JAVA PROJECT FILE

Q1. write a java program that prints all real solutions to the quadratic equation ax^2 + bx + c = 0. Read in a, b, c and use quadratic formula. If the discriminate b^2 - 4ac is negative, display a message stating that there are no real solutions.

Ans. import java.util.Scanner;

public class QuadraticSolver {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Read coefficients a, b, and c from the user

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

double a = scanner.nextDouble();

System.out.print("Enter coefficient b: ");

double b = scanner.nextDouble();

System.out.print("Enter coefficient c: ");

double c = scanner.nextDouble();

// Calculate the discriminant

double discriminant = b \* b - 4 \* a \* c;

// Determine the number and type of solutions

if (a == 0) {

if (b != 0) {

double x = -c / b;

System.out.println("Linear solution: x = " + x);

} else {

System.out.println("This is not a valid equation.");

}

} else if (discriminant > 0) {

double root1 = (-b + Math.sqrt(discriminant)) / (2 \* a);

double root2 = (-b - Math.sqrt(discriminant)) / (2 \* a);

System.out.println("Two real solutions: x = " + root1 + " and x = " + root2);

} else if (discriminant == 0) {

double root = -b / (2 \* a);

System.out.println("One real solution: x = " + root);

} else {

System.out.println("No real solutions.");

}

scanner.close();

}

}

Q2. The fibonacci sequence is defined by the following rule: the first two values in the sequence are 1 and 1. Every subsequent value is the sum of two values preceding it. Write a java progrm that prints the nth value in fibnocci sequence.

Ans: import java.util.Scanner;

public class Fibonacci {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt user to enter the position n

System.out.print("Enter the position (n) in the Fibonacci sequence: ");

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Please enter a positive integer.");

} else {

int result = fibonacci(n);

System.out.println("The " + n + "th value in the Fibonacci sequence is: " + result);

}

scanner.close();

}

// Method to compute the nth Fibonacci number

public static int fibonacci(int n) {

if (n == 1 || n == 2) {

return 1;

}

int prev = 1, curr = 1;

for (int i = 3; i <= n; i++) {

int next = prev + curr;

prev = curr;

curr = next;

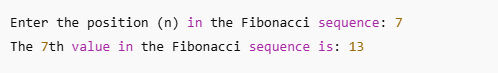
}

return curr;

}

}

Output:



Q3. Write a java program that prompts the user for an integer and then prints out all prime numbers up to that integer.

Ans: import java.util.Scanner;

public class PrimeNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt user for an integer

System.out.print("Enter an integer: ");

int max = scanner.nextInt();

System.out.println("Prime numbers up to " + max + " are:");

for (int num = 2; num <= max; num++) {

if (isPrime(num)) {

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

}

}

scanner.close();

}

// Helper method to check if a number is prime

public static boolean isPrime(int n) {

if (n <= 1) return false;

for (int i = 2; i <= Math.sqrt(n); i++) {

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

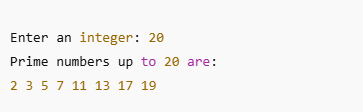
}

return true;

}

}

Output:



Q4. Write a java program that multiply two given matrices.

Ans: import java.util.Scanner;

public class MatrixMultiplication {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input dimensions of matrix A

System.out.print("Enter number of rows in Matrix A: ");

int rowsA = scanner.nextInt();

System.out.print("Enter number of columns in Matrix A: ");

int colsA = scanner.nextInt();

// Input dimensions of matrix B

System.out.print("Enter number of rows in Matrix B: ");

int rowsB = scanner.nextInt();

System.out.print("Enter number of columns in Matrix B: ");

int colsB = scanner.nextInt();

// Check if multiplication is possible

if (colsA != rowsB) {

System.out.println("Matrix multiplication is not possible. Columns of A must equal rows of B.");

return;

}

// Input Matrix A

int[][] A = new int[rowsA][colsA];

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

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

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

A[i][j] = scanner.nextInt();

