

1)

To demonstrate the use of exception handling in Java by performing division and handling the `ArithmeticException` that occurs when dividing by zero.

```
Package vaishnav;
```

```
Public class Abcd {
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Int numerator=20;
```

```
        Int denominator=0;
```

```
        Try{
```

```
            Int result = numerator/denominator;
```

```
            System.out.println(result);
```

```
        }
```

```
        Catch(ArithmeticException e) {
```

```
            System.out.println("Error: cannot divisible by zero");
```

```
        }
```

```
    }
```

```
}
```

2)To demonstrate the use of constructors in Java for adding two numbers and displaying their sum.

Package vaishnav;

```
Public class AddNum {  
    Int c ;  
    AddNum(int a, int b){  
        C = a+b;  
    }  
    Public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        AddNum a2 = new AddNum(2,3);  
        System.out.println("sum is: "+a2.c);  
    }  
}
```

3)To read characters from a file and write alternate characters to another file using Java file handling, demonstrating the use of FileReader, FileWriter, and exception handling.

```
Package vaishnav;

Import java.io.FileReader;

Import java.io.FileWriter;

Import java.io.IOException;

Public class AlternateCharacterCopy {

    Public static void main(String[] args) {

        // TODO Auto-generated method stub

        Try{

            FileReader reader=new FileReader("file1.txt");

            FileWriter writer=new FileWriter("file2.txt");

            Int character;

            Int index=0;

            While ((character=reader.read())!=-1) {

                If (index %2==0) {

                    Writer.write(character);

                }

                Index++;

            }

            Writer.close();

            Reader.close();

        }

    }

}
```

```

        System.out.println("Alternate character copied succesfully");
    }
    Catch (IOException e) {
        System.out.println("An error occurred"+e.getMessage());
        e.printStackTrace();
    }
}
}
}

```

4)To demonstrate the declaration, initialization, and printing of different data types (int, double, and char) in Java.

```
Package vaishnav;
```

```
Public class Example {
```

```

    Public static void main(String[] args) {
        // TODO Auto-generated method stub
        Int a = 50;
        Double b = 10.98;
        Char c = 'V';
    }
}

```

```
        System.out.println("int :"+a);  
        System.out.println("double :"+b);  
        System.out.println("char:"+c);  
    }  
}
```

5)To demonstrate the concept of multithreading in Java by creating and managing two threads that print odd and even numbers, and using the join() method to control the execution order of threads.

```
Package vaishnav;  
  
Import java.lang.*;  
  
Class Group1 extends Thread{  
    Public void run() {  
        For(int i=1;i<20;i+=2) {  
            System.out.println("Odd Thread:"+i);  
        }  
    }  
}
```

```

Class Group2 extends Thread{
    Public void run() {
        For(int i=2;i<20;i+=2) {
            System.out.println("Even Thread:"+i);
        }
    }
}

Public class Group3 {

    Public static void main(String[] args) {
        // TODO Auto-generated method stub

        Group1 g1 = new Group1();
        Group2 g2 = new Group2();
        G2.start();

        Try {
            G2.join();
        }

        Catch(InterruptedException e) {
            System.out.println("Error Occurred");
        }

        G1.start();
    }

}

```

6)To demonstrate the use of a non-static method in Java for performing the addition of two numbers and returning the result.

```
Package vaishnav;
```

```
Public class NonstaticMethod {
```

```
    Int add(int c,int d) {
```

```
        Int sum = c+d;
```

```
        Return sum;
```

```
    }
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        NonstaticMethod v1 = new NonstaticMethod();
```

```
        Int result = v1.add(32, 64);
```

```
        System.out.println("The sum is "+result);
```

```
    }
```

```
}
```

7)To demonstrate exception handling in Java by handling a NullPointerException using a try-catch block and ensuring smooth program execution.

```
Package vaishnav;
```

```
Public class NullPointerException {
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        String str = null;
```

```
        Try {
```

```
            System.out.println(str.length());
```

```
        }
```

```
    Catch(Exception e) {
```

```
        System.out.println("Error: NullPointerException occurs");
```

```
    }
```

```
        System.out.println("the program continues smoothly..... ");
```

```
    }
```

```
}
```


