CSA09 Programming in Java

# Day 4 Assignment Questions

SUBMITTED BY:K.HARSHASAI TEJA REGISTER NO:192111481

1. Create File1.txt file, in which, store more than one line of text. Write a Java program to

count the no. of words, characters and lines from the input file File1.txt.

# PROGRAM:

import java.util.\*;

public class filestats {

public static void main(String[] args) {

String fileName = "File1.txt";

int wordCount = 0;

int charCount = 0;

int lineCount = 0;

try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

String line;

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

lineCount++;

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

wordCount += words.length;

for (String word : words) {

charCount += word.length();

}

}

System.out.println("Word count: " + wordCount);

System.out.println("Character count: " + charCount);

System.out.println("Line count: " + lineCount);

} catch (IOException e) {

e.printStackTrace();

}

}

}

2. Create Customer class with deposit() and withdraw() as synchronized methods. Declare

AccountNo, AccName and Balance as Instance Variables inside the class. From the main

class, Input the amount for withdraw() operation and if requested amount is not available

in existing Balance amount, withdraw() method should be temporarily suspended using

wait() method until deposit() method receives the input for amount, to be added in the

existing Balance amount and then withdraw() would be completed in a successful

manner. Develop the above scenario using Synchronization and Inter-Thread

Communication.

Debugging

# PROGRAM:

import java.util.\*;

class customer {

private int AccountNo;

private String AccName;

private int Balance;

public Customer(int AccountNo, String AccName, int Balance) {

this.AccountNo = AccountNo;

this.AccName = AccName;

this.Balance = Balance;

}

public synchronized void deposit(int amount) {

Balance += amount;

System.out.println("Deposited " + amount + " to " + AccName + "'s account. New balance: " + Balance);

notify();

}

public synchronized void withdraw(int amount) {

while (Balance < amount) {

try {

System.out.println("Insufficient balance. Waiting for deposit...");

wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

Balance -= amount;

System.out.println("Withdrawn " + amount + " from " + AccName + "'s account. New balance: " + Balance);

}

}

3. Given an integer n, return a string array answer (1-indexed) where:

answer[i] == &quot;FizzBuzz&quot; if i is divisible by 3 and 5.

answer[i] == &quot;Fizz&quot; if i is divisible by 3.

answer[i] == &quot;Buzz&quot; if i is divisible by 5.

answer[i] == i (as a string) if none of the above conditions are true.

Example 1:

Input: n = 3

Output: [&quot;1&quot;,&quot;2&quot;,&quot;Fizz&quot;]

# PROGRAM:

import java.util.\*;

public class fizzbuzz {

public static String[] fizzBuzz(int n) {

String[] answer = new String[n];

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

if(i%3 == 0 && i%5 == 0){

answer[i-1] = "FizzBuzz";

}

else if(i%3 == 0){

answer[i-1] = "Fizz";

}

else if(i%5 == 0){

answer[i-1] = "Buzz";

}

else{

answer[i-1] = Integer.toString(i);

}

}

return answer;

}

public static void main(String[] args) {

int n = 3;

String[] answer = fizzBuzz(n);

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

}

}

Example 2:

Input: n = 5

Output: [&quot;1&quot;,&quot;2&quot;,&quot;Fizz&quot;,&quot;4&quot;,&quot;Buzz&quot;]

# PROGRAM:

import java.util.\*;

public class fizzbuzz1 {

public static void main(String[] args) {

int n = 15; // input value

String[] answer = fizzBuzz(n); // calling fizzBuzz function

for (String s : answer) {

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

}

}

public static String[] fizzBuzz(int n) {

String[] answer = new String[n];

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

if (i % 3 == 0 && i % 5 == 0) {

answer[i - 1] = "FizzBuzz";

} else if (i % 3 == 0) {

answer[i - 1] = "Fizz";

} else if (i % 5 == 0) {

answer[i - 1] = "Buzz";

} else {

answer[i - 1] = String.valueOf(i);

}

}

return answer;

}

}

Example 3:

Input: n = 15

Output:

[&quot;1&quot;,&quot;2&quot;,&quot;Fizz&quot;,&quot;4&quot;,&quot;Buzz&quot;,&quot;Fizz&quot;,&quot;7&quot;,&quot;8&quot;,&quot;Fizz&quot;,&quot;Buzz&quot;,&quot;11&quot;,&quot;Fizz&quot;,&quot;13&quot;,&quot;14&quot;,&quot;FizzBuzz&quot;]

Constraints:

1 &lt;= n &lt;= 104

class Solution {

    vector&lt;string&gt; fizzBuzz(int n) {

    }

}

# PROGRAM:

import java.util.\*;

class fizzbuzzsolution {

public List<String> fizzBuzz(int n) {

List<String> answer = new ArrayList<>();

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

if (i % 3 == 0 && i % 5 == 0) {

answer.add("FizzBuzz");

} else if (i % 3 == 0) {

answer.add("Fizz");

} else if (i % 5 == 0) {

answer.add("Buzz");

} else {

answer.add(Integer.toString(i));

}

}

return answer;

}

}

public class Main {

public static void main(String[] args) {

Solution solution = new Solution();

int n = 15;

List<String> answer = solution.fizzBuzz(n);

System.out.println(answer);

}

}

4. Given two strings s and goal, return true if and only if s can become goal after some

number of shifts on s.

A shift on s consists of moving the leftmost character of s to the rightmost position.

For example, if s = &quot;abcde&quot;, then it will be &quot;bcdea&quot; after one shift.

Example 1:

Input: s = &quot;abcde&quot;, goal = &quot;cdeab&quot;

Output: true

# PROGRAM:

import java.util.\*;

public class goal {

public boolean canShift(String s, String goal) {

if (s.length() != goal.length()) {

return false;

}

String s2 = s + s;

return s2.contains(goal);

}

}

Example 2:

Input: s = &quot;abcde&quot;, goal = &quot;abced&quot;

Output: false

Constraints:

1 &lt;= s.length, goal.length &lt;= 100

s and goal consist of lowercase English letters.

class Solution {

    bool rotateString(string s, string goal) {

    }

}

# PROGRAM:

import java.util.\*;

class goalsolution {

public boolean rotateString(String s, String goal) {

if (s.length() != goal.length()) {

return false;

}

String rotated = s + s;

return rotated.contains(goal);

}

}

5. Program to Find Prime No using multithreading: Find/Debug errors in the following

code

class PrimeExample implements Runnable

//extends Thread

{

public void run()

{

int i,m=20,flag=1;

for(i=1;i&lt;=m;i++);

{

if(i&lt;=3){

System.out.println(i + &quot; is prime number&quot;);

continue;

}

else if(i&gt;3)

{

for(int j=2;j&lt;i;i++)

{

if(i%j==0)

{

flag=0;

break;

}

}

if (flag!= 1)

{

System.out.printn(i + &quot; is not prime number&quot;);

flag=1;

}

else

System.out.prinln(i + &quot; is prime number&quot;);

}

}

}

}

class prime

{

public static void main(String args[]){

try

{

PrimeExample p1 = new PrimeExample();

Thread t1= new Thread(p2);

t1.start();

}

catch(Exception e)

{

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

}

}

}

# PROGRAM:

import java.util.\*;

class primeexampleimplementsrunnable {

public void run() {

int i, m = 20, flag = 1;

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

if (i <= 3) {

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

continue;

} else if (i > 3) {

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

if (i % j == 0) {

flag = 0;

break;

}

}

if (flag != 1) {

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

flag = 1;

} else {

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

}

}

}

}

}

class prime {

public static void main(String args[]) {

try {

PrimeExample p1 = new PrimeExample();

Thread t1 = new Thread(p1);

t1.start();

} catch (Exception e) {

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

}

}

}

6. Write a program to find whether the person is eligible for vote or not. And if that

particular person is not eligible, then print how many years are left to be eligible.