}

}

// Input Matrix B

int[][] B = new int[rowsB][colsB];

System.out.println("Enter elements of Matrix B:");

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

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

B[i][j] = scanner.nextInt();

}

}

// Multiply matrices

int[][] result = new int[rowsA][colsB];

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

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

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

result[i][j] += A[i][k] \* B[k][j];

}

}

}

// Print the result

System.out.println("Result of Matrix Multiplication:");

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

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

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

}

System.out.println();

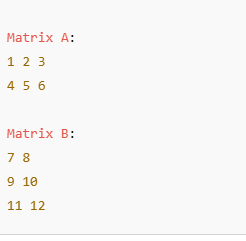
}

scanner.close();

}

}

Output:



Q5. Write a java program that reads a line of integers , and then display each integer , and the sum of all integers .

Ans: import java.util.Scanner;

public class SumOfIntegers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a line of integers

System.out.print("Enter a line of integers separated by spaces: ");

String line = scanner.nextLine();

// Split the input line into individual strings

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

int sum = 0;

System.out.println("You entered:");

// Parse and display each integer, and calculate the sum

for (String numStr : numbers) {

int num = Integer.parseInt(numStr);

System.out.println(num);

sum += num;

}

// Print the total sum

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

scanner.close();

}

}

Output:



Q6. Write a java program that checks whether a given string is a palindrome or not.

Ans: import java.util.Scanner;

public class PalindromeChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user for a string

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

String input = scanner.nextLine();

// Clean the string (optional: remove spaces and make it lowercase)

String cleaned = input.replaceAll("\\s+", "").toLowerCase();

// Check if it's a palindrome

boolean isPalindrome = true;

int length = cleaned.length();

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

if (cleaned.charAt(i) != cleaned.charAt(length - 1 - i)) {

isPalindrome = false;

break;

}

}

// Output the result

if (isPalindrome) {

System.out.println("The string is a palindrome.");

} else {

System.out.println("The string is not a palindrome.");

}

scanner.close();

}

}

Output:



Q7. Write a java program for sorting list of names. Read input from command line.

Ans: import java.util.Arrays;

public class SortNames {

public static void main(String[] args) {

if (args.length == 0) {

System.out.println("Please provide names as command line arguments.");

return;

}

// Sort the names

Arrays.sort(args, String.CASE\_INSENSITIVE\_ORDER);

// Print the sorted names

System.out.println("Sorted names:");

for (String name : args) {

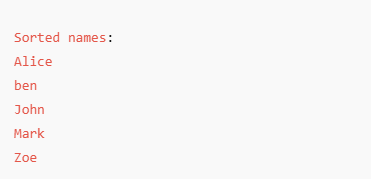
System.out.println(name);

}

}

}

Output:



Q8. Write a java program to create a student class with following fields a. Hall ticket number b. Student name c. Department Create 'n' number of student object where 'n' value is passed as input to constructor.

Ans: import java.util.Scanner;

// Student class definition

class Student {

String hallTicketNumber;

String name;

String department;

// Constructor

Student(String hallTicketNumber, String name, String department) {

this.hallTicketNumber = hallTicketNumber;

this.name = name;

this.department = department;

}

// Display method

void display() {

System.out.println("Hall Ticket No: " + hallTicketNumber);

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

System.out.println("Department : " + department);

System.out.println("------------------------------");

}

}

// Main class

public class StudentRecords {

Student[] students;

// Constructor to create n students

public StudentRecords(int n) {

students = new Student[n];

Scanner scanner = new Scanner(System.in);

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

System.out.println("Enter details for Student " + (i + 1) + ":");

System.out.print("Hall Ticket Number: ");

String hallTicket = scanner.nextLine();

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

String name = scanner.nextLine();

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

String dept = scanner.nextLine();

students[i] = new Student(hallTicket, name, dept);

System.out.println();

}

scanner.close();

}

// Method to display all students

public void displayAll() {

System.out.println("----- Student Details -----");

for (Student s : students) {

s.display();

