1.Display odd numbers between 1 to 100 and count total number of odd number as well as find sum of odd numbers.

```
public class OddNumberDetails {
  public static void main(String[] args) {
    int count = 0;
    int sum = 0;

    System.out.println("Odd numbers between 1 to 100 are:");

    for (int i = 1; i <= 100; i++) {
      if (i % 2 != 0) {
        System.out.print(i + " ");
        count++;
        sum += i;
      }
    }

    System.out.println("\n\nTotal number of odd numbers: " + count);
    System.out.println("Sum of odd numbers: " + sum);
  }
}
```

Sample Output:

```
Odd numbers between 1 to 100 are:

1 3 5 7 9 11 ... 99

Total number of odd numbers: 50

Sum of odd numbers: 2500
```

----

2. Write a Java program to find the sum of first n numbers.

```java
import java.util.Scanner;

public class SumFirstN {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter n: ");

        int n = sc.nextInt();

        int sum = n * (n + 1) / 2;

        System.out.println("Sum of first " + n + " numbers is: " + sum);

    }

}
```

Sample Output:

```
Enter n: 10

Sum of first 10 numbers is: 55
```

---

3. Write a Java program to find sum of digits of a numbers.

```
import java.util.Scanner;

public class SumOfDigits {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number: ");

        int num = sc.nextInt();

        int sum = 0;


        while (num > 0) {

            sum += num % 10;

            num /= 10;

        }


        System.out.println("Sum of digits: " + sum);

    }

}
```

Sample Output:

```
Enter a number: 1234

Sum of digits: 10
```

---

4. Write a java program to calculate electricity bill based on following condition if number of unit less than 200.3 rupees per unit. If number of unit 200<500 4 rupees per unit and number of unit 1000<2000 6 rupees.

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

```
public class ElectricityBill {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of units: ");
        int units = sc.nextInt();
        double bill = 0;

        if (units < 200) {
            bill = units * 3.25;
        } else if (units < 500) {
            bill = units * 4.25;
        } else if (units < 1000) {
            bill = units * 5.25;
        } else if (units <= 2000) {
            bill = units * 6.25;
        } else {
            System.out.println("Above range");
        }

        System.out.println("Electricity bill: ₹" + bill);
    }
}
```

Sample Output:

```
Enter number of units: 350
Electricity bill: ₹1487.5
```

---

5. Write a java program to check whether a number is Armstrong or not.

```
import java.util.Scanner;
public class Armstrong {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int original = num, sum = 0;

        while (num > 0) {
            int digit = num % 10;
            sum += digit * digit * digit;
            num /= 10;
        }
        if (sum == original) {
            System.out.println(original + " is an Armstrong number.");
        } else {
            System.out.println(original + " is not an Armstrong number.");
        }
    }
}
```

Sample Output:

```
Enter a number: 153

153 is an Armstrong number.
```

---

6. Write a Java program to check whether a number is palindrome or not.

```java
import java.util.Scanner;


public class Palindrome {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        int original = num, reversed = 0;


        while (num > 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }


        if (original == reversed) {
            System.out.println(original + " is a Palindrome number.");
        } else {
            System.out.println(original + " is not a Palindrome number.");
        }
    }
}
```

Sample Output:

```
Enter a number: 121
121 is a Palindrome number.
```

 ---

7. Write a java program to check whether a number is prime number or not.

```
import java.util.Scanner;

public class PrimeCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        boolean isPrime = true;

        if (num <= 1) {
            isPrime = false;
        } else {
            for (int i = 2; i <= num / 2; i++) {
                if (num % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

        if (isPrime) {
            System.out.println(num + " is a Prime number.");
        } else {
            System.out.println(num + " is not a Prime number.");
        }
    }
}
```

Sample Output:

```
Enter a number: 7
7 is a Prime number.
```

8.Write a java program to print the table of a particular number.

```
import java.util.Scanner;

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

        System.out.println("Multiplication Table of " + num + ":");
for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }
    }
}
```

Sample Output:

```
Enter a number: 5
Multiplication Table of 5:
5 x 1 = 5
5 x 2 = 10
...
5 x 10 = 50
```

---

9. Write a java program to print a pyramid size stars.

```
import java.util.Scanner;

public class StarPyramid {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of rows: ");
        int rows = sc.nextInt();

        for (int i = 1; i <= rows; i++) {
            // Print spaces
            for (int j = i; j < rows; j++) {
                System.out.print(" ");
            }
            // Print stars
            for (int k = 1; k <= (2 * i - 1); k++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Sample Output (for rows = 5):

