**MAHARAJA SURAJMAL INSTITUTE**

**Affiliated to GGSIP university & NAAC ‘A’ grade accredited**

A white background with black and white clouds

Description automatically generated with medium confidence

**DEPARTMENT OF COMPUTER APPLICATIONS (IInd shift)**

**JAVA LAB**

**Subject Code : BCA104P**

**Semester-IInd (Section-A)**

**Submitted To:**

**Mr. Harjender Singh**

**Associate Professor**

**Submitted By:**

**Name: Ishvi Sharma**

**Enroll. No.: 00421202024**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Assignment** | **Date** | **Sign** |
| **1** | **Write a program to perform various scanner methods.** | **09-03-25** |  |
| **2** | **Write a program to perform arithmetic operators.** | **09-03-25** |  |
| **3** | **Write a program to perform literals.** | **09-03-25** |  |
| **4** | **Write a program to swap three numbers without using a third variable.** | **09-03-25** |  |
| **5** | **Write a program to compute the volume of cone.** | **09-03-25** |  |
| **6** | **Write a program to print total and average marks of 5 students.** | **09-03-25** |  |
| **7** | **Write a program to demonstrate scope of variables:**  **• Class instance variable**  **• Local variable**  **• Static variable.** | **09-03-25** |  |
| **8** | **Write a program to demonstrate decision statements in Java:**  **1) Else-if ladder**  **2) Nested if** | **09-03-25** |  |
| **9** | **Write a program to demonstrate the control flow-**  **1) to print the sum of 10 natural number (using for, while & do while)**  **2) to determine the odd and even number (using for & while).** | **09-03-25** |  |
| **10** | **Write a program to demonstrate switch and extended switch.** | **09-03-25** |  |
| **11** | **Write a program to find the sum of 10 number using array.** | **09-03-25** |  |
| **12** | **Write a program to generate the following pattern using for loop:**   1. **\***   **\*\***  **\*\*\***   1. **\***   **\* \***  **\* \* \***  **\* \***  **\*** | **09-03-25** |  |
| **13** | **Write a program to check if a number is positive, negative, or zero, and further checks if the positive number is even or odd using nested if and else if ladder.** | **09-03-25** |  |
| **14** | **Write a program to assign a grade based on the marks obtained by a student using nested if and else if ladder.** | **09-03-25** |  |
| **15** | **Write a program to performs basic arithmetic operations (addition, subtraction, multiplication, division) based on user input using switch case.** | **09-03-25** |  |
| **16** | **Write a program to determine the number of days in a given month using enhanced switch case.** | **09-03-25** |  |
| **17** | **Write a program that prints all characters in a string except vowels using continue and break statement.** | **09-03-25** |  |
| **18** | **Write a program that prints numbers from 1 to 100 but skips multiples of 5 using continue.** | **09-03-25** |  |
| **19** | **Write a program to demonstrate array.fill() method in java.** | **09-03-25** |  |
| **20** | **Write a program to print first 8 values with their index position using for loop in java.** | **09-03-25** |  |
| **21** | **Write a program to add two matrices and stores the result in a third matrix.** | **09-03-25** |  |
| **22** | **Write a program to search for a specific value in a 2D array.** | **09-03-25** |  |
| **23** | **Write a program to find the transpose of a matrix (rows become columns and vice versa).** | **09-03-25** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **24** | **Write a program to create a class, methods, variables and access the variables, methods using object and class.** | **22-04-25** |  |
| **25** | **Write a program to demonstrate static variables and non-static variables.** | **22-04-25** |  |
| **26** | **Write a program to demonstrate static methods and non-static methods.** | **22-04-25** |  |
| **27** | **Write a program to demonstrate static block.** | **22-04-25** |  |
| **28** | **Write a program to demonstrate static block. Write a program to declare two methods and calculate** -  **• Area of circle**  **• Area of square** | **22-04-25** |  |
| **29** | **Write a program to display all the types of inheritance such as single, multilevel and hierarchical.** | **22-04-25** |  |
| **30** | **Write a program to demonstrate final method and class.** | **22-04-25** |  |
| **31** | **Write a program to implement package in Java.** | **22-04-25** |  |
| **32** | **Write a program to demonstrate Interface to Interface in java.** | **22-04-25** |  |
| **33** | **Write a program to demonstrate interface to class in java.** | **22-04-25** |  |
| **34** | **Write a program to demonstrate multiple inheritance in java.** | **22-04-25** |  |
| **35** | **Write a program to demonstrate compile time and run time polymorphism.** | **22-04-25** |  |
| **36** | **Write a program to demonstrate instance of operator.** | **22-04-25** |  |
| **37** | **Write a program to create a new file and perform the following methods:**  **a) canread**  **b) delete**  **c) exists**  **d) length**  **e) getname**  **f) get absolute path**  **g) renameto** | **22-04-25** |  |
| **38** | **Write a program to implement filereader and filewriter methods in java.** | **22-04-25** |  |
| **39** | **Write a program to copy the content of file and paste into another file.** | **22-04-25** |  |
| **40** | **Write a program to implement exception handling using try & catch block-**  **a) IOException**  **b) Arithmetic Exception**  **c) Null pointer Exception** | **22-04-25** |  |
| **41** | **Write a program to demonstrate exception using throws and same can be managed by try, catch, and finally block.** | **04-05-25** |  |
| **42** | **Write a program to handle exception with multiple try and catch block.** | **04-05-25** |  |
| **43** | **Write a program to handle exception with one try and multiple catch block: a)nullpointer exception**  **b)Arithmetic exception**  **c)Numberformat exception d)Stringindexoutofbound exception** | **04-05-25** |  |
| **44** | **Write a program to demonstrate the thread class by using extend.** | **04-05-25** |  |
| **45** | **Write a program to demonstrate extending thread class using throws exception.** | **04-05-25** |  |
| **46** | **Write a program to implementing runnable interface in multithreading.** | **04-05-25** |  |
| **47** | **Write a program to create thread in java and run thread parallel.** | **04-05-25** |  |

