1. Write a program that calculates the total sales and commission for a group of salespeople.

import java.util.Scanner;

public class SalesCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Sales and Commission Calculator");

// Input the number of salespeople

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

int numSalespeople = scanner.nextInt();

// Input the commission rate

System.out.print("Enter the commission rate (in %): ");

double commissionRate = scanner.nextDouble();

// Initialize total sales and total commission

double totalSales = 0;

double totalCommission = 0;

// Loop through each salesperson to collect sales data

for (int i = 1; i <= numSalespeople; i++) {

System.out.println("\nSalesperson " + i + ":");

System.out.print("Enter the sales amount for salesperson " + i + ": ");

double sales = scanner.nextDouble();

// Calculate commission

double commission = sales \* (commissionRate / 100);

// Update totals

totalSales += sales;

totalCommission += commission;

// Display individual commission

System.out.printf("Commission for salesperson %d: $%.2f%n", i, commission);

}

// Display total sales and total commission

System.out.println("\nSummary:");

System.out.printf("Total Sales: $%.2f%n", totalSales);

System.out.printf("Total Commission: $%.2f%n", totalCommission);

scanner.close();

}

}

1. Design a Java program that merges two sorted arrays into a single sorted array.

import java.util.Arrays;

public class MergeArrays {

public static int[] mergeArrays(int[] array1, int[] array2) {

int[] mergedArray = new int[array1.length + array2.length];

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

while (i < array1.length && j < array2.length) {

mergedArray[k++] = (array1[i] <= array2[j]) ? array1[i++] : array2[j++];

}

while (i < array1.length) {

mergedArray[k++] = array1[i++];

}

while (j < array2.length) {

mergedArray[k++] = array2[j++];

}

return mergedArray;

}

public static void main(String[] args) {

int[] array1 = {1, 3, 5};

int[] array2 = {2, 4, 6};

int[] mergedArray = mergeArrays(array1, array2);

System.out.println("Merged Array: " + Arrays.toString(mergedArray));

}

}

1. Develop a program that checks whether a given number is prime or not. Use a for loop

import java.util.Scanner;

public class PrimeCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

boolean isPrime = number > 1;

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

if (number % i == 0) {

isPrime = false;

break;

}

}

System.out.println(number + (isPrime ? " is a prime number." : " is not a prime number."));

scanner.close();

}

}

1. Write a Java program that finds the first even number in a list and breaks the loop when it finds one.

import java.util.Arrays;

import java.util.List;

public class FirstEven {

public static void main(String[] args) {

// Create a list of integers

List<Integer> numbers = Arrays.asList(1, 3, 5, 7, 8, 10, 13);

// Variable to store the first even number

Integer firstEven = null;

// Iterate through the list

for (Integer number : numbers) {

// Check if the number is even

if (number % 2 == 0) {

firstEven = number;

break; // Exit the loop once an even number is found

}

}

// Output the result

if (firstEven != null) {

System.out.println("The first even number is: " + firstEven);

} else {

System.out.println("No even number found in the list.");

}

}

}

1. Write a program that prints all numbers from 1 to 10 except for the number 5, using a continue statement (from original list).

class Skip

{

public static void main(String[] args){

for(int i = 1;i<=10;i++){

if(i==1){

continue;

}

System.out.println(i);

}

}

}