NAME:Gundupalli Srujan Deep

REG:192125009

Course:CSA0978-Programming in java

1.

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class EventHandlingApplet extends Applet implements ActionListener {

Button button;

public void init() {

button = new Button("Click me!");

add(button);

button.addActionListener(this);

}

public void actionPerformed(ActionEvent event) {

if (event.getSource() == button) {

System.out.println("Button clicked!");

}

}

}

2.

import java.io.\*;

public class FileReadWriteExample {

public static void main(String[] args) {

String filename = "welcome.txt";

String welcomeString = "WELCOME TO SSE";

try {

// Write the string to a file using FileOutputStream

FileOutputStream outputStream = new FileOutputStream(filename);

byte[] bytes = welcomeString.getBytes();

outputStream.write(bytes);

outputStream.close();

// Read the string from the file using FileInputStream

FileInputStream inputStream = new FileInputStream(filename);

bytes = new byte[inputStream.available()];

inputStream.read(bytes);

inputStream.close();

String readString = new String(bytes);

System.out.println(readString);

} catch (IOException e) {

e.printStackTrace();

}

}

}

3.

import java.util.regex.Pattern;

public class Solution {

public static boolean detectCapitalUse(String word) {

Pattern pattern1 = Pattern.compile("[A-Z]\*");

Pattern pattern2 = Pattern.compile("[a-z]\*");

Pattern pattern3 = Pattern.compile("[A-Z][a-z]\*");

return pattern1.matcher(word).matches() ||

pattern2.matcher(word).matches() ||

pattern3.matcher(word).matches();

}

}

4.

public class Solution {

public static char nextGreatestLetter(char[] letters, char target) {

int left = 0;

int right = letters.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (letters[mid] <= target) {

left = mid + 1;

} else {

right = mid - 1;

}

}

if (left == letters.length) {

return letters[0];

} else {

return letters[left];

}

}

}

5.

import java.util.Scanner;

public class Menusel {

public static void main(String args[]) {

Scanner scan = new Scanner(System.in);

char choice = '0';

do {

System.out.println("Help on : ");

System.out.println("1. if");

System.out.println("2. switch");

System.out.println("3. while");

System.out.println("4. do-while");

System.out.println("5. for\n");

System.out.println("Choose any one : ");

choice = scan.next().charAt(0);

System.out.println("\n");

if (choice < '1' || choice > '5') {

System.out.println("Invalid choice. Please choose again.\n");

}

} while (choice < '1' || choice > '5');

switch (choice) {

case '1':

System.out.println("The if :\n");

System.out.println("if(condition)\n{\n\tstatement\n}");

System.out.println("else\n{\n\tstatement\n}");

break;

case '2':

System.out.println("The switch :\n");

System.out.println("switch(expression)\n{");

System.out.println("\tcase constant: statement sequence\n\tbreak;");

System.out.println("\t//...\n}");

break;

case '3':

System.out.println("The while :\n");

System.out.println("while(condition)\n{");

System.out.println("\t// body of loop\n}");

break;

case '4':

System.out.println("The do-while :\n");

System.out.println("do\n{");

System.out.println("\t// body of loop\n\n}while(condition);");

break;

case '5':

System.out.println("The for :\n");

System.out.println("for(initialization; condition; iteration)\n{");

System.out.println("\t// body of loop\n}");

break;

}

}

}

6.

import java.util.Scanner;

public class PrimeCompositeCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int primesCount = 0;

int compositesCount = 0;

System.out.print("Enter numbers separated by spaces: ");

String input = scanner.nextLine();

String[] numbers = input.split(" ");

for (String number : numbers) {

int num = Integer.parseInt(number);

boolean isPrime = true;

// Check if the number is prime or composite

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

if (num % i == 0) {

isPrime = false;

break;

}

}

// Increment the counters based on the number's status

if (isPrime && num > 1) {

primesCount++;

} else {

compositesCount++;

}

}System.out.println("Number of prime numbers entered: " + primesCount);

System.out.println("Number of composite numbers entered: " + compositesCount);

}

}

7.

import java.util.Arrays;

public class MinMaxSumDiff {

public static void main(String[] args) {

int[] arr = {10, 20, 30, 40, 50, 60};

int m = 2; // Mth maximum number

int n = 3; // Nth minimum number

Arrays.sort(arr);

int mthMax = arr[arr.length - m];

Arrays.sort(arr);

int nthMin = arr[n - 1];

int sum = mthMax + nthMin;

int diff = mthMax - nthMin;

System.out.println("Mth maximum number: " + mthMax);

System.out.println("Nth minimum number: " + nthMin);

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

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

}

}

8.

import java.util.Scanner;

public class BonusCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the salary: ");

double salary = scanner.nextDouble();

System.out.print("Enter the grade (A/B): ");

char grade = scanner.next().charAt(0);

double bonusPercentage;

if (grade == 'A') {

bonusPercentage = 5.0;