Sample Input:

Enter your age: 7

Sample output:

You are allowed to vote after 11 years

Test cases:

1. 25

2. Eighteen

3. 12

4. -18

5. 34.5

# PROGRAM:

import java.util.Scanner;

public class voteeligibility {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter your age: ");

int age = sc.nextInt();

if (age < 0) {

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

} else if (age >= 18) {

System.out.println("You are eligible to vote!");

} else {

int yearsLeft = 18 - age;

System.out.println("You are not eligible to vote.");

System.out.println("You will be eligible to vote after " + yearsLeft + " years.");

}

}

}

7. Find the LCM and GCD of n numbers?

Sample Input:

N value = 2

Number 1 = 16

Number 2 = 20

Sample Output:

LCM = 80

GCD = 4

Test cases:

1. N = 3, {12, 25, 30}

2. N = 2, {52, 25, 63}

3. N = 3, {17, 19, 11}

4. N = -2, {52, 60}

5. N = 2, {30, 45}

# PROGRAM:

import java.util.Scanner;

import java.util.\*;

public class lcmandgcd {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

if(n<=0){

System.out.println("Invalid input");

return;

}

int[] arr = new int[n];

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

System.out.print("Enter number " + (i + 1) + ": ");

arr[i] = sc.nextInt();

}

int lcm = arr[0];

int gcd = arr[0];

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

lcm = getLCM(lcm, arr[i]);

gcd = getGCD(gcd, arr[i]);

}

System.out.println("LCM = " + lcm);

System.out.println("GCD = " + gcd);

}

public static int getLCM(int a, int b) {

return (a \* b) / getGCD(a, b);

}

public static int getGCD(int a, int b) {

if (b == 0) {

return a;

}

return getGCD(b, a % b);

}

}

8. Write a program using function to calculate the simple interest. Suppose the customer is a

senior citizen. He is being offered 12 percent rate of interest; for all other customers, the

ROI is 10 percent.

Sample Input:

Enter the principal amount: 200000

Enter the no of years: 3

Is customer senior citizen (y/n): n

Sample Output:

Interest: 60000

Test Cases:

1. Principal: 2000 , Years: 0

2. Principal: 20000 , Years: -2

3. Principal: -2000 , Years: 2

4. Principal: 2 , Years: 2000

5. Principal: 0 , Years: 5

# PROGRAM:

import java.util.Scanner;

public class simpleinterestcalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the principal amount: ");

double principal = scanner.nextDouble();

System.out.print("Enter the no of years: ");

double years = scanner.nextDouble();

System.out.print("Is customer senior citizen (y/n): ");

String seniorCitizen = scanner.next();

boolean isSeniorCitizen = seniorCitizen.equalsIgnoreCase("y");

double rateOfInterest = isSeniorCitizen ? 0.12 : 0.10;

double interest = calculateSimpleInterest(principal, years, rateOfInterest);

System.out.println("Interest: " + interest);

}

public static double calculateSimpleInterest(double principal, double years, double rateOfInterest) {

if (principal <= 0 || years <= 0) {

return 0;

}

double simpleInterest = (principal \* rateOfInterest \* years);

return simpleInterest;

}

}

9. Write a program to print the Fibonacci series.

Sample Input:

Enter the n value: 6

Sample Output:

0 1 1 2 3 5

# PROGRAM:

import java.util.Scanner;

public class fibonacciseries {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int n = input.nextInt();

int a = 0, b = 1;

System.out.print(a + " " + b);

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

int c = a + b;

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

a = b;

b = c;

}

}

}

10. Java Program to Find Even Sum of Fibonacci Series Till number N?

Sample Input: n = 4

Sample Output: 33

(N = 4, So here the fibonacci series will be produced from 0th term till 8th term:0, 1, 1, 2, 3,

5, 8, 13, 21

Sum of numbers at even indexes = 0 + 1 + 3 + 8 + 21 = 33)

# PROGRAM:

import java.util.Scanner;

public class evensumoffibonacciseries {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

sc.close();

int sum = 0;

int a = 0;

int b = 1;

int fib = 0;

while (fib <= n) {

if (fib % 2 == 0) {

sum += fib;

}

a = b;

b = fib;

fib = a + b;

}

System.out.println("The sum of even numbers in the Fibonacci series till " + n + " is: " + sum);

}

}

11. Write a program to print the numbers from M to N by skipping K numbers in between?

Sample Input:

M = 50

N = 100

K = 7

Sample Output:

50, 58, 66, 74, …..

