**1. BankAccount Class with Custom Exceptions**

class LowBalanceException extends Exception {

public LowBalanceException(String message) {

super(message);

}

}

class NegativeNumberException extends Exception {

public NegativeNumberException(String message) {

super(message);

}

}

class BankAccount {

private double balance;

public BankAccount(double balance) {

this.balance = balance;

}

public void balanceEnquiry() {

System.out.println("Current Balance: " + balance);

}

public void deposit(double amount) throws NegativeNumberException {

if (amount < 0)

throw new NegativeNumberException("Amount cannot be negative");

balance += amount;

}

public void withdraw(double amount) throws LowBalanceException, NegativeNumberException {

if (amount < 0)

throw new NegativeNumberException("Amount cannot be negative");

if (amount > balance)

throw new LowBalanceException("Insufficient balance");

balance -= amount;

}

}

public class BankDemo {

public static void main(String[] args) {

BankAccount acc = new BankAccount(5000);

try {

acc.deposit(2000);

acc.withdraw(8000); // This will throw exception

} catch (NegativeNumberException | LowBalanceException e) {

System.out.println("Exception: " + e.getMessage());

}

acc.balanceEnquiry();

}

}

**2. Matrix Operations (2x2)**

public class MatrixOps {

public static void main(String[] args) {

int[][] a = {{1, 2}, {3, 4}};

int[][] b = {{5, 6}, {7, 8}};

int[][] result = new int[2][2];

// Addition

System.out.println("Addition:");

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

result[i][j] = a[i][j] + b[i][j];

System.out.print(result[i][j] + " ");

}

System.out.println();

}

// Multiplication

System.out.println("Multiplication:");

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

result[i][j] = 0;

for (int k = 0; k < 2; k++) {

result[i][j] += a[i][k] \* b[k][j];

}

System.out.print(result[i][j] + " ");

}

System.out.println();

}

// Transpose of A

System.out.println("Transpose of Matrix A:");

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

System.out.print(a[j][i] + " ");

}

System.out.println();

}

}

}

**3. Employee Class with Raise**

class Employee {

private String firstName;

private String lastName;

private double monthlySalary;

public Employee(String firstName, String lastName, double monthlySalary) {

this.firstName = firstName;

this.lastName = lastName;

setMonthlySalary(monthlySalary);

}

public void setFirstName(String firstName) { this.firstName = firstName; }

public void setLastName(String lastName) { this.lastName = lastName; }

public void setMonthlySalary(double salary) {

this.monthlySalary = salary > 0 ? salary : 0.0;

}

public String getFirstName() { return firstName; }

public String getLastName() { return lastName; }

public double getMonthlySalary() { return monthlySalary; }

public double getYearlySalary() { return monthlySalary \* 12; }

public void raiseSalary(double percent) {

monthlySalary += monthlySalary \* percent / 100;

}}

public class EmployeeTest {

public static void main(String[] args) {

Employee e1 = new Employee("John", "Doe", 3000);

Employee e2 = new Employee("Jane", "Smith", 4000);

System.out.println("Yearly Salary before raise:");

System.out.println(e1.getFirstName() + ": " + e1.getYearlySalary());

System.out.println(e2.getFirstName() + ": " + e2.getYearlySalary());

e1.raiseSalary(10);

e2.raiseSalary(10);

System.out.println("\nYearly Salary after 10% raise:");

System.out.println(e1.getFirstName() + ": " + e1.getYearlySalary());

System.out.println(e2.getFirstName() + ": " + e2.getYearlySalary());

}

}

**4. Stack Operations Using Class and Switch Case**

import java.util.Scanner;

class Stack {

int top = -1;

int[] arr = new int[5];

void push(int value) {

if (top == arr.length - 1)

System.out.println("Stack Overflow");

else

arr[++top] = value;

}

void pop() {

if (top == -1)

System.out.println("Stack Underflow");

else

System.out.println("Popped: " + arr[top--]);

}

void display() {

if (top == -1)

System.out.println("Stack is empty");

else {

System.out.println("Stack contents:");

for (int i = top; i >= 0; i--)

System.out.println(arr[i]);

}

}

}

public class StackDemo {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Stack s = new Stack();

int choice;

do {

System.out.println("\n1.Push 2.Pop 3.Display 4.Exit");

choice = sc.nextInt();

switch (choice) {

case 1:

