        name = sc.nextLine();

        System.out.print("Enter Math Marks: ");
        mathMarks = sc.nextInt();

        System.out.print("Enter Physics Marks: ");
        phyMarks = sc.nextInt();

        System.out.print("Enter Chemistry Marks: ");
        chemMarks = sc.nextInt();
    }

    public void displayAverage() {
        double average = (mathMarks + phyMarks + chemMarks) / 3.0;
        System.out.println("\\nStudent: " + name + " (Roll No: " + rollNumber + ")");
        System.out.println("Average Marks: " + average);
    }
}

public class Students {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Student s1 = new Student();
        s1.getRequiredDetails(sc);
        s1.displayAverage();
        sc.close();
    }
}
```

Question 7: Abstract Class Shape and Inheritance (Java)

```
abstract class Shape {
    public abstract double calculateArea();
    public abstract double calculatePerimeter();
}

class Circle extends Shape {
    double radius;

    public Circle(double radius) {
        this.radius = radius;
    }
    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
    }

    @Override
    public double calculatePerimeter() {
        return 2 * Math.PI * radius;
    }
}

class Triangle extends Shape {
    double sidel;
    double side2;
    double side3;

    public Triangle(double sidel, double side2, double side3) {
        this.sidel = sidel;
        this.side2 = side2;
        this.side3 = side3;
    }

    @Override
    public double calculateArea() {
        double s = (sidel + side2 + side3) / 2;
        return Math.sqrt(s * (s - sidel) * (s - side2) * (s - side3));
    }

    @Override
    public double calculatePerimeter() {
        return sidel + side2 + side3;
    }
}

public class Shapes {
    public static void main(String[] args) {
        Shape circle = new Circle(5);
        Shape triangle = new Triangle(3, 4, 5);

        System.out.println("Circle Area: " + circle.calculateArea());
        System.out.println("Circle Perimeter: " + circle.calculatePerimeter());
        System.out.println("Triangle Area: " + triangle.calculateArea());
        System.out.println("Triangle Perimeter: " + triangle.calculatePerimeter());
    }
}
```

Question 8: Inheritance (Java)

```
class Person {
    String name;
    int age;
    String add;

    public Person(String name, int age, String address) {
        this.name = name;
        this.age = age;
        this.add = address;
    }

    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Address: " + add);
    }
}

class Student extends Person {
    int roll;
    int mark1, mark2, mark3;

    public Student(String name, int age, String address, int rollno, int mark1, int mark2, int 
        super(name, age, address);
        this.roll = rollno;
        this.mark1 = mark1;
        this.mark2 = mark2;
        this.mark3 = mark3;
    }

    @Override
    public void display() {
        System.out.println("Roll Number: " + roll);
        super.display();
        double average = (mark1 + mark2 + mark3) / 3.0;
        System.out.println("Marks: " + mark1 + ", " + mark2 + ", " + mark3);
        System.out.println("Average: " + average);

        String grade;
        if (average >= 90) grade = "A";
        else if (average >= 75) grade = "B";
        else if (average >= 60) grade = "C";
        else grade = "D";

        System.out.println("Grade: " + grade);
    }
}

class Faculty extends Person {
    int faculty;
    String dep;
    double pay;
    double DA;
    public Faculty(String name, int age, String address, int faculty, String dep, double pay, d
        super(name, age, address);
        this.faculty = faculty;
        this.dep = dep;
        this.pay = pay;
        this.DA = DA;
    }

    @Override
    public void display() {
        super.display();
        System.out.println("Faculty ID: " + faculty);
        System.out.println("dep: " + dep);
        double totalSalary = (pay + DA) + 0.70 * (pay + DA);
        System.out.println("Basic Pay: " + pay);
        System.out.println("DA: " + DA);
        System.out.println("Total Salary: " + totalSalary);
    }
}

public class Persons {
    public static void main(String[] args) {
        Student s = new Student("Sudha", 18, "Kochi", 65, 85, 90, 88);
        Faculty f = new Faculty("adi", 44, "ktm", 101, "Computer Science", 50000, 10000);

        System.out.println("=== Student Details ===");
        s.display();

        System.out.println("\\n=== Faculty Details ===");
        f.display();
    }
}
```