

1. Write a program to find the sum of individual digits of a positive integer.

```
import java.util.Scanner;
public class SumOfDigits
{
    public static void main(String args[])
    {
        int number, digit, sum = 0;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number: ");
        number = sc.nextInt();
        while(number > 0)
        {
            digit = number % 10;
            sum = sum + digit;
            number = number / 10;
        }
        System.out.println("Sum of Digits: "+sum);
    }
}
```

Output:

Enter the number: 7896

Sum of Digits: 30

2. Develop a java application with an Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club funds. Generate pay slips for the employees with their gross and net salary

```
class Employee {
    String Emp_name;
    String Emp_id;
```

```
String Address;
String Mail_id;
String Mobile_no;

Employee(String name, String id, String address,
String mail, String mobile) {
    Emp_name = name;
    Emp_id = id;
    Address = address;
    Mail_id = mail;
    Mobile_no = mobile;
}

class Programmer extends Employee {
    double BasicPay;

    Programmer(String name, String id, String address,
String mail, String mobile, double pay) {
        super(name, id, address, mail, mobile);
        BasicPay = pay;
    }

    double calculateSalary() {
        double DA = 0.97 * BasicPay;
        double HRA = 0.10 * BasicPay;
        double PF = 0.12 * BasicPay;
        double StaffClubFund = 0.001 * BasicPay;
        double grossSalary = BasicPay + DA + HRA;
        double netSalary = grossSalary - PF -
StaffClubFund;
        return netSalary;
    }

    void generatePaySlip() {
        double netSalary = calculateSalary();
        System.out.println("Pay Slip for " + Emp_name);
        System.out.println("Employee ID: " + Emp_id);
        System.out.println("Basic Pay: " + BasicPay);
        System.out.println("Net Salary: " + netSalary);
    }
}
```

```

    }
}

public class EmployeeSalary {
    public static void main(String[] args) {
        Programmer p = new Programmer("John", "P123",
            "123 Street", "john@example.com", "1234567890", 50000);
        p.generatePaySlip();
    }
}

```

Output:

```

Pay Slip for John
Employee ID: P123
Basic Pay: 50000.0
Net Salary: 57300.0

```

3. Write a program to generate the first n terms of the sequence.

```

import java.util.*;
public class Sequence {
    public static void main(String[] args)
    {
        System.out.println("Enter Number: ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        for(int i = 1; i <= n; i++)
            System.out.print(i+" ");
    }
}

```

Output

```

Enter Number:
10
1 2 3 4 5 6 7 8 9 10

```

4. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape. Use interface

```

abstract class Shape {
    int dimension1;
    int dimension2;

    Shape(int d1, int d2) {
        dimension1 = d1;
        dimension2 = d2;
    }

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        super(length, breadth);
    }

    void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " +
area);
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        super(base, height);
    }

    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " +
area);
    }
}

```

```
class Circle extends Shape {
    Circle(int radius) {
        super(radius, 0);
    }

    void printArea() {
        double area = Math.PI * Math.pow(dimension1,
2);
        System.out.println("Area of Circle: " + area);
    }
}

public class ShapeTest {
    public static void main(String[] args) {
        Shape s1 = new Rectangle(10, 20);
        s1.printArea();

        Shape s2 = new Triangle(10, 20);
        s2.printArea();

        Shape s3 = new Circle(7);
        s3.printArea();
    }
}
```

Output:

```
Area of Rectangle: 200
Area of Triangle: 100.0
Area of Circle: 153.93804002589985
```

5. Write a program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

```
import java.util.Scanner;

public class PrimeNumbers{

    public static void main(String arg[]){

        int i,n,counter, j;

        Scanner scanner = new Scanner(System.in);

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

        n=scanner.nextInt();

        System.out.print("Prime numbers between 1 to n are ");

        for(j=2;j<=n;j++){

            counter=0;

            for(i=1;i<=j;i++){

                if(j%i==0){

                    counter++;

                }

            }

            if(counter==2)

                System.out.print(j+" ");

        }

    }

}
```

Output:

Enter the n value : 32

Prime numbers between 1 to n are 2 3 5 7 11 13 17 19 23 29
31

6. Write a program to perform string operations using ArrayList. Write functions for the following a. Append - add at end b. Insert – add at particular index c. Search d. List all string starts with given letter

```
import java.util.ArrayList;

public class StringOperations {

    ArrayList<String> strings = new ArrayList<>();

    void appendString(String str) {
        strings.add(str);
    }

    void insertString(int index, String str) {
        strings.add(index, str);
    }

    boolean searchString(String str) {
        return strings.contains(str);
    }

    void listStringsStartingWith(char letter) {
        for (String str : strings) {
            if (str.charAt(0) == letter) {
                System.out.println(str);
            }
        }
    }

    public static void main(String[] args) {
        StringOperations so = new StringOperations();

        so.appendString("Apple");
        so.appendString("Banana");
        so.appendString("Avocado");
        so.insertString(1, "Apricot");

        System.out.println("Search for 'Banana': " +
so.searchString("Banana"));
```

```

        System.out.println("Strings starting with
'A':");
        so.listStringsStartingWith('A');
    }
}

```

Output:

```

Search for 'Banana': true
Strings starting with 'A':
Apple
Apricot
Avocado

```

7. Write a program to find both the largest and smallest number in a list of integers.

```

import java.util.*;
public class MinMax
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter array size: ");
        int n = sc.nextInt();
        int arr[]=new int[n];
        System.out.print("Enter array elements: ");
        for(int i=0;i<n;i++)
            arr[i]=sc.nextInt();
        System.out.println("Entered Array: ");
        for(int i=0;i<n;i++)
            System.out.print(arr[i]+ " ");
        int min=arr[0],max=arr[0];
        for(int i=0;i<n;i++)
        {
            if(min>arr[i])
                min=arr[i];

```



```

if(max<arr[i])
max=arr[i];
}
System.out.println("\nMaximum is : "+max);
System.out.println("Minimum is : "+min);
}
}

```

Output:

```

Enter array size: 5
Enter array elements: 67
45
89
80
3
Entered Array:
67 45 89 80 3
Maximum is : 89
Minimum is : 3

```

-
8. Implement exception handling for dividing by zero exception and array index out of bounds exception.

```

public class ExceptionHandling
{
public static void main(String[] args)
{
    int[] numbers = {1, 2, 3, 4, 5};
    try
    {
        int result = divide(10, 0);
        System.out.println("Result: " + result);
    }
    catch (ArithmeticException e)
    {

```

```

        System.out.println("Error: Division by zero is
not allowed. " + e.getMessage());
    }

    try
    {
        int value = numbers[5];
        System.out.println("Value at index 5: " +
value);
    }
    catch (ArrayIndexOutOfBoundsException e)
    {
        System.out.println("Error: Array index out
of bounds. " + e.getMessage());
    }
}
}

```

Output

```

Error: Division by zero is not allowed
Error: Array index out  of bounds.

```

9. Write a Java program to create a new array list, add some colors and print the collection.

```

import java.util.ArrayList;
public class RainbowColors {
    public static void main(String[] args) {
        ArrayList<String> rainbow = new ArrayList<>();
        rainbow.add("Violet");
        rainbow.add("Indigo");
        rainbow.add("Blue");
        rainbow.add("Green");
        rainbow.add("Yellow");
        rainbow.add("Orange");
        rainbow.add("Red");
        // Retrieve the 2nd color (index 1)
        String secondColor = rainbow.get(1);
    }
}

```

```

        System.out.println("The 2nd color in the
rainbow is: " + secondColor);
        // Retrieve the 5th color (index 4)
        String fifthColor = rainbow.get(4);
        System.out.println("The 5th color in the
rainbow is: " + fifthColor);
    }
}

```

Output

```

The 2nd color in the rainbow is: Indigo
The 5th color in the rainbow is: Yellow

```

10.Design a Java interface for ADT Stack using Java collections.
import java.util.Stack;

```

public class ShoppingCartStack {
    public static void main(String[] args) {
        Stack<String> cart = new Stack<>();

        System.out.println("Adding items to cart:");
        cart.push("Laptop");
        System.out.println("Added: Laptop, Cart: " +
cart);
        cart.push("Mouse");
        System.out.println("Added: Mouse, Cart: " +
cart);
        cart.push("Keyboard");
        System.out.println("Added: Keyboard, Cart: " +
cart);

        System.out.println("\nRemoving the last added
item:");
        if (!cart.empty()) {
            String removedItem = cart.pop();
            System.out.println("Removed: " +
removedItem + ", Cart: " + cart);
        }
        System.out.println("\nLast item in cart:");
        if (!cart.empty()) {