```
*
   ***
  *****
 *******
*********
```

---

10. Write a java program to create 10 values perform following task:

(i). Sort array

(ii). Find max

(iii). Find min

(iv). Find average

(v). Find sum of all elements

```
import java.util.Arrays;
import java.util.Scanner;
public class ArrayOperations {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements in array: ");
        int n = sc.nextInt();

        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");

        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        // 1. Sort
        Arrays.sort(arr);
        System.out.println("Sorted Array: " + Arrays.toString(arr));
        // 2. Find Max
        int max = arr[n - 1];
        System.out.println("Maximum value: " + max);

        // 3. Find Min
        int min = arr[0];
        System.out.println("Minimum value: " + min);

        // 4. Find Average
        int sum = 0;
        for (int num : arr) {
            sum += num;
        }
        double avg = (double) sum / n;
        System.out.println("Average: " + avg);

        // 5. Sum
        System.out.println("Sum of all elements: " + sum);
    }
}
```

Sample Output:

```
 Enter number of elements in array: 5
Enter 5 elements:
10 40 30 20 50
```

```
Sorted Array: [10, 20, 30, 40, 50]
Maximum value: 50
Minimum value: 10
Average: 30.0
Sum of all elements: 150
```

---

11. Write a java program that creates 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;
    }
}

class Rectangle extends Shape {
    double length, width;

    Rectangle(double l, double w) {
        length = l;
        width = w;
    }

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

public class Main1 {
    public static void main(String[] args) {
        Rectangle r = new Rectangle(5, 3);
        System.out.println("Area of Rectangle: " + r.getArea());
    }
}
```

Output:

```
Area of Rectangle: 15.0
```

---

12. Write a java program that creates a class called Employee with methods called work() and getsalary(). Create a subclass called HR manager that overrides the work() method and adds a new method called addemployee().

```
class Employee {
    public void work() {
        System.out.println("Employee working...");
    }

    public double getSalary() {
        return 25000;
    }
}

class HRManager extends Employee {
    @Override
    public void work() {
        System.out.println("HR Manager is working...");
    }

    public void addEmployee() {
        System.out.println("Adding new employee...");
    }
}

public class Main2 {
    public static void main(String[] args) {
        HRManager hr = new HRManager();
        hr.work();
        System.out.println("Salary: " + hr.getSalary());
        hr.addEmployee();
    }
}
```

Output:

```
 HR Manager is working...
Salary: 25000.0
```

```
Adding new employee...
```

---

13. Write a java program that creates a class known as “bankaccount” with methods called deposit() and withdraw(). Create a subclass called savingaccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

```
class BankAccount {
  protected double balance = 1000;

  public void deposit(double amount) {
    balance += amount;
    System.out.println("Deposited: ₹" + amount);
  }

  public void withdraw(double amount) {
    balance -= amount;
    System.out.println("Withdrawn: ₹" + amount);
  }

  public void showBalance() {
    System.out.println("Current Balance: ₹" + balance);
  }
}

class SavingAccount extends BankAccount {
  @Override
```

```
    public void withdraw(double amount) {
        if (balance - amount < 100) {
            System.out.println("Withdrawal denied! Balance can't go below ₹100.");
        } else {
            super.withdraw(amount);
        }
    }
}

public class Main3 {
    public static void main(String[] args) {
        SavingAccount sa = new SavingAccount();
        sa.deposit(500);
        sa.withdraw(1300); // Should be denied
        sa.withdraw(200);  // Allowed
        sa.showBalance();
    }
}
```

Output:

```
 Deposited: ₹500.0
Withdrawal denied! Balance can't go below ₹100.
Withdrawn: ₹200.0
Current Balance: ₹1300.0
```

---

14. Write a java program that creates a class called Animal with a method named move(). Create a subclass called cheetah that overrides the move() method to run.

```
class Animal {
   public void move() {
      System.out.println("Animal is moving...");
   }
}

class Cheetah extends Animal {
   @Override
   public void move() {
      System.out.println("Cheetah is running fast...");
   }
}

public class Main4 {
   public static void main(String[] args) {
      Cheetah c = new Cheetah();
      c.move();
   }
}
```

