## CSA0980 -- PROGRAMMING IN JAVA FOR IDL TECHNOLOGY :-

### M . NAGA KESHAVA

### REG NO :-192125019

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.io.\*;

public class Test {

public static void main(String[] args)

throws IOException

{

File file = new File("C:\Users\lenova\\TextReader.txt");

FileInputStream fileInputStream = new FileInputStream(file);

InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream); BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

String line;

int wordCount= 0;

int characterCount = 0;

int paraCount = 0;

int whiteSpaceCount = 0; int sentenceCount = 0;

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

if (line.equals("")) {

paraCount += 1;

}

else {

characterCount += line.length();

String words[] = line.split("\\s+"); wordCount += words.length; whiteSpaceCount += wordCount - 1;

String sentence[] = line.split("[!?.:]+");

sentenceCount += sentence.length;

}

}

if (sentenceCount >= 1) {

paraCount++;

}

System.out.println("Total word count = "+ wordCount);

System.out.println("Total number of sentences = "+ sentenceCount);

System.out.println("Total number of characters = "+ characterCount);

System.out.println("Number of paragraphs = "+ paraCount);

System.out.println("Total number of whitespaces = "+ whiteSpaceCount);

}

}

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.

**Program**

class Bank {

// Initial balance $100

int total = 100;

void withdrawn(String name, int withdrawal)

{

if (total >= withdrawal) {

System.out.println(name + " withdrawn "

+ withdrawal);

total = total - withdrawal;

System.out.println("Balance after withdrawal: "

+ total);

// Making the thread sleep for 1 second after

// each withdrawal

// Try block to check for exceptions

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

}

}

else {

// Print statements

System.out.println(name

+ " you can not withdraw "

+ withdrawal);

System.out.println("your balance is: " + total);

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

void deposit(String name, int deposit)

{

System.out.println(name + " deposited " + deposit);

total = total + deposit;

System.out.println("Balance after deposit: "+ total);

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

class GFG {

public static void main(String[] args)

{

Bank obj = new Bank();

// Custom input - Transactions

obj.withdrawn("Arnab", 20);

obj.withdrawn("Monodwip", 40);

obj.deposit("Mukta", 35);

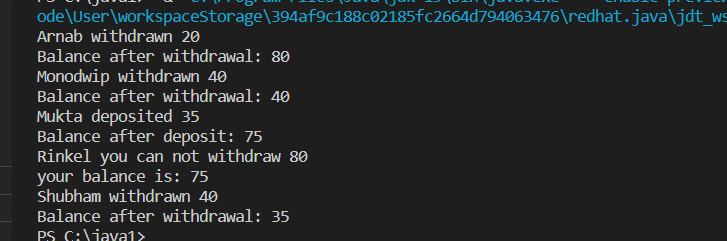
obj.withdrawn("Rinkel", 80);

obj.withdrawn("Shubham", 40);

}

}

**Output**



QUESTION 3:

class Solution {

public List<String> fizzBuzz(int n)

{ List<String> result = new ArrayList<String>(); for(int i=1;i<=n;i++){

if(i%3==0 && i%5==0){ result.add("FizzBuzz");

continue;

}

if(i%3==0){ result.add("Fizz"); continue; }

if(i%5==0){

result.add("Buzz"); continue;

} result.add(i+"");

} return result;

}}

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

**Program**

import java.util.Scanner;

public class Vote

{

public static void main(String[] args)

{

int age;

Scanner sc = new Scanner(System.in);

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

age = sc.nextInt();

if (age >= 18)

{

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

}

else

{

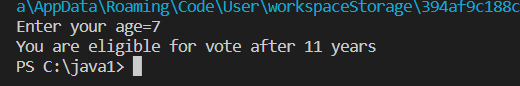
System.out.println("You are not eligible for vote.");

}

}

}

**Output**

****

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

**Program**

import java.util.Scanner;

public class LcmGcd

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

if (b == 0)

return a;

return gcd(b, a % b);

}

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

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

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter the values: ");

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

arr[i] = sc.nextInt();

}

int gcd = arr[0], lcm = arr[0];

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

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

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

}

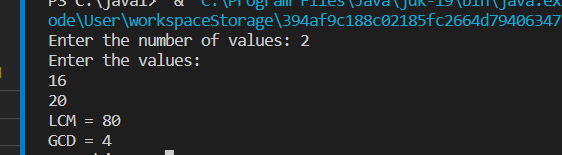
System.out.println("LCM = " + lcm);

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

sc.close();

}}

**Output**

****

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.

import java.util.Scanner;

public class SimpleInterest {

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: ");

int years = scanner.nextInt();

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

String isSeniorCitizen = scanner.next();

double rateOfInterest = isSeniorCitizen.equalsIgnoreCase("y") ? 0.12 : 0.10;