} else {

bonusPercentage = 10.0;

}

if (salary < 10000) {

bonusPercentage += 2.0;

}

double bonusAmount = salary \* (bonusPercentage / 100.0);

double finalSalary = salary + bonusAmount;

System.out.printf("Bonus percentage: %.2f%%\n", bonusPercentage);

System.out.printf("Bonus amount: $%.2f\n", bonusAmount);

System.out.printf("Final salary: $%.2f\n", finalSalary);

scanner.close();

}

}

9.

import java.util.Scanner;

public class PerfectNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

scanner.close();

int count = 0;

int num = 1;

while (count < n) {

if (isPerfectNumber(num)) {

System.out.println(num);

count++;

}

num++;

}

}

public static boolean isPerfectNumber(int num) {

int sum = 0;

for (int i = 1; i <= num / 2; i++) {

if (num % i == 0) {

sum += i;

}

}

return sum == num;

}

}

10.

import java.util.Scanner;

public class StudentGrade {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the marks of the student in four subjects:");

int mark1 = scan.nextInt();

int mark2 = scan.nextInt();

int mark3 = scan.nextInt();

int mark4 = scan.nextInt();

int total = mark1 + mark2 + mark3 + mark4;

double aggregate = total / 4.0;

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

System.out.println("Aggregate percentage: " + aggregate);

String grade = "";

if (aggregate >= 75) {

grade = "Distinction";

} else if (aggregate >= 60 && aggregate < 75) {

grade = "First Division";

} else if (aggregate >= 50 && aggregate < 60) {

grade = "Second Division";

} else if (aggregate >= 40 && aggregate < 50) {

grade = "Third Division";

} else {

grade = "Fail";

}

System.out.println("Grade obtained: " + grade);

}

}

11.

import java.util.Scanner;

public class AverageOfNumbers {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int num, positiveSum = 0, negativeSum = 0, positiveCount = 0, negativeCount = 0;

System.out.println("Enter numbers (enter -1 to stop):");

while (true) {

num = input.nextInt();

if (num == -1) {

break;

} else if (num > 0) {

positiveSum += num;

positiveCount++;

} else {

negativeSum += num;

negativeCount++;

}

}

double positiveAvg = (positiveCount > 0) ? (double) positiveSum / positiveCount : 0;

double negativeAvg = (negativeCount > 0) ? (double) negativeSum / negativeCount : 0;

System.out.println("Average of positive numbers: " + positiveAvg);

System.out.println("Average of negative numbers: " + negativeAvg);

}

}

12.

import java.util.Scanner;

public class CountCharacters {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int upperCount = 0, lowerCount = 0, numCount = 0;

char ch;

System.out.println("Enter a character (\* to stop): ");

do {

ch = scanner.next().charAt(0);

if (Character.isUpperCase(ch)) {

upperCount++;

} else if (Character.isLowerCase(ch)) {

lowerCount++;

} else if (Character.isDigit(ch)) {

numCount++;

}

} while (ch != '\*');

System.out.println("Number of uppercase characters: " + upperCount);

System.out.println("Number of lowercase characters: " + lowerCount);

System.out.println("Number of numeric characters: " + numCount);

}

}

13.

public class PrimeChecker implements Runnable {

private int number;

public PrimeChecker(int number) {

this.number = number;

}

public void run() {

if (isPrime(number)) {

System.out.println(number + " is a prime number");

} else {

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

}

}

private boolean isPrime(int number) {

if (number <= 1) {

return false;

}

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

if (number % i == 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

int[] numbers = { 7, 8, 11, 21, 23, 27 };

for (int number : numbers) {

PrimeChecker checker = new PrimeChecker(number);

Thread thread = new Thread(checker);

thread.start();

}

}

}

14.

class SuperClass {

int num;

public SuperClass(int num) {

this.num = num;

}

}

class SubClass extends SuperClass {

int num;

public SubClass(int num1, int num2) {

super(num1);

this.num = num2;

}

public void display() {

System.out.println("Value of num in SuperClass: " + super.num);

System.out.println("Value of num in SubClass: " + this.num);

}

}

public class Main {

public static void main(String[] args) {

SubClass obj = new SubClass(10, 20);

obj.display();

}

}

15.

public class MultiplicationTable implements Runnable {

private int table;

public MultiplicationTable(int table) {

this.table = table;

}

@Override

public void run() {

System.out.println(Thread.currentThread().getName() + " started");

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

System.out.println(table + " x " + i + " = " + (table \* i));

try {

Thread.sleep(500);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println(Thread.currentThread().getName() + " ended");

}

public static void main(String[] args) {

System.out.println("Main thread started");

Thread t1 = new Thread(new MultiplicationTable(5), "Table 5 thread");

Thread t2 = new Thread(new MultiplicationTable(10), "Table 10 thread");

t1.start();

t2.start();

System.out.println("Main thread ended");

}

}