```

```

        System.out.println("Top item: " +
        cart.peek());
    }

    System.out.println("\nRemoving another item:");
    if (!cart.empty()) {
        String removedItem = cart.pop();
        System.out.println("Removed: " +
        removedItem + ", Cart: " + cart);
    }
    System.out.println("\nIs the cart empty? " +
    cart.empty());
}

```

Output

Adding items to cart:

Added: Laptop, Cart: [Laptop]

Added: Mouse, Cart: [Laptop, Mouse]

Added: Keyboard, Cart: [Laptop, Mouse, Keyboard]

Removing the last added item:

Removed: Keyboard, Cart: [Laptop, Mouse]

Last item in cart:

Top item: Mouse

Removing another item:

Removed: Mouse, Cart: [Laptop]

Is the cart empty? false

11. Write a Java program to create a new array list, add some colors and shuffle the arraylist.

```

import java.util.ArrayList;
import java.util.Collections;
public class ShuffleRainbow {
    public static void main(String[] args) {

```

```

        // Create an ArrayList to store the colors of
the rainbow
        ArrayList<String> rainbow = new ArrayList<>();
        // Add the colors in order
        rainbow.add("Violet");
        rainbow.add("Indigo");
        rainbow.add("Blue");
        rainbow.add("Green");
        rainbow.add("Yellow");
        rainbow.add("Orange");
        rainbow.add("Red");
        System.out.println("Original order of colors: "
+ rainbow);
        // Shuffle the elements of the ArrayList
        Collections.shuffle(rainbow);
        System.out.println("Shuffled order of colors: "
+ rainbow);
    }
}

```

Output

```

Original order of colors: [Violet, Indigo, Blue, Green,
Yellow, Orange, Red]
Shuffled order of colors: [Blue, Yellow, Indigo, Green,
Violet, Red, Orange]

```

12. Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes

```

import java.io.File;

public class FileInfo {
    public static void main(String[] args) {
        File file = new File("test.txt");

        if (file.exists()) {
            System.out.println("File exists");
            System.out.println("Readable: " +
file.canRead());

```

```

        System.out.println("Writable: " +
file.canWrite());
        System.out.println("File Type: " +
(file.isDirectory() ? "Directory" : "File"));
        System.out.println("File Size: " +
file.length() + " bytes");
    } else {
        System.out.println("File does not exist");
    }
}
}

```

Output:

```

File exists
Readable: true
Writable: true
File Type: File
File Size: 1024 bytes

```

13. Write a Java program using Queue Collection for Cinema Ticket Sale.

```

import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

class CinemaTicketQueue {
    public static void main(String[] args) {
        // Create a Queue for customers
        Queue<String> ticketQueue = new LinkedList<>();
        Scanner scanner = new Scanner(System.in);

        // Adding customers to the queue
        System.out.println("Enter customer names (type
'done' to stop): ");
        while (true) {
            String name = scanner.nextLine();
            if (name.equalsIgnoreCase("done")) break;
            ticketQueue.add(name);
        }
    }
}

```

```

        // Processing ticket sales
        System.out.println("\nProcessing ticket
sales...");
        while (!ticketQueue.isEmpty()) {
            String customer = ticketQueue.poll(); //
Serve the first customer
            System.out.println("Ticket sold to: " +
customer);
        }

        System.out.println("\nAll tickets sold. Queue
is empty!");
        scanner.close();
    }
}

```

Output

How It Works:

Customers enter their names → They are added to the Queue.

First customer in line gets served first (poll() method).

Loop continues until all customers are served.

Output

Enter customer names (type 'done' to stop):

A

B

C

E

F

D

done

Processing ticket sales...

Ticket sold to: A

Ticket sold to: B

Ticket sold to: C
Ticket sold to: E
Ticket sold to: F
Ticket sold to: D

All tickets sold. Queue is empty!