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

s.push(sc.nextInt());

break;

case 2:

s.pop();

break;

case 3:

s.display();

break;

case 4:

break;

default:

System.out.println("Invalid choice!");

}

} while (choice != 4);

}

}

**5. File Handling with Character Streams**

import java.io.\*;

import java.util.Scanner;

public class FileHandler {

public static void main(String[] args) {

if (args.length == 0) {

System.out.println("Provide file name as command line argument.");

return;

}

String filename = args[0];

File file = new File(filename);

Scanner sc = new Scanner(System.in);

try {

if (file.exists()) {

BufferedReader br = new BufferedReader(new FileReader(file));

String line;

System.out.println("File Contents:");

while ((line = br.readLine()) != null)

System.out.println(line);

br.close();

System.out.print("Do you want to add data to the file? (Yes/No): ");

String response = sc.nextLine();

if (response.equalsIgnoreCase("Yes")) {

BufferedWriter bw = new BufferedWriter(new FileWriter(file, true));

String input;

System.out.println("Enter data (type 'exit' to stop):");

while (!(input = sc.nextLine()).equalsIgnoreCase("exit"))

bw.write(input + "\n");

bw.close();

}

} else {

System.out.println("File doesn't exist. Creating new file.");

BufferedWriter bw = new BufferedWriter(new FileWriter(file));

String input;

System.out.println("Enter data (type 'exit' to stop):");

while (!(input = sc.nextLine()).equalsIgnoreCase("exit"))

bw.write(input + "\n");

bw.close();

}

} catch (IOException e) {

System.out.println("Error: " + e.getMessage());

}

}

}

**6. Merge Two Sorted Arrays**

import java.util.Arrays;

public class MergeSortedArrays {

public static void main(String[] args) {

int[] a = {1, 3, 5, 7};

int[] b = {2, 4, 6, 8};

int[] c = new int[a.length + b.length];

int i = 0, j = 0, k = 0;

while (i < a.length && j < b.length) {

if (a[i] < b[j])

c[k++] = a[i++];

else

c[k++] = b[j++];

}

while (i < a.length) c[k++] = a[i++];

while (j < b.length) c[k++] = b[j++];

System.out.println(Arrays.toString(c));

}

}

**7a. Static Variable, Static Block & Static Method**

public class StaticExample {

static int count;

static {

count = 5;

}

static void display() {

System.out.println("Count: " + count);

}

public static void main(String[] args) {

StaticExample.display();

}

}

**7b. Count Vowels, Words, 'a' in a File**

import java.io.\*;

public class FileStats {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new FileReader("sample.txt"));

String line;

int vowels = 0, words = 0, aCount = 0;

while ((line = br.readLine()) != null) {

String[] w = line.split("\\s+");

words += w.length;

for (char c : line.toLowerCase().toCharArray()) {

if ("aeiou".indexOf(c) != -1) vowels++;

if (c == 'a') aCount++;

} }

System.out.println("Vowels: " + vowels);

System.out.println("Words: " + words);

System.out.println("Occurrences of 'a': " + aCount);

}}

**8. Interface Stack & IntegerStack Implementation**

interface Stack {

int size = 5;

void push(int x);

void pop();

void display();

void overflow();

void underflow();

}

class IntegerStack implements Stack {

int[] stack = new int[size];

int top = -1;

public void push(int x) {

if (top == size - 1) overflow();

else stack[++top] = x;

}

public void pop() {

if (top == -1) underflow();

else System.out.println("Popped: " + stack[top--]);

}

public void display() {

for (int i = top; i >= 0; i--) System.out.println(stack[i]);

}

public void overflow() {

System.out.println("Stack Overflow");

}

public void underflow() {

System.out.println("Stack Underflow");

}

}

public class StackTest {

public static void main(String[] args) {

IntegerStack s = new IntegerStack();

s.push(10);

s.push(20);

s.display();

s.pop();

s.pop();

s.pop();

}

}

**9. Interface Shape with Rectangle and Triangle**

interface Shape {

void area();

}

class Rectangle implements Shape {

int l = 5, b = 4;

public void area() {

System.out.println("Rectangle Area: " + (l \* b));

}

}

class Triangle implements Shape {

int b = 6, h = 3;

public void area() {

System.out.println("Triangle Area: " + (0.5 \* b \* h));

}

}

public class ShapeTest {

public static void main(String[] args) {

Shape r = new Rectangle();

Shape t = new Triangle();

r.area();

t.area();