Output:

```
Cheetah is running fast...
```

---

15. Write a java program that creates a class known as person with methods called getFirstName() and getLastName(). Create a subclass called Employee that adds a new method named getEmployeed() and overrides the getLastName() method to include the Employee’s job title.

```
class Person {
    String firstName, lastName;

    Person(String fName, String lName) {
        this.firstName = fName;
        this.lastName = lName;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }
}

class Employee extends Person {
    String jobTitle;
    int employeeID;
    Employee(String fName, String lName, String jobTitle, int empID) {
        super(fName, lName);
        this.jobTitle = jobTitle;
        this.employeeID = empID;
    }

    public int getEmployeeID() {
        return employeeID;
    }

    @Override
    public String getLastName() {
        return lastName + " (" + jobTitle + ")";
    }
}

public class Main5 {
    public static void main(String[] args) {
        Employee emp = new Employee("John", "Doe", "Manager", 101);
        System.out.println("First Name: " + emp.getFirstName());
        System.out.println("Last Name: " + emp.getLastName());
        System.out.println("Employee ID: " + emp.getEmployeeID());
    }
}
```

Output:

```
 First Name: John
Last Name: Doe (Manager)
Employee ID: 101
 ---
```

16. Write a java program that creates a class shape with methods called getperimeter() and getarea().create a subclass called circle that overrides the getperimeter() and getarea()methods to calculate the area and perimeter of a circle .

```
class Shape {
    public double getPerimeter() {
        return 0;
    }

    public double getArea() {
        return 0;
    }
}

class Circle extends Shape {
    double radius;

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

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

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

public class Main6 {
    public static void main(String[] args) {
        Circle c = new Circle(5);
        System.out.println("Perimeter of Circle: " + c.getPerimeter());
        System.out.println("Area of Circle: " + c.getArea());
    }
}
```

Output:

```
Perimeter of Circle: 31.41592653589793
Area of Circle: 78.53981633974483
```

---

17. 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 {
    public void start() {
        System.out.println("Vehicle starting...");
    }
}

class Truck extends Vehicle {
    @Override
    public void start() {
        System.out.println("Truck starting with heavy load...");
    }
}

class Car extends Vehicle {
    @Override
    public void start() {
        System.out.println("Car starting with comfort...");
    }
}

class Motorcycle extends Vehicle {
    @Override
    public void start() {
        System.out.println("Motorcycle starting with speed...");
    }
}
```

```
public class Main7 {
    public static void main(String[] args) {
        Vehicle v1 = new Truck();
        Vehicle v2 = new Car();
        Vehicle v3 = new Motorcycle();

        v1.start();
        v2.start();
        v3.start();
    }
}
```

Output:

```
Truck starting with heavy load...
Car starting with comfort...
Motorcycle starting with speed...
```

---

18 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,salary,address and job title.Implement methods for calculating bonuses,generating perfoemance reports,and managing projects.

```
class Employee {
    String name, address, jobTitle;
    }

    double salary;

    Employee(String name, String address, double salary, String jobTitle) {
        this.name = name;
        this.address = address;
        this.salary = salary;
        this.jobTitle = jobTitle;
    }

    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Address: " + address);
        System.out.println("Salary: ₹" + salary);
        System.out.println("Job Title: " + jobTitle);
    }
}

class Manager extends Employee {
    Manager(String name, String address, double salary) {
        super(name, address, salary, "Manager");
    }

    public void calculateBonus() {
        System.out.println("Manager bonus: ₹" + (salary * 0.20));
    }
    public void manageProject() {
        System.out.println("Managing a large project...");
    }
}

class Developer extends Employee {
    Developer(String name, String address, double salary) {
        super(name, address, salary, "Developer");
    }

    public void calculateBonus() {
        System.out.println("Developer bonus: ₹" + (salary * 0.15));
    }

    public void generateReport() {
        System.out.println("Generating development report...");
    }
}

class Programmer extends Employee {
    Programmer(String name, String address, double salary) {
        super(name, address, salary, "Programmer");
    }

    public void calculateBonus() {
        System.out.println("Programmer bonus: ₹" + (salary * 0.10));
```

```java
    }

    public void generateReport() {
        System.out.println("Generating code review report...");
    }
}

public class Main8 {
    public static void main(String[] args) {
        Manager m = new Manager("Alice", "Delhi", 80000);
        m.displayDetails();
        m.calculateBonus();
        m.manageProject();

        System.out.println();

        Developer d = new Developer("Bob", "Mumbai", 60000);
        d.displayDetails();
        d.calculateBonus();
        d.generateReport();
    }
}
```

Output:

```
Name: Alice
Address: Delhi
```

```
Salary: ₹80000.0
Job Title: Manager
Manager bonus: ₹16000.0
Managing a large project...

Name: Bob
Address: Mumbai
Salary: ₹60000.0
Job Title: Developer
Developer bonus: ₹9000.0
Generating development report...

--------------------
```