14. Write a java program that implements a multi-threaded application that has three threads. The first thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.

```
import java.util.Random;
class Square extends Thread
{
    int x;
    Square(int n)
    {
        x = n;
    }
    public void run()
    {
        int sqr = x * x;
        System.out.println("Square of " + x + " = " + sqr );
    }
}
class Cube extends Thread
{
    int x;
    Cube(int n)
    {
        x = n;
    }
    public void run()
    {
        int cub = x * x * x;
        System.out.println("Cube of " + x + " = " + cub );
    }
}
```



```

}
class Number extends Thread
{
    public void run()
    {
        Random random = new Random();
        for(int i =0; i<10; i++)
        {
            int randomInteger = random.nextInt(100);
            System.out.println("Random Integer generated : " +
randomInteger);
            if((randomInteger%2) == 0)
            {
                Square sThread = new Square(randomInteger);
                sThread.start();
            }
            else {
                Cube cThread = new Cube(randomInteger);
                cThread.start();
            }
            try
            {
                Thread.sleep(1000);
            }
            catch (InterruptedException ex) {
                System.out.println(ex); }

        }
    }
}

public class Multithreading {
    public static void main(String args[])
    {
        Number n = new Number();
        n.start();
    }
}

Output
Random Integer generated : 80
Square of 80 = 6400

```

Random Integer generated : 39
Cube of 39 = 59319
Random Integer generated : 76
Square of 76 = 5776
Random Integer generated : 5
Cube of 5 = 125
Random Integer generated : 18
Square of 18 = 324
Random Integer generated : 83
Cube of 83 = 571787

15. Write a Java program to iterate through all elements in a linked list

```
import java.util.LinkedList;
import java.util.Collections;
import java.util.Iterator;
public class NameList {
    public static void main(String[] args) {
        LinkedList<String> names = new LinkedList<>();
        names.add("Alice");
        names.add("Bob");
        names.add("Charlie");
        names.add("David");
        names.add("Eve");
        names.add("Fiona");
        names.add("George");
        names.add("Hannah");
        names.add("Ivy");
        names.add("Jack");
        System.out.println("Original order of
names:");
        Iterator<String> iterator =
names.iterator();
        while (iterator.hasNext()) {
            System.out.println(iterator.next());
        }
        Collections.sort(names);
```

```
        System.out.println("\nAlphabetical order of  
names:");  
        for (String name : names) {  
            System.out.println(name);  
        }  
    }  
}
```

Output

Original order of names:

Alice
Bob
Charlie
David
Eve
Fiona
George
Hannah
Ivy
Jack

Alphabetical order of names:

Alice
Bob
Charlie
David
Eve
Fiona
George
Hannah
Ivy
Jack

-
16. Develop an Applet program to accept two numbers from the user and output the sum and difference in the respective text boxes.

```
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;
```

```

public class SumDifferenceApplet extends Applet
implements ActionListener {
    TextField num1Field, num2Field, sumField,
diffField;
    Button calculateButton;

    public void init() {
        // Labels
        Label num1Label = new Label("Enter Number 1:");
        Label num2Label = new Label("Enter Number 2:");
        Label sumLabel = new Label("Sum:");
        Label diffLabel = new Label("Difference:");

        // Text Fields
        num1Field = new TextField(10);
        num2Field = new TextField(10);
        sumField = new TextField(10);
        diffField = new TextField(10);

        sumField.setEditable(false); // Output field
(readonly)
        diffField.setEditable(false); // Output field
(readonly)

        // Button
        calculateButton = new Button("Calculate");
        calculateButton.addActionListener(this);

        // Layout
        setLayout(new GridLayout(5, 2)); // 5 Rows, 2
Columns

        // Adding Components
        add(num1Label);
        add(num1Field);

```

```
        add(num2Label) ;
        add(num2Field) ;
        add(sumLabel) ;
        add(sumField) ;
        add(diffLabel) ;
        add(diffField) ;
        add(calculateButton) ;
    }

    public void actionPerformed(ActionEvent e) {
        try {
            int num1 =
Integer.parseInt(num1Field.getText()) ;
            int num2 =
Integer.parseInt(num2Field.getText()) ;

            int sum = num1 + num2 ;
            int diff = num1 - num2 ;

            sumField.setText(String.valueOf(sum)) ;
            diffField.setText(String.valueOf(diff)) ;
        } catch (NumberFormatException ex) {
            sumField.setText("Invalid Input") ;
            diffField.setText("Invalid Input") ;
        }
    }
}
```