}

}

**10. Student, Test, Sports, and Result Classes**

interface Sports {

int smarks = 20;

void set();

}

class Student {

int rollno;

void setrollno(int r) { rollno = r; }

int getrollno() { return rollno; }

}

class Test extends Student {

int sub1, sub2;

void setmarks(int a, int b) { sub1 = a; sub2 = b; }

void getmarks() {

System.out.println("Sub1: " + sub1 + ", Sub2: " + sub2);

}

}

class Result extends Test implements Sports {

int total;

public void set() {

total = sub1 + sub2 + smarks;

}

void display() {

System.out.println("Roll No: " + getrollno());

getmarks();

System.out.println("Sports Marks: " + smarks);

System.out.println("Total Marks: " + total);

}

}

public class ResultDemo {

public static void main(String[] args) {

Result r = new Result();

r.setrollno(101);

r.setmarks(70, 80);

r.set();

r.display();

}

}

**11. Standard Calculator using Swing**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Calculator extends JFrame implements ActionListener {

JTextField t;

double a, b, result;

String op;

public Calculator() {

setLayout(new BorderLayout());

t = new JTextField();

add(t, BorderLayout.NORTH);

JPanel p = new JPanel();

p.setLayout(new GridLayout(5, 4));

String[] buttons = {"7", "8", "9", "/", "4", "5", "6", "\*",

"1", "2", "3", "-", "0", ".", "=", "+",

"%", "√", "x²", "x³"};

for (String s : buttons) {

JButton b = new JButton(s);

b.addActionListener(this);

p.add(b);

}

add(p, BorderLayout.CENTER);

setSize(300, 400);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e) {

String s = e.getActionCommand();

if (s.matches("[0-9.]")) t.setText(t.getText() + s);

else if (s.equals("+") || s.equals("-") || s.equals("\*") || s.equals("/") || s.equals("%")) {

a = Double.parseDouble(t.getText());

op = s;

t.setText("");

} else if (s.equals("=")) {

b = Double.parseDouble(t.getText());

if (op.equals("+")) result = a + b;

if (op.equals("-")) result = a - b;

if (op.equals("\*")) result = a \* b;

if (op.equals("/")) result = a / b;

if (op.equals("%")) result = a % b;

t.setText("" + result);

} else if (s.equals("√")) {

a = Double.parseDouble(t.getText());

t.setText("" + Math.sqrt(a));

} else if (s.equals("x²")) {

a = Double.parseDouble(t.getText());

t.setText("" + (a \* a));

} else if (s.equals("x³")) {

a = Double.parseDouble(t.getText());

t.setText("" + (a \* a \* a));

}

}

public static void main(String[] args) {

new Calculator();

}

}

**14. Abstract Class Shape**

abstract class Shape {

double dim1, dim2;

abstract void area();

}

class Rectangle extends Shape {

Rectangle(double a, double b) { dim1 = a; dim2 = b; }

void area() { System.out.println("Rectangle Area: " + (dim1 \* dim2)); }

}

class Triangle extends Shape {

Triangle(double a, double b) { dim1 = a; dim2 = b; }

void area() { System.out.println("Triangle Area: " + (0.5 \* dim1 \* dim2)); }

}

public class ShapeTest {

public static void main(String[] args) {

Shape r = new Rectangle(10, 5);

Shape t = new Triangle(10, 5);

r.area();

t.area();

}

}

**15. Math Package with Subpackage**

**math/Stat.java**

package math;

public class Stat {

public static int factorial(int n) {

int f = 1;

for (int i = 1; i <= n; i++) f \*= i;

return f;

}

public static int cube(int n) {

return n \* n \* n;

}

}

**math/convert/Convert.java**

package math.convert;

public class Convert {

public static String toBinary(int n) { return Integer.toBinaryString(n); }

public static String toOctal(int n) { return Integer.toOctalString(n); }

public static String toHex(int n) { return Integer.toHexString(n); }

public static int fromBinary(String s) { return Integer.parseInt(s, 2); }

public static int fromOctal(String s) { return Integer.parseInt(s, 8); }

public static int fromHex(String s) { return Integer.parseInt(s, 16); }

}

**App.java**

import math.Stat;

import math.convert.Convert;

public class App {

public static void main(String[] args) {

System.out.println(Stat.factorial(5));

System.out.println(Stat.cube(3));

System.out.println(Convert.toBinary(10));