}

}

// Main method

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of students: ");

int n = scanner.nextInt();

scanner.nextLine(); // Consume the leftover newline

StudentRecords records = new StudentRecords(n);

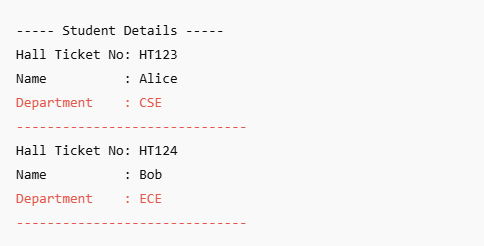
records.displayAll();

scanner.close();

}

}

Output:



Q9. Write a java program to demostrate string comparison using == and equals method.

Ans: public class StringComparisonDemo {

public static void main(String[] args) {

// Create strings in different ways

String str1 = "Hello";

String str2 = "Hello"; // Same literal - points to the same object

String str3 = new String("Hello"); // New object created in memory

// Comparing with ==

System.out.println("Using == operator:");

System.out.println("str1 == str2: " + (str1 == str2)); // true (same object in string pool)

System.out.println("str1 == str3: " + (str1 == str3)); // false (different objects)

// Comparing with equals()

System.out.println("\nUsing equals() method:");

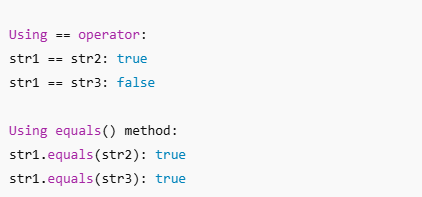
System.out.println("str1.equals(str2): " + str1.equals(str2)); // true (same content)

System.out.println("str1.equals(str3): " + str1.equals(str3)); // true (same content)

}

}

Output:



Q10. Write a java program to create an abstract class named shape that contains an empty method named number of sides(). Provide three classes named Trapeziod, triangle and hexagon such that each one of the classes extends the class shape. each one of the classes contains only the method number of slides() that shows the number of sides in the geometrical figures.

Ans: // Abstract class Shape

abstract class Shape {

// Abstract method

abstract void numberOfSides();

}

// Trapezoid class

class Trapezoid extends Shape {

@Override

void numberOfSides() {

System.out.println("A Trapezoid has 4 sides.");

}

}

// Triangle class

class Triangle extends Shape {

@Override

void numberOfSides() {

System.out.println("A Triangle has 3 sides.");

}

}

// Hexagon class

class Hexagon extends Shape {

@Override

void numberOfSides() {

System.out.println("A Hexagon has 6 sides.");

}

}

// Main class to test

public class ShapeDemo {

public static void main(String[] args) {

// Create objects of each shape

Shape trapezoid = new Trapezoid();

Shape triangle = new Triangle();

Shape hexagon = new Hexagon();

// Call the method on each object

trapezoid.numberOfSides();

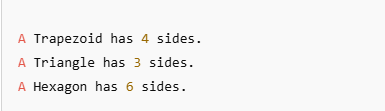
triangle.numberOfSides();

hexagon.numberOfSides();

}

}

Output:



Q11. Write a java program that creates threee threads. first thread displays " Good Morning" everyone second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.

Ans: class GoodMorningThread extends Thread {

public void run() {

while (true) {

System.out.println("Good Morning");

try {

Thread.sleep(1000); // 1 second

} catch (InterruptedException e) {

System.out.println("Good Morning thread interrupted.");

}

}

}

}

class HelloThread extends Thread {

public void run() {

while (true) {

System.out.println("Hello");

try {

Thread.sleep(2000); // 2 seconds

} catch (InterruptedException e) {

System.out.println("Hello thread interrupted.");

}

}

}

}

class WelcomeThread extends Thread {

public void run() {

while (true) {

System.out.println("Welcome");

try {

Thread.sleep(3000); // 3 seconds

} catch (InterruptedException e) {

System.out.println("Welcome thread interrupted.");

}

}

}

}

public class MultiThreadDemo {

public static void main(String[] args) {

// Create and start threads

new GoodMorningThread().start();

new HelloThread().start();

new WelcomeThread().start();

}

}

Output:



Q12. Write a java program that correctly implements producer consumer problem using the concept of inter thread communication.