Steps to Run the Applet

Save the file as SumDifferenceApplet.java

Compile the Java file

```
javac SumDifferenceApplet.java
```

Create an HTML file (applet.html)

```

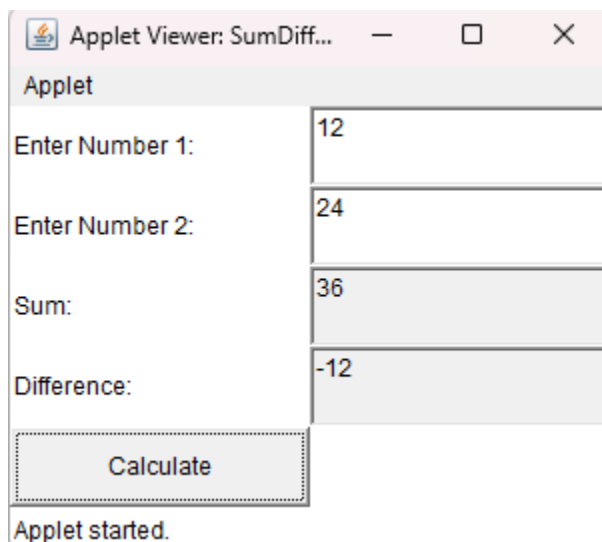
<html>
<body>

    <applet code="SumDifferenceApplet.class"
width="300" height="200"></applet>

</body>
</html>

```

Run using Applet Viewer
 appletviewer applet.html



17. Write a Java program to design a login page using Swing controls.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class UserInformationForm {

    public UserInformationForm() {
        JFrame frame = new JFrame("User Information
Form");
        frame.setLayout(new GridLayout(5, 2));

```

```
        frame.setSize(400, 200);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE
);

        // Name label and text field
        JLabel nameLabel = new JLabel("Name:");
        JTextField nameField = new JTextField();

        // Phone label and text field
        JLabel phoneLabel = new JLabel("Phone
Number:");
        JTextField phoneField = new JTextField();

        // Gender label and radio buttons
        JLabel genderLabel = new JLabel("Gender:");
        JRadioButton maleButton = new
JRadioButton("Male");
        JRadioButton femaleButton = new
JRadioButton("Female");

        // Grouping radio buttons
        ButtonGroup genderGroup = new
ButtonGroup();
        genderGroup.add(maleButton);
        genderGroup.add(femaleButton);

        // Submit and Reset buttons
        JButton submitButton = new
JButton("Submit");
        JButton resetButton = new JButton("Reset");

        // Adding components to the frame
        frame.add(nameLabel);
        frame.add(nameField);
        frame.add(phoneLabel);
        frame.add(phoneField);
        frame.add(genderLabel);
        frame.add(maleButton);
```

```

        frame.add(new JLabel("")); // Empty label
for spacing
        frame.add(femaleButton);
        frame.add(submitButton);
        frame.add(resetButton);

        // Submit button action
        submitButton.addActionListener(new
ActionListener() {
                public void
actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Information
Stored");
        }
    });

        // Reset button action
        resetButton.addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent
e) {

                    nameField.setText("");
                    phoneField.setText("");
                    genderGroup.clearSelection();

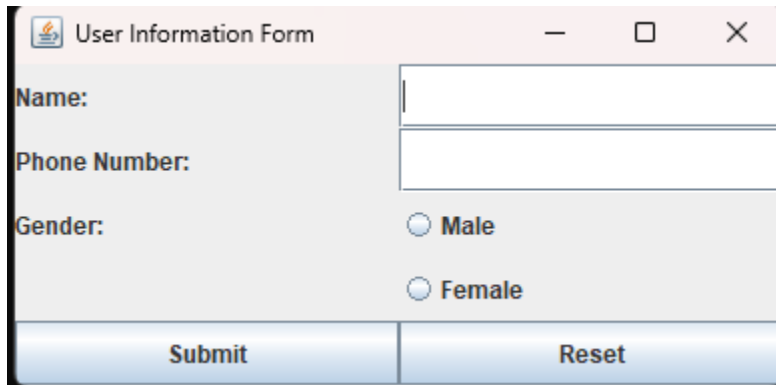
                }
            });

        frame.setVisible(true);
    }

    public static void main(String[] args) {
        UserInformationForm obj = new
UserInformationForm();
    }
}

```


Output



A screenshot of a Java Swing window titled "User Information Form". The window has a standard title bar with minimize, maximize, and close buttons. The form is divided into three sections: "Name:" with a text input field, "Phone Number:" with a text input field, and "Gender:" with two radio buttons labeled "Male" and "Female". At the bottom, there are two buttons: "Submit" and "Reset".

Name:	<input type="text"/>
Phone Number:	<input type="text"/>
Gender:	<input type="radio"/> Male <input type="radio"/> Female
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	