Test cases:

1. M = 15, N = 05, K = 02

2. .M = 25, N = 50, K = 04

3. M = 15, N = 100, K = -02

4. M = 0 , N = 0 , K = 2

5. M = 200 , N = 200 , K = 50

# PROGRAM:

import java.util.Scanner;

public class skipnumbers {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int M = sc.nextInt();

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

int N = sc.nextInt();

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

int K = sc.nextInt();

for (int i = M; i <= N; i += K + 1) {

System.out.print(i + ", ");

}

}

}

12. Write a program to print all the composite numbers between a and b?

Sample Input:

A = 12

B = 19

Sample Output

14, 15, 16, 18

Test cases:

1. A = 11, B = 11

2. A = 20, B = 10

3. A = 0, B = 0

4. A = -5, B = 5

5. A = 7, B = -12

# PROGRAM:

import java.util.Scanner;

public class compositenumbers {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int a = input.nextInt();

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

int b = input.nextInt();

System.out.print("Composite numbers between " + a + " and " + b + " are: ");

for (int i = a; i <= b; i++) {

if (i > 1 && isComposite(i)) {

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

}

}

}

public static boolean isComposite(int num) {

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

if (num % i == 0) {

return true;

}

}

return false;

}

}

13. Find the factorial of n?

Sample Input:

N = 4

Sample Output:

4 Factorial = 24

Test cases:

1. N = 0

2. N = -5

3. N = 1

4. N = Q

5. N = 3A

# PROGRAM:

import java.util.Scanner;

public class factorial {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a number to find its factorial: ");

int n = sc.nextInt();

if (n < 0) {

System.out.println("Factorial of negative numbers is not defined.");

} else {

int factorial = 1;

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

factorial \*= i;

}

System.out.println(n + " Factorial = " + factorial);

}

}

}

14. Find the year of the given date is leap year or not

Sample Input:

Enter Date: 04/11/1947

Sample Output:

Given year is Non Leap Year

Test cases:

1. 04/11/19.47

2. 11/15/1936

3. 31/45/1996

4. 64/09/1947

5. 00/00/2000

# PROGRAM:

import java.util.Scanner;

public class leapyearchecker {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String inputDate = sc.nextLine();

String[] dateArr = inputDate.split("/");

int year = Integer.parseInt(dateArr[2]);

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {

System.out.println("Given year is Leap Year");

} else {

System.out.println("Given year is Non Leap Year");

}

sc.close();

}

}

15. Find the number of factors for the given number

Sample Input:

Given number: 100

Sample Output:

Number of factors = 9

Test cases:

1. 343

2. 1080

3. -243

4. 101010

5. 0

# PROGRAM:

import java.util.Scanner;

public class numberoffactors {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Given number: ");

int num = sc.nextInt();

if (num < 0) {

System.out.println("Invalid input: Please enter a positive number.");

return;

} else if (num == 0) {

System.out.println("Number of factors = 0");

return;

}

int count = 0;

for (int i = 1; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

count++;

if (i != num/i) {

count++;

}

}

}

System.out.println("Number of factors = " + count);

}

}

16. Write a program to print the given number is Perfect number or not?

Sample Input:

Given Number: 6

Sample Output:

It’s a Perfect Number

Test cases:

1. 17

2. 26!