Ans: class SharedBuffer {

private int data;

private boolean isProduced = false;

// Producer puts data into the buffer

public synchronized void produce(int value) {

while (isProduced) {

try {

wait(); // wait until consumer consumes

} catch (InterruptedException e) {

e.printStackTrace();

}

}

data = value;

isProduced = true;

System.out.println("Produced: " + data);

notify(); // notify the consumer

}

// Consumer gets data from the buffer

public synchronized void consume() {

while (!isProduced) {

try {

wait(); // wait until producer produces

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("Consumed: " + data);

isProduced = false;

notify(); // notify the producer

}

}

// Producer thread

class Producer extends Thread {

SharedBuffer buffer;

Producer(SharedBuffer buffer) {

this.buffer = buffer;

}

public void run() {

int i = 1;

while (true) {

buffer.produce(i++);

try {

Thread.sleep(1000); // simulate time to produce

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

// Consumer thread

class Consumer extends Thread {

SharedBuffer buffer;

Consumer(SharedBuffer buffer) {

this.buffer = buffer;

}

public void run() {

while (true) {

buffer.consume();

try {

Thread.sleep(1500); // simulate time to consume

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

// Main class

public class ProducerConsumerDemo {

public static void main(String[] args) {

SharedBuffer buffer = new SharedBuffer();

Producer producer = new Producer(buffer);

Consumer consumer = new Consumer(buffer);

producer.start();

consumer.start();

}

}

Output:



Q13. Write a GUI java program that stimulates a traffic light. The program lets the user select one of three lights: red , yellow , or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when program starts.

Ans: import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class TrafficLightSimulator extends JFrame implements ActionListener {

JRadioButton redButton, yellowButton, greenButton;

ButtonGroup group;

LightPanel lightPanel;

public TrafficLightSimulator() {

setTitle("Traffic Light Simulator");

setSize(300, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

// Create radio buttons

redButton = new JRadioButton("Red");

yellowButton = new JRadioButton("Yellow");

greenButton = new JRadioButton("Green");

// Group buttons so only one can be selected

group = new ButtonGroup();

group.add(redButton);

group.add(yellowButton);

group.add(greenButton);

// Add action listeners

redButton.addActionListener(this);

yellowButton.addActionListener(this);

greenButton.addActionListener(this);

// Panel for radio buttons

JPanel buttonPanel = new JPanel(new FlowLayout());

buttonPanel.add(redButton);

buttonPanel.add(yellowButton);

buttonPanel.add(greenButton);

// Panel for drawing traffic light

lightPanel = new LightPanel();

add(buttonPanel, BorderLayout.SOUTH);

add(lightPanel, BorderLayout.CENTER);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

if (redButton.isSelected()) {

lightPanel.setLight("RED");

} else if (yellowButton.isSelected()) {

lightPanel.setLight("YELLOW");

} else if (greenButton.isSelected()) {

lightPanel.setLight("GREEN");

}

}

// Inner class for drawing lights

class LightPanel extends JPanel {

private String currentLight = "";

public void setLight(String light) {

currentLight = light;

repaint(); // Request panel repaint

}

public void paintComponent(Graphics g) {

super.paintComponent(g);

// Background

g.setColor(Color.LIGHT\_GRAY);

g.fillRect(90, 30, 100, 250);

// Draw circles

g.setColor(currentLight.equals("RED") ? Color.RED : Color.GRAY);

g.fillOval(110, 50, 60, 60);

g.setColor(currentLight.equals("YELLOW") ? Color.YELLOW : Color.GRAY);

g.fillOval(110, 120, 60, 60);

g.setColor(currentLight.equals("GREEN") ? Color.GREEN : Color.GRAY);

g.fillOval(110, 190, 60, 60);

}

}

public static void main(String[] args) {

new TrafficLightSimulator();

}

}

Q14. Write a java program that allows the user to draw lines, rectangles and ovals.