System.out.println(Convert.fromHex("A"));

}

}

**16. Employee Payroll using Packages: employee/Emp.java**

package employee;

public class Emp {

public String name;

public int empid;

public String category;

public double bpay, hra, da, npay, pf, grosspay, incometax, allowance;

public void calculate() {

hra = 0.09 \* bpay;

da = 0.05 \* bpay;

pf = 0.11 \* bpay;

allowance = 0.10 \* bpay;

grosspay = bpay + hra + da + allowance;

incometax = 0.1 \* grosspay;

npay = grosspay - pf - incometax;

}

public void display() {

System.out.println("Net Pay: " + npay);

}}

**Emppay.java**

import employee.Emp;

public class Emppay {

public static void main(String[] args) {

Emp e = new Emp();

e.name = "A";

e.empid = 101;

e.category = "Developer";

e.bpay = 30000;

e.calculate();

e.display();

}}

**17. Queue Operations with Class & Object**

import java.util.Scanner;

class Queue {

int[] q = new int[5];

int front = -1, rear = -1;

void insert(int x) {

if (rear == 4) System.out.println("Overflow");

else {

if (front == -1) front = 0;

q[++rear] = x;

}

}

void delete() {

if (front == -1 || front > rear) System.out.println("Underflow");

else System.out.println("Deleted: " + q[front++]);

}

void display() {

for (int i = front; i <= rear; i++) System.out.println(q[i]);

}

}

public class QueueDemo {

public static void main(String[] args) {

Queue q = new Queue();

Scanner s = new Scanner(System.in);

int ch;

do {

System.out.println("1.Insert 2.Delete 3.Display 4.Exit");

ch = s.nextInt();

switch (ch) {

case 1: q.insert(s.nextInt()); break;

case 2: q.delete(); break;

case 3: q.display(); break;

}

} while (ch != 4);

}

}

**20A. Change Background Color using Swing**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class ColorChanger extends JFrame implements ActionListener {

Container c;

public ColorChanger() {

c = getContentPane();

String[] colors = {"Red", "Green", "Blue"};

JComboBox<String> cb = new JComboBox<>(colors);

cb.addActionListener(this);

add(cb);

setSize(300, 200);

setLayout(new FlowLayout());

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e) {

JComboBox cb = (JComboBox) e.getSource();

String color = (String) cb.getSelectedItem();

if (color.equals("Red")) c.setBackground(Color.RED);

else if (color.equals("Green")) c.setBackground(Color.GREEN);

else c.setBackground(Color.BLUE);

}

public static void main(String[] args) {

new ColorChanger();

}

}

**20B. Prime Check using Swing**

import javax.swing.\*;

import java.awt.event.\*;

public class PrimeCheck extends JFrame {

JTextField t;

JLabel l;

public PrimeCheck() {

t = new JTextField(10);

JButton b = new JButton("Check");

l = new JLabel();

add(t);

add(b);

add(l);

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int n = Integer.parseInt(t.getText());

boolean prime = true;

if (n < 2) prime = false;

for (int i = 2; i <= n / 2; i++)

if (n % i == 0) prime = false;

l.setText(prime ? "Prime" : "Not Prime");

}

});

setLayout(new java.awt.FlowLayout());

setSize(200, 150);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public static void main(String[] args) {

new PrimeCheck();

}

}

**21. Ecommerce Package**

**ecommerce/Product.java**

package ecommerce;

public class Product {

public String name;

public double price;

public Product(String name, double price) {

this.name = name;

this.price = price;

}

public void display() {

System.out.println("Product: " + name + " Price: " + price);

}

}

**ecommerce/Customer.java**

package ecommerce;

public class Customer {

public String name;

public Customer(String name) {

this.name = name;

}

}

**ecommerce/Order.java**

package ecommerce;

public class Order {

public Product product;

public Customer customer;

public int quantity;

public Order(Product product, Customer customer, int quantity) {

this.product = product;

this.customer = customer;

this.quantity = quantity;

}

public double totalCost() {

return product.price \* quantity;

}

public void placeOrder() {

System.out.println("Order placed by " + customer.name);

product.display();

System.out.println("Quantity: " + quantity);

System.out.println("Total Cost: " + totalCost());

}

}

**EcommerceApp.java**

import ecommerce.\*;

public class EcommerceApp {

public static void main(String[] args) {

Product p = new Product("Phone", 20000);

Customer c = new Customer("Shravani");