3. 143

4. 84.1

5. -963

# PROGRAM:

import java.util.Scanner;

public class perfectnumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

sc.close();

if (isPerfectNumber(n)) {

System.out.println("It's a Perfect Number");

} else {

System.out.println("It's not a Perfect Number");

}

}

public static boolean isPerfectNumber(int n) {

int sum = 0;

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

if (n % i == 0) {

sum += i;

}

}

return sum == n;

}

}

17. Write a program to find the square, cube of the given decimal number

Sample Input:

Given Number: 0.6

Sample Output:

Square Number: 0.36

Cube Number:0.216

Test cases:

1. 12

2. 0

3. -0.5

4. 14.25

5. -296

# PROGRAM:

import java.util.Scanner;

public class squareandcube {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

double number = input.nextDouble();

double square = number \* number;

double cube = number \* number \* number;

System.out.println("Square Number: " + square);

System.out.println("Cube Number: " + cube);

}

}

18. Find the n th odd number after n odd number

Sample Input: N : 7

Sample Output:

Hence the values printed for i are 1 , 3 , 5.

Test cases:

1. N = 0

2. N = -6

3. N = 2021

4. N = -14.5

5. N = -196

# PROGRAM:

import java.util.Scanner;

public class oddnumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

if (n < 0) {

System.out.println("Invalid input: N should be a positive integer.");

return;

}

int count = 0;

for (int i = 1; count < n; i += 2) {

if (i > 1) {

System.out.print(", ");

}

if (count >= n - 1) {

System.out.print(i);

}

count++;

}

}

}

19. Program to find the frequency of each element in the array.

Sample Input &amp; Output:

{1, 2, 8, 3, 2, 2, 2, 5, 1}

Pseudo:

Element | Frequency

--------------------------

1 | 2

2 | 4

8 | 1

3 | 1

4 | 1

# PROGRAM:

import java.util.\*;

public class frequencyofelements {

public static void main(String[] args) {

int[] arr = {1, 2, 8, 3, 2, 2, 2, 5, 1};

Map<Integer, Integer> freqMap = new HashMap<>();

for (int i = 0; i < arr.length; i++) {

if (freqMap.containsKey(arr[i])) {

freqMap.put(arr[i], freqMap.get(arr[i]) + 1);

} else {

freqMap.put(arr[i], 1);

}

}

System.out.println("Element | Frequency");

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

for (Map.Entry<Integer, Integer> entry : freqMap.entrySet()) {

System.out.println(entry.getKey() + " | " + entry.getValue());

}

}

}

20. Program to find whether the given number is Armstrong number or not

Sample Input:

Enter number: 153

Sample Output:

Given number is Armstrong number

Test cases:

1. 370

2. 1

3. 371

4. 145678

5. 0.21345

# PROGRAM:

import java.util.Scanner;

public class armstrongnumber {

public static void main(String[] args) {

int num, originalNum, remainder, result = 0;

Scanner sc = new Scanner(System.in);

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

num = sc.nextInt();

originalNum = num;

while (originalNum != 0) {

remainder = originalNum % 10;

result += Math.pow(remainder, 3);

originalNum /= 10;

}

if (result == num) {

System.out.println("Given number is Armstrong number");

} else {

System.out.println("Given number is not Armstrong number");

}

}

}

21. Write a program to find the sum of digits of N digit number (sum should be single digit)

Sample Input:

Enter N value: 3

Enter 3 digit numbers: 143

Test cases:

1. N = 2, 158

2. N = 3, 14

3. N = 4, 0148

4. N = 1, 0004

5. N = 4, 7263

# PROGRAM:

import java.util.Scanner;

public class singledigitsum {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

System.out.print("Enter " + n + " digit numbers: ");

int num = sc.nextInt();

int sum = 0;

while (num > 0) {

sum += num % 10;

num /= 10;

}

while (sum > 9) {

sum = sum / 10 + sum % 10;

}

System.out.println("The single digit sum is: " + sum);

}

}

22. Write a program to find the square root of a perfect square number(print both the positive

and negative values)

Sample Input:

Enter the number: 6561

Sample Output:

Square Root: 81, -81

Test cases:

1. 1225

2. 9801

3. 1827

4. -100

5. 0

# PROGRAM:

import java.util.Scanner;

public class squareroot {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

double sqrt = Math.sqrt(num);

if (sqrt == (int)sqrt) {

System.out.println("Square Root: " + (int)sqrt + ", " + -(int)sqrt);

} else {

System.out.println(num + " is not a perfect square.");

}

sc.close();

}

}