8)To perform and demonstrate basic arithmetic operations (addition, subtraction, multiplication, division, and modulus) in Java using operators.

```
Package vaishnav;
```

```
Public class Operators {
```

```
    Public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Int a = 10, b =15;  
  
        Int add = a+b;  
  
        Int sub = a-b;  
  
        Int product = a*b;  
  
        Int div = a/b;  
  
        Int mod = a%b;  
  
        System.out.println("add"+ add);  
  
        System.out.println("sub"+ sub);  
  
        System.out.println("product" +product);  
  
        System.out.println("div" +div);  
  
        System.out.println("mod"+ mod);  
  
    }
```

```
}
```

9)To demonstrate the handling of `ArrayIndexOutOfBoundsException` in Java when accessing an invalid index in an array using a try-catch block.

```
Package vaishnav;
```

```
Public class Qrst {
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Int[]arr= {4,5,6};
```

```
        System.out.println("number of element : "+arr.length);
```

```
        Try {
```

```
            System.out.println("Accessing element : "+arr[7 ]);
```

```
        }
```

```
        Catch(ArrayIndexOutOfBoundsException e) {
```

```
            System.out.println("Error: the number is not present in the array");
```

```
        }
```

```
    }
```

```
}
```

10)To demonstrate the use of a default constructor in Java for initializing object data members and displaying their values.

```
Package vaishnav;
```

```
Public class Rcoem {
```

```
    String name;
```

```
    Int Rollno;
```

```
    Rcoem(){
```

```
        System.out.println("Constructor called");
```

```
        Name = "Vaishnu";
```

```
        Rollno = 49;
```

```
    }
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Rcoem obj = new Rcoem();
```

```
        System.out.println("The name is : "+obj.name);
```

```
        System.out.println("The Rollno is : "+obj.Rollno);
```

```
    }
```

```
}
```

11)To demonstrate the use of a static method in Java for performing the addition of two numbers and returning the result.

```
Package vaishnav;
```

```
Public class StaticMethod {
```

```
    Static int addNum(int x,int y) {
```

```
        Return x+y;
```

```
    }
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Int c=4,d=6;
```

```
        Int result = addNum(c,d);
```

```
        System.out.println("The sum is "+result);
```

```
    }
```

```
}
```

12)To demonstrate the creation of a simple Java class with instance variables, initializing those variables, and printing their values.

```
Package vaishnav;
```

```
Public class Student {  
    Int Rollno;  
    String name;  
    Public static void main(String [] args) {  
        Student s1 = new Student();  
        S1.Rollno = 01;  
        S1.name = "Ram";  
        System.out.println(s1.Rollno);  
        System.out.println(s1.name);  
    }  
}
```

13)To demonstrate the creation and use of multiple objects in Java and the invocation of an instance method to display the values of instance variables.

```
Package vaishnav;
```

```
Public class StudentMethod2 {
```

```
    Int Rollno;
```

```
    String name;
```

```
    Void displayMethod() {
```

```
        System.out.println(Rollno+" "+name);
```

```
    }
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        StudentMethod2 s1 = new StudentMethod2();
```

```
        StudentMethod2 s2 = new StudentMethod2();
```

```
        StudentMethod2 s3 = new StudentMethod2();
```

```
        StudentMethod2 s4 = new StudentMethod2();
```

```
        StudentMethod2 s5 = new StudentMethod2();
```

```
        S1.displayMethod();
```

```
        S2.displayMethod();
```

```
        S3.displayMethod();
```

```
        S4.displayMethod();
```

```
        S5.displayMethod();
```

```
}  
  
}
```