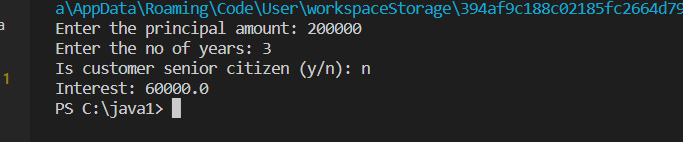
double simpleInterest = principal \* rateOfInterest \* years;

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

}

}

**Output**

****

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.\*;

class fibonacci{

public static void main(String[] args) {

scanner sc=new Scanner(System.in);

System.out.println(“enter the n value:”) ;

Int n=sc.nextInt();

int firstTerm = 0, secondTerm = 1;

System.out.println("Fibonacci Series till " + n + " terms:");

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

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

// compute the next term

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

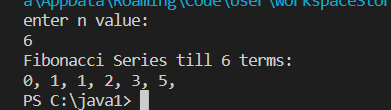
secondTerm = nextTerm;

}

}

}

**Output**



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

Sample Input: n = 4

Sample Output: 33

**Program**

import java.util.Scanner;

import java.io.\*;

public class FabonacciSum {

public static void main(String[] args){

int my\_input, i, sum;

System.out.println("Required packages have been imported");

Scanner my\_scanner = new Scanner(System.in);

System.out.println("A reader object has been defined ");

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

my\_input = my\_scanner.nextInt();

int fabonacci[] = new int[2 \* my\_input + 1];

fabonacci[0] = 0;

fabonacci[1] = 1;

sum = 0;

for (i = 2; i <= 2 \* my\_input; i++) {

fabonacci[i] = fabonacci[i - 1] + fabonacci[i - 2];

if (i % 2 == 0)

sum += fabonacci[i];

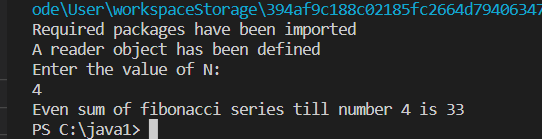
}

System.out.printf("Even sum of fibonacci series till number %d is %d" , my\_input, sum);

}

}

**Output**

****

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, …..

**Program**

public class SkipNumbers {

public static void main(String[] args) {

int m = 50;

int n = 100;

int k = 7;

for (int i = m; i <= n; i += k) {

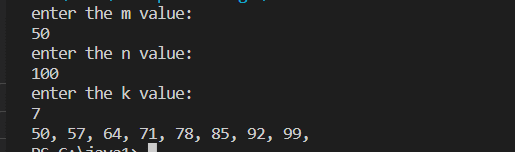
System.out.print(i + ", ");

}

}

}

**Output**



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

**Program**

public class PrintCompositeNumbers {

public static boolean isComposite(int num) {

if (num <= 1) {

return false;

}

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

if (num % i == 0) {

return true;

}

}

return false;

}

public static void main(String[] args) {

int a = 12;

int b = 19;

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

if (isComposite(i)) {

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

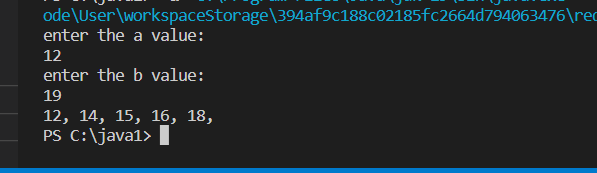
}

}

}

}

**Output**

****

13. Find the factorial of n?

Sample Input:

N = 4

Sample Output:

4 Factorial = 24

**Program**

**class** FactorialExample{

**public** **static** **void** main(String args[]){

**int** i,fact=1;

**int** number=5;//It is the number to calculate factorial

**for**(i=1;i<=number;i++){

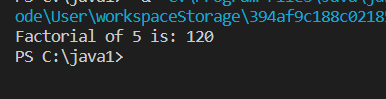
      fact=fact\*i;

  }

   System.out.println("Factorial of "+number+" is: "+fact);

 }

**Output**

****

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

**Program**

class Main {

public static void main(String[] args) {

// year to be checked int year = 1900; boolean leap = false;

// if the year is divided by 4 if (year % 4 == 0) {

// if the year is century if (year % 100 == 0) {

// if year is divided by 400 // then it is a leap year if (year % 400 == 0) leap = true; else leap = false;

}

// if the year is not century else leap = true;

}

else leap = false;

if (leap)

System.out.println(year + " is a leap year.");

else

System.out.println(year + " is not a leap year.");

}

15. Find the number of factors for the given number

Sample Input:

Given number: 100

Sample Output:

Number of factors = 9

**Program**

public class Main {

public static void main(String[] args) {

// positive number

int number = 60;

System.out.print("Factors of " + number + " are: ");

// loop runs from 1 to 60

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

// if number is divided by i

// i is the factor

if (number % i == 0) {

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

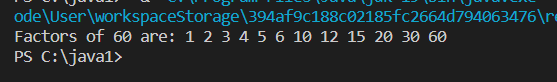
}

}

}

}

**Output**

****

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

**Program**

**import** java.util.Scanner;

**public** **class** PerfectNumberExample1

{

**public** **static** **void** main(String args[])

{

**long** n, sum=0;

Scanner sc=**new** Scanner(System.in);