Order o = new Order(p, c, 2);

o.placeOrder();

}

}

**22. LibraryManagement Package**

**library/Book.java**

package library;

public class Book {

public String title, author, isbn;

public Book(String title, String author, String isbn) {

this.title = title;

this.author = author;

this.isbn = isbn;

}

public void display() {

System.out.println("Book: " + title + " Author: " + author + " ISBN: " + isbn);

}

}

**library/Member.java**

package library;

public class Member {

public String name;

public int id;

public Member(String name, int id) {

this.name = name;

this.id = id;

}

public void display() {

System.out.println("Member: " + name + " ID: " + id);

}

}

**LibraryApp.java**

import library.\*;

public class LibraryApp {

public static void main(String[] args) {

Book b = new Book("Java", "James", "12345");

Member m = new Member("Shravani", 1);

b.display();

m.display();

}

}

**23. Simple Chat Application Using Socket Programming**

**ChatServer.java**

import java.net.\*;

import java.io.\*;

public class ChatServer {

public static void main(String[] args) throws Exception {

ServerSocket ss = new ServerSocket(1234);

Socket s = ss.accept();

BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

PrintWriter out = new PrintWriter(s.getOutputStream(), true);

BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));

String msg;

while (!(msg = in.readLine()).equals("exit")) {

System.out.println("Client: " + msg);

out.println(stdin.readLine());

}

s.close();

ss.close();

}

}

**ChatClient.java**

import java.net.\*;

import java.io.\*;

public class ChatClient {

public static void main(String[] args) throws Exception {

Socket s = new Socket("localhost", 1234);

BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

PrintWriter out = new PrintWriter(s.getOutputStream(), true);

BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));

String msg;

while (true) {

out.println(stdin.readLine());

msg = in.readLine();

System.out.println("Server: " + msg);

if (msg.equals("exit")) break;

}

s.close();

}

}

**25. BankAccount and SavingsAccount**

class BankAccount {

double balance = 0;

void deposit(double amount) {

balance += amount;

System.out.println("Deposited: " + amount);

}

void withdraw(double amount) {

balance -= amount;

System.out.println("Withdrawn: " + amount);

}

}

class SavingsAccount extends BankAccount {

void withdraw(double amount) {

if (balance - amount < 100) {

System.out.println("Withdrawal denied. Balance cannot go below 100");

} else {

balance -= amount;

System.out.println("Withdrawn: " + amount);

}

}

public static void main(String[] args) {

SavingsAccount s = new SavingsAccount();

s.deposit(500);

s.withdraw(450);

s.withdraw(50);

}

}

**26A. Student Info Using Data Streams**

import java.io.\*;

import java.util.Scanner;

public class StudentData {

public static void main(String[] args) throws Exception {

Scanner sc = new Scanner(System.in);

FileOutputStream fos = new FileOutputStream("student.dat");

DataOutputStream dos = new DataOutputStream(fos);

System.out.print("Name: ");

dos.writeUTF(sc.nextLine());

System.out.print("Age: ");

dos.writeInt(sc.nextInt());

System.out.print("Weight: ");

dos.writeDouble(sc.nextDouble());

System.out.print("Height: ");

dos.writeDouble(sc.nextDouble());

sc.nextLine();

System.out.print("City: ");

dos.writeUTF(sc.nextLine());

System.out.print("Phone: ");

dos.writeUTF(sc.nextLine());

dos.close();

FileInputStream fis = new FileInputStream("student.dat");

DataInputStream dis = new DataInputStream(fis);

System.out.println("Name: " + dis.readUTF());

System.out.println("Age: " + dis.readInt());

System.out.println("Weight: " + dis.readDouble());

System.out.println("Height: " + dis.readDouble());

System.out.println("City: " + dis.readUTF());

System.out.println("Phone: " + dis.readUTF());

dis.close();

}

}

**26B. Arrange 10 Names Alphabetically**

import java.util.\*;

public class SortNames {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String[] names = new String[10];

for (int i = 0; i < 10; i++) {

names[i] = sc.nextLine();

}

Arrays.sort(names);

for (String name : names) {

System.out.println(name);

}

}

}

**27.**

**1. OddNumberException**

class OddNumberException extends Exception {

OddNumberException(String msg) {

super(msg);

}

}

public class CheckOdd {

static void checkEven(int n) throws OddNumberException {

if (n % 2 != 0) throw new OddNumberException("Number is odd");