14)To demonstrate the use of a method in Java that adds two user-input numbers and displays the result, using the Scanner class for input handling.

```
Package Vaishanv2022;
```

```
Import java.util.*;
```

```
Public class Aaetunc{
```

```
    Void add(int a,int b) {
```

```
        Int sum=a+b;
```

```
        System.out.println("sum: "+sum);
```

```
    }
```

```
    Public static void main(String[] args) {
```

```
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("Enter the value of a: ");
```

```
        Int a = s.nextInt();
```

```
        System.out.println("a: "+a);
```

```
        System.out.println("Enter the value of b: ");
```

```
        Int b = s.nextInt();
```

```
        System.out.println("b: "+b);
```

```
        Aaetunc f = new Aaetunc();  
        f.add(a, b);  
    }  
}
```

15)To demonstrate the use of a for loop in Java for iterating through a range of values and printing a message with the current iteration value.

```
Package Vaishanv2022;
```

```
Public class Ab {
```

```
    Public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Int a;  
        For(a=0;a<10;a++) {  
            System.out.println("This is a" +a);  
        }  
    }  
}
```



```
}
```

16)To demonstrate the creation and management of a bank account in Java, including depositing and withdrawing money, and displaying the account balance using methods in a class.

```
Package Vaishanv2022;

Import java.util.*;

Public class BankAccount {

String accountnumber;

String accountholder;

Double balance;

BankAccount(String acc,String acco,double bal){

    Accountnumber = acc;

    Accountholder = acco;

    Balance = bal;

}

Void deposit(double amount) {

    If(amount > 0) {

        Balance += amount;

        System.out.println("Invalid Amount");

    }

}
```

```

Void withdrawal(double amount) {
    If (amount > 0 && amount <= balance) {
        Balance = amount;
        System.out.println("Withdrawn:"+amount);
    }
    Else {
        System.out.println("Insufficient balance");
    }
}

Void displaybalance() {
    System.out.println("Account :"+accountnumber);
    System.out.println("Holder :"+accountholder);
    System.out.println("Balance :"+balance);
}

Public static void main(String[] args) {
    // TODO Auto-generated method stub
    Scanner c1 = new Scanner(System.in);
    String accountnumber1 = c1.nextLine();
    String accountholder1 = c1.nextLine();
    Double balance1 = c1.nextDouble();
    BankAccount acc1 = new
BankAccount(accountnumber1,accountholder1,balance1);
    Acc1.displaybalance();
    Acc1.deposit(50000);
    Acc1.withdrawal(2000000);
    Acc1.displaybalance();
}

```

```
}  
  
}
```

17)To demonstrate the use of a class and object in Java for calculating the volume of a box by using its width, height, and depth.

```
Package Vaishanv2022;  
  
Class Box{  
    Double w;  
    Double h;  
    Double d;  
}  
  
Public class Boxdemo {  
  
    Public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Box b = new Box();  
        Double vol;  
        b.w=8;  
        b.h=9;
```

```

        b.d=2;

        vol = b.w*b.h*b.d;

        System.out.println("Volume is: "+vol);

    }

}

```

18)To demonstrate method usage in Java by creating a class with a method to calculate and display the volume of a box, using method parameters for width, height, and depth.

```

Package Vaishanv2022;

Class Box3{

    Void Casting(double width,double height,double depth){

        Double vol= width*height*depth;

        System.out.println("Volume is:"+vol );

    }

}

Public class Boxdemo3 {

    Public static void main(String[] args) {

```

```

        // TODO Auto-generated method stub

        Box3 b1 =new Box3();

        B1.Casting(1,2,3);

    }

}

```

19)To demonstrate method overriding in Java, where a subclass provides a specific implementation of a method defined in its superclass.

```

Package Vaishanv2022;

Class Animal{

    Void makeSound(){

        System.out.println("I am an Animal");

    }

}

Public class Caaat extends Animal{

    Void makeSound() {

        System.out.println("I am Cat");

    }

    Public static void main(String[] args) {

```

```

        // TODO Auto-generated method stub

        Caaat c1 = new Caaat ();

        C1.makeSound();

    }

}