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

n=sc.nextLong();

**int** i=1;

//executes until the condition becomes false

**while**(i <= n/2)

{

**if**(n % i == 0)

{

//calculates the sum of factors

Sum = sum + i;

} //end of if

//after each iteration, increments the value of variable i by 1

i++;

} //end of while

//compares sum with the number

**if**(sum==n)

{

//prints if sum and n are equal

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

} //end of if

**else**

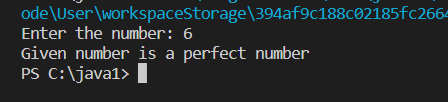
//prints if sum and n are not equal

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

}

}

**Output**



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

**Program**

import java.util.\*;

public class j5 {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

int num;

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

num = sc.nextInt();

System.out.println("Square of " + num + " is: " + Math.pow(num, 2));

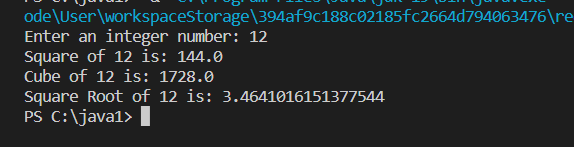
System.out.println("Cube of " + num + " is: " + Math.pow(num, 3));

System.out.println("Square Root of " + num + " is: " + Math.sqrt(num));

}

}

**Output**

****

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.

**Program**

**public** **class** oddnumber

{

**public** **static** **void** main(String args[])

{

**int** number=100;

System.out.print("List of odd numbers from 1 to "+number+": ");

**for** (**int** i=1; i<=number; i++)

{

**if** (i%2!=0)

{

System.out.print(i + " ");

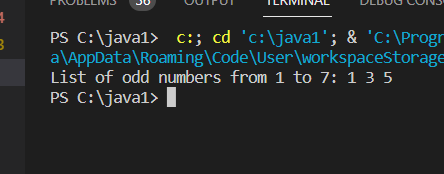
}

}

}

}

**output**

****

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}

**Program**

**public** **class** Frequency {

**public** **static** **void** main(String[] args) {

        //Initialize array

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

        //Array fr will store frequencies of element

**int** [] fr = **new** **int** [arr.length];

**int** visited = -1;

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

**int** count = 1;

**for**(**int** j = i+1; j < arr.length; j++){

**if**(arr[i] == arr[j]){

                    count++;

                    //To avoid counting same element again

                    fr[j] = visited;

                }

            }

**if**(fr[i] != visited)

                fr[i] = count;

        }

        //Displays the frequency of each element present in array

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

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

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

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

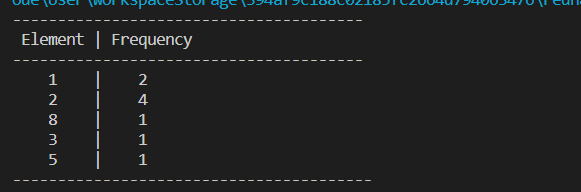
**if**(fr[i] != visited)

                System.out.println("    " + arr[i] + "    |    " + fr[i]);

        }

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

    }}

**Output**

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

**Program**

public class Armstrong {

public static void main(String[] args) {

int number = 371, originalNumber, remainder, result = 0;

originalNumber = number;

while (originalNumber != 0)

{

remainder = originalNumber % 10;

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

originalNumber /= 10;

}

if(result == number)

System.out.println(number + " is an Armstrong number.");

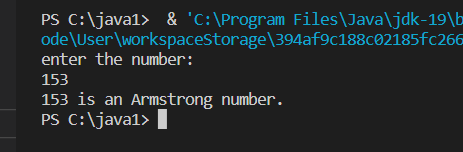
else

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

}

}

**Output**

****

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

**Program**

import java.util.";

class R192110115{

public static void main(String[] args){

int r, sum=0;

Scanner z = new Scanner(System.in); System.out.println("Enter N value:");

int Nz.nextInt();

System.out.println("Enter "+N+" digit number: ");

int a = z.nextInt();

while(a>0){

r = ax10;

Sum = sum+r;

a/=10;

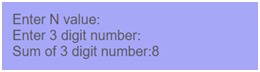
}

System.out.println("Sum of "+N+" digit number:"+sum);

}

}

**Output**



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

**Program**

1.

import java.util.\*;

class R192110115{

public static void main(String[] args){

int m;

double p;

Scanner v = new Scanner(System.in);

System.out.println("enter the number : ");

m = v.nextInt();

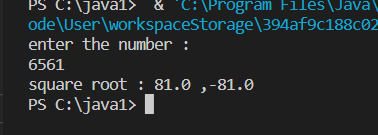
p = Math.sqrt(m);

System.out.println("square root : " + p +" "-"+p);

}

}

**Output**

****