System.out.println("Number is even");

}

public static void main(String[] args) {

try {

checkEven(5);

} catch (OddNumberException e) {

System.out.println(e.getMessage());

}

}

}

**2. Display Contents of sample.txt**

import java.io.\*;

public class DisplayFile {

public static void main(String[] args) throws Exception {

BufferedReader br = new BufferedReader(new FileReader("sample.txt"));

String line;

while ((line = br.readLine()) != null) {

System.out.println(line);

} }

}

**28.**

**1. MathOperations Package**

**mathops/MathFunctions.java**

package mathops;

public class MathFunctions {

public static double floor(double n) {

return Math.floor(n);

}

public static double ceil(double n) {

return Math.ceil(n);

}

public static long round(double n) {

return Math.round(n);

}

}

**MathApp.java**

import mathops.MathFunctions;

public class MathApp {

public static void main(String[] args) {

System.out.println(MathFunctions.floor(4.7));

System.out.println(MathFunctions.ceil(4.2));

System.out.println(MathFunctions.round(4.5));

}

}

**2. Simple Timer with Swing**

import javax.swing.\*;

import java.awt.event.\*;

public class TimerApp {

public static void main(String[] args) {

JFrame f = new JFrame();

JLabel label = new JLabel("0");

JButton start = new JButton("Start");

JButton stop = new JButton("Stop");

label.setBounds(100, 50, 100, 30);

start.setBounds(50, 100, 80, 30);

stop.setBounds(150, 100, 80, 30);

f.add(label);

f.add(start);

f.add(stop);

f.setSize(300, 200);

f.setLayout(null);

f.setVisible(true);

Timer t = new Timer(1000, new ActionListener() {

int count = 0;

public void actionPerformed(ActionEvent e) {

count++;

label.setText(String.valueOf(count));

}

});

start.addActionListener(e -> t.start());

stop.addActionListener(e -> t.stop());

}

}

**12A. Counter GUI**

import javax.swing.\*;

import java.awt.event.\*;

public class CounterApp {

public static void main(String[] args) {

JFrame f = new JFrame("Counter");

JLabel l = new JLabel("Counter");

JTextField t = new JTextField("0");

JButton up = new JButton("Count Up");

JButton down = new JButton("Count Down");

JButton reset = new JButton("Reset");

l.setBounds(20, 20, 60, 30);

t.setBounds(90, 20, 80, 30);

up.setBounds(180, 20, 100, 30);

down.setBounds(290, 20, 120, 30);

reset.setBounds(420, 20, 80, 30);

f.add(l);

f.add(t);

f.add(up);

f.add(down);

f.add(reset);

up.addActionListener(e -> t.setText(String.valueOf(Integer.parseInt(t.getText()) + 1)));

down.addActionListener(e -> t.setText(String.valueOf(Integer.parseInt(t.getText()) - 1)));

reset.addActionListener(e -> t.setText("0"));

f.setSize(550, 100);

f.setLayout(null);

f.setVisible(true);

}

}

**12B. JTextArea Character and Word Count with KeyListener**

import javax.swing.\*;

import java.awt.event.\*;

public class TextStats {

public static void main(String[] args) {

JFrame f = new JFrame("Text Counter");

JTextArea area = new JTextArea();

JLabel label = new JLabel("Chars: 0 Words: 0");

area.setBounds(20, 20, 250, 100);

label.setBounds(20, 130, 250, 30);

area.addKeyListener(new KeyAdapter() {

public void keyReleased(KeyEvent e) {

String text = area.getText();

int chars = text.length();

int words = text.trim().isEmpty() ? 0 : text.trim().split("\\s+").length;

label.setText("Chars: " + chars + " Words: " + words);

}

});

f.add(area);

f.add(label);

f.setSize(300, 200);

f.setLayout(null);

f.setVisible(true);

}

}

**13 BorderLayout Binary, Octal, Hex Converter**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class BorderConverter {

public static void main(String[] args) {

JFrame f = new JFrame("Converter");

f.setLayout(new BorderLayout());

JPanel top = new JPanel();

JLabel l = new JLabel("Enter the number");

JTextField t = new JTextField(10);

top.add(l);

top.add(t);

JPanel center = new JPanel(new GridLayout(1, 3));

JButton b1 = new JButton("Binary");

JButton b2 = new JButton("Octal");

JButton b3 = new JButton("Hex");

center.add(b1);

center.add(b2);

center.add(b3);