```

20) To demonstrate the use of an abstract class and method inheritance in Java by creating a calculator with basic operations (addition, subtraction, multiplication, division) and using the subclass to implement these methods.

```

Package Vaishanv2022;

Abstract class Calculator{

    Int p;

    Public void add(int m,int n) {

        P=m+n;

        System.out.println("Sum: "+p);

    }

    Public void sub(int m,int n) {

        P=m-n;

        System.out.println("Difference: "+p);

    }

    Public void multi(int m,int n) {

```

```

        P=m*n;

        System.out.println("Product: "+p);
    }

    Public void div(int m,int n) {

        P=m/n;

        System.out.println("Result: "+p);

    }

}

Public class Calci extends Calculator {

    Public static void main(String[] args) {

        // TODO Auto-generated method stub

        Calci c = new Calci();

        c.add(4,6);

        c.sub(68, 60);

        c.multi(2, 30);

        c.div(68, 02);

    }

}

```

21)To demonstrate the use of interfaces in Java by creating a calculator interface with basic operations and implementing these operations in a class.

```
Package Vaishanv2022;
```

```
Interface Calculator1{
```

```
    Public void add(int m,int n);
```

```
    Public void sub(int m,int n);
```

```
    Public void multi(int m,int n);
```

```
    Public void div(int m,int n);
```

```
}
```

```
Public class Calci1 implements Calculator1 {
```

```
    Public void add(int m,int n) {
```

```
        Int p=m+n;
```

```
        System.out.println("Sum: "+p);
```

```
    }
```

```
    Public void sub(int m,int n) {
```

```
        Int p=m-n;
```

```
        System.out.println("Difference: "+p);
```

```
    }
```

```
    Public void multi(int m,int n) {
```

```
        Int p=m*n;
```

```
        System.out.println("Product: "+p);
```

```
    }
```

```
    Public void div(int m,int n) {
```

```
        Double p=m/n;
```

```
        System.out.println("Result: "+p);
```



```

    }

    Public static void main(String[] args) {

        // TODO Auto-generated method stub

        Calci1 c = new Calci1();

        c.add(5465,876);

        c.sub(6867, 680);

        c.multi(86, 30);

        c.div(90, 5);

    }

}

```

22)To demonstrate method overriding in Java, where the subclass provides its own implementation of a method defined in the superclass.

```

Package Vaishanv2022;

Class Vehical{

    Void drive() {

        System.out.println("Idk");

    }

}

Public class Car extends Vehical {

```

```

Void drive() {
    System.out.println("Repairing a car");
}

Public static void main(String[] args) {
    // TODO Auto-generated method stub
    Vehical f = new Vehical();
    f.drive();
}
}

```

23)To demonstrate the conversion of distance units in Java by implementing methods that convert kilometers to meters, meters to centimeters, and centimeters to millimeters based on user input.

```

Package Vaishanv2022;

Import java.util.Scanner;

Public class DistanceConversion {
    Int km, m, cm, ml;

    Void kmtom() {
        M = km * 1000;
    }
}

```

```

        System.out.println("The value in meters: " + m);
    }

    Void mtocm() {
        Cm = m * 100;

        System.out.println("The value in centimeters: " + cm);
    }

    Void cmtoml() {
        Ml = cm * 10;

        System.out.println("The value in millimeters: " + ml);
    }

    Public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        DistanceConversion dc = new DistanceConversion();

        System.out.print("Enter the value in kilometers: ");
        dc.km = sc.nextInt();

        dc.kmtom();

        System.out.print("Enter the value in meters: ");
        dc.m = sc.nextInt();

        dc.mtocm();

        System.out.print("Enter the value in meters: ");
        dc.cm = sc.nextInt();

        dc.cmtoml();
    }
}

```

24)To determine whether a given number is even or odd in Java using the modulus operator and conditional statements.

```
Package Vaishanv2022;
```

```
Public class EvenOdd {
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Int a=8;
```

```
        If(a%2==0) {
```

```
            System.out.println("a is even");
```

```
        }
```

```
        Else {
```

```
            System.out.println("a is odd");
```

```
        }
```

```
    }
```

```
}
```