Ans: import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class DrawingShapes extends JFrame {

private String currentShape = "";

private int startX, startY, endX, endY;

public DrawingShapes() {

setTitle("Draw Shapes");

setSize(600, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

DrawingPanel drawPanel = new DrawingPanel();

add(drawPanel, BorderLayout.CENTER);

JPanel buttonPanel = new JPanel();

JButton lineBtn = new JButton("Line");

JButton rectBtn = new JButton("Rectangle");

JButton ovalBtn = new JButton("Oval");

lineBtn.addActionListener(e -> currentShape = "Line");

rectBtn.addActionListener(e -> currentShape = "Rectangle");

ovalBtn.addActionListener(e -> currentShape = "Oval");

buttonPanel.add(lineBtn);

buttonPanel.add(rectBtn);

buttonPanel.add(ovalBtn);

add(buttonPanel, BorderLayout.NORTH);

setVisible(true);

}

class DrawingPanel extends JPanel {

public DrawingPanel() {

setBackground(Color.WHITE);

addMouseListener(new MouseAdapter() {

public void mousePressed(MouseEvent e) {

startX = e.getX();

startY = e.getY();

}

public void mouseReleased(MouseEvent e) {

endX = e.getX();

endY = e.getY();

repaint();

}

});

}

protected void paintComponent(Graphics g) {

super.paintComponent(g);

if (currentShape == null || currentShape.isEmpty()) return;

int x = Math.min(startX, endX);

int y = Math.min(startY, endY);

int width = Math.abs(endX - startX);

int height = Math.abs(endY - startY);

switch (currentShape) {

case "Line":

g.drawLine(startX, startY, endX, endY);

break;

case "Rectangle":

g.drawRect(x, y, width, height);

break;

case "Oval":

g.drawOval(x, y, width, height);

break;

}

}

}

public static void main(String[] args) {

new DrawingShapes();

}

}

Q15. Develop an GUI application for calculator.

Ans: import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class CalculatorGUI extends JFrame implements ActionListener {

private JTextField display;

private double num1 = 0, num2 = 0;

private String operator = "";

public CalculatorGUI() {

setTitle("Calculator");

setSize(300, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Display field

display = new JTextField();

display.setEditable(false);

display.setFont(new Font("Arial", Font.BOLD, 24));

add(display, BorderLayout.NORTH);

// Button panel

JPanel buttonPanel = new JPanel();

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

// Buttons

String[] buttons = {

"7", "8", "9", "/",

"4", "5", "6", "\*",

"1", "2", "3", "-",

"0", "C", "=", "+"

};

for (String text : buttons) {

JButton btn = new JButton(text);

btn.setFont(new Font("Arial", Font.PLAIN, 20));

btn.addActionListener(this);

buttonPanel.add(btn);

}

add(buttonPanel, BorderLayout.CENTER);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

String input = e.getActionCommand();

if (input.matches("[0-9]")) {

display.setText(display.getText() + input);

} else if (input.matches("[+\\-\*/]")) {

if (!display.getText().isEmpty()) {

num1 = Double.parseDouble(display.getText());

operator = input;

display.setText("");

}

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

if (!display.getText().isEmpty()) {

num2 = Double.parseDouble(display.getText());

double result = 0;

switch (operator) {

case "+": result = num1 + num2; break;

case "-": result = num1 - num2; break;

case "\*": result = num1 \* num2; break;

case "/":

if (num2 != 0) result = num1 / num2;

else {

display.setText("Error");

return;

}

break;

}

display.setText(String.valueOf(result));

operator = "";

}

} else if (input.equals("C")) {

display.setText("");

num1 = num2 = 0;

operator = "";

}

}

public static void main(String[] args) {

new CalculatorGUI();

}

}