JPanel bottom = new JPanel();

JLabel result = new JLabel("Result");

JTextField r = new JTextField(15);

bottom.add(result);

bottom.add(r);

f.add(top, BorderLayout.NORTH);

f.add(center, BorderLayout.CENTER);

f.add(bottom, BorderLayout.SOUTH);

b1.addActionListener(e -> r.setText(Integer.toBinaryString(Integer.parseInt(t.getText()))));

b2.addActionListener(e -> r.setText(Integer.toOctalString(Integer.parseInt(t.getText()))));

b3.addActionListener(e -> r.setText(Integer.toHexString(Integer.parseInt(t.getText()))));

f.setSize(400, 200);

f.setVisible(true);

}

}

**19.Flowlayout and gridlayout**

import javax.swing.\*;

import java.awt.\*;

public class SimpleGUI {

public static void main(String[] args) {

JFrame frame1 = new JFrame("FlowLayout Example");

frame1.setLayout(new FlowLayout(FlowLayout.LEFT, 10, 20));

frame1.add(new JCheckBox("Java"));

frame1.add(new JCheckBox("Python"));

frame1.add(new JCheckBox("C++"));

frame1.setSize(300, 150);

frame1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame1.setVisible(true);

JFrame frame2 = new JFrame("GridLayout Example");

frame2.setLayout(new GridLayout(2, 3));

for (int i = 1; i <= 6; i++) frame2.add(new JButton(String.valueOf(i)));

frame2.setSize(300, 150);

frame2.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame2.setVisible(true);

}

}

**18 stationary purchase system**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class StationaryPurchaseSystem extends JFrame {

private JCheckBox notebookCheckBox, penCheckBox, pencilCheckBox;

private JButton orderButton;

public StationaryPurchaseSystem() {

setTitle("Stationary Purchase System");

setSize(300, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

notebookCheckBox = new JCheckBox("Notebook @ 50");

notebookCheckBox.setBounds(20, 20, 150, 20);

add(notebookCheckBox);

penCheckBox = new JCheckBox("Pen @ 30");

penCheckBox.setBounds(20, 50, 150, 20);

add(penCheckBox);

pencilCheckBox = new JCheckBox("Pencil @ 10");

pencilCheckBox.setBounds(20, 80, 150, 20);

add(pencilCheckBox);

orderButton = new JButton("Order");

orderButton.setBounds(20, 110, 80, 30);

add(orderButton);

orderButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (notebookCheckBox.isSelected()) {

int quantity = Integer.parseInt(JOptionPane.showInputDialog("Enter Quantity for Notebook"));

int total = quantity \* 50;

JOptionPane.showMessageDialog(null, "Notebook Quantity: " + quantity + " Total: " + total);

}

JOptionPane.showMessageDialog(null, "Successfully Ordered.");

}

});

}

public static void main(String[] args) {

new StationaryPurchaseSystem().setVisible(true);

}

}

**24 A. Multilevel inheritance program:**

class Student {

int rollNo;

Student(int rollNo) {

this.rollNo = rollNo;

}

}

class Test extends Student {

int sub1, sub2;

Test(int rollNo, int sub1, int sub2) {

super(rollNo);

this.sub1 = sub1;

this.sub2 = sub2;

}

}

class Result extends Test {

Result(int rollNo, int sub1, int sub2) {

super(rollNo, sub1, sub2);

}

void display() {

System.out.println("Roll No: " + rollNo);

System.out.println("Subject 1: " + sub1);

System.out.println("Subject 2: " + sub2);

System.out.println("Result: " + (sub1 + sub2));

}

public static void main(String[] args) {

Result r = new Result(101, 85, 90);

r.display();

}

}

**24 B. Java program to display number of words, lines, vowels from a text file:**

import java.io.\*;

public class FileStats {

public static void main(String[] args) throws Exception {

BufferedReader br = new BufferedReader(new FileReader("input.txt"));

String line;

int lines = 0, words = 0, vowels = 0;

while ((line = br.readLine()) != null) {

lines++;

String[] w = line.split("\\s+");

words += w.length;

for (char c : line.toLowerCase().toCharArray()) {

if ("aeiou".indexOf(c) != -1) vowels++;

}

}

br.close();

System.out.println("Lines: " + lines);

System.out.println("Words: " + words);

System.out.println("Vowels: " + vowels);

}

}