25)To demonstrate the use of conditional if statements in Java to compare two variables and execute different actions based on their values.

```
Package Vaishanv2022;
```

```
Public class IfSample {
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Int x,y;
```

```
        X=10;
```

```
        Y=20;
```

```
        If(x<y)System.out.println("x is less than y");
```

```
        X = x*2;
```

```
        If(x==y)System.out.println("now x is equal to y");
```

```
        X = x*2;
```

```
        If(x>y)System.out.println("now x is greater than y");
```

```
        If(x==y)System.out.println("You won't see this");
```

```
    }
```

```
}
```

26)To demonstrate the default initialization of instance variables in Java, showing how primitive and reference types are initialized in a class.

```
Package Vaishanv2022;
```

```
Class Test {
```

```
    Int I;Test t;Boolean b;byte bt;float ft;
```

```
}
```

```
Public class Main {
```

```
    Public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Test t = new Test();
```

```
        System.out.println(t.i);
```

```
        System.out.println(t.t);
```

```
        System.out.println(t.b);
```

```
        System.out.println(t.bt);
```

```
        System.out.println(t.ft);
```

```
}  
  
}
```

27)To demonstrate the creation of an instance method that stores and displays personal information (name, age, and gender) in Java.

```
Package Vaishanv2022;
```

```
Public class PratikshaB {
```

```
    Public void info(){
```

```
        String name;
```

```
        Int age;
```

```
        Char gender;
```

```
        Name = "Pratiksha";
```

```
        Age = 19;
```

```
        Gender = 'F';
```

```
        System.out.println("Fill the info");
```

```
        System.out.println("Name: "+name);
```

```
        System.out.println("Age:"+age);
```

```

        System.out.println("Gender:"+gender);
    }

    Public static void main(String[] args) {

        // TODO Auto-generated method stub

        PratikshaB pb = new PratikshaB();

        Pb.info();

    }

}

```

28)To demonstrate method overriding in Java, where a subclass provides a specific implementation of a method (calculating the area) defined in the superclass (Shape).

```

Package Vaishanv2022;

Class Shape{

    Void getArea(int length,int breadth) {

        System.out.println("Area is ");

    }

}

Public class Rectangle extends Shape{

    Void getArea(int length,int breadth) {

        Int area = length*breadth;
    }

}

```



```
        System.out.println("The area is : "+area);
    }

    Public static void main(String[] args) {
        // TODO Auto-generated method stub
        Rectangle R1 = new Rectangle();
        R1.getArea(6,5);
    }
}
```

29)To demonstrate the use of the Scanner class in Java for accepting user input, specifically capturing a string (name) and displaying it.

```
Package Vaishanv2022;

Import java.util.Scanner;

Public class Scannerclass {

    Public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("Enter your name: ");  
        Double name = s.nextLine();  
        System.out.println("name is : "+name);  
    }  
}
```

30)To demonstrate the use of the ternary operator in Java for calculating the absolute value of a number.

```
Package Vaishanv2022;
```

```
Public class Terminay {
```

```
    Public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Int l,j;  
        l=10;  
        J=i<0?-i:l;  
        System.out.println("Absolute value of");
```

```
        System.out.println(i+"is"+j);

    }

}
```

31)To demonstrate inheritance in Java by creating a class hierarchy with Railway, Express, and VandeBharat, showcasing method inheritance and usage from parent to child classes.

```
Package Vaishanv2022;
```

```
Class Railway {
```

```
    Void speed() {
```

```
        System.out.println("Hello this is Railway");
```

```
    }
```

```
}
```

```
Class Express extends Railway {
```

```
    Void fast() {
```

```
        System.out.println("Hello this is Express");
```

```
    }
```

```
}
```

```
Public class VandeBharat extends Railway {
```

```
Void superfast() {  
    System.out.println("Hello this is Vande Bharat");  
}  
  
Public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    VandeBharat vb = new VandeBharat();  
    Express ex = new Express();  
    Ex.speed();  
    Ex.fast();  
    Vb.speed();  
    Vb.superfast();  
}  
}
```