**Assignment – 1**

* **Write a program to perform various scanner methods.**

**INPUT**

import java.util.Scanner;

class chanduu{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter Name:");

String name = sc.nextLine();

System.out.println("Enter Gender:");

char gender = sc.next().charAt(0);

System.out.println("Enter Age:");

int age = sc.nextInt();

System.out.println("Enter Phone Number:");

long PhoneNo = sc.nextLong();

System.out.println("Enter Marks:");

float marks = sc.nextFloat();

System.out.println("Details entered are:");

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

System.out.println("Gender : "+gender);

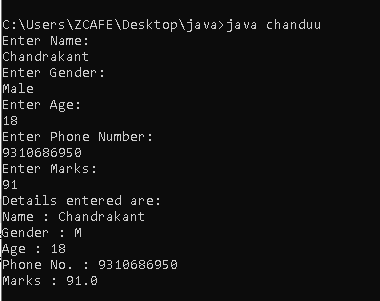
System.out.println("Age : "+age);

System.out.println("Phone No. : "+PhoneNo);

System.out.println("Marks : "+marks);

}}

**OUTPUT**



**Assignment – 2**

* **Write a program to perform arithmetic operators.**

**INPUT**

import java.util.Scanner;

class chanduu{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Arithmetic Operations");

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

int num1 = sc.nextInt();

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

int num2 = sc.nextInt();

int add = num1 + num2;

int sub = num1 - num2;

int mul = num1 \* num2;

int div = num1 / num2;

int mod = num1 % num2;

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

System.out.println("Addition = "+add);

System.out.println("Subtraction = "+sub);

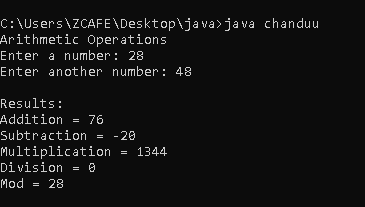
System.out.println("Multiplication = "+mul);

System.out.println("Division = "+div);

System.out.println("Mod = "+mod);

}}

**OUTPUT**



**Assignment – 3**

* **Write a program to perform literals.**

**INPUT**

public class chandu{

public static void main(String[] args)

{

int i=100;

System.out.println(i);

double f=123.456;

System.out.println(f);

char ch='c';

System.out.println(ch);

String str="chandrakant";

System.out.println(str);

boolean b=true;

System.out.println(b);

}

}

**OUTPUT**



**Assignment – 4**

* **Write a program to swap three numbers without using a third variable.**

**INPUT**

class chanduu{

public static void main(String[] args){

int a, b, c;

a = 2;

b = 6;

c = 7;

System.out.println(" Numbers Before Swapping ");

System.out.println(" First Number(a): "+a);

System.out.println(" Second Number(b): "+b);

System.out.println(" Third Number(c): "+c);

a = a + b + c;

b = a - (b + c);

c = a - (b + c);

a = a - (b + c);

System.out.println("\n Numbers After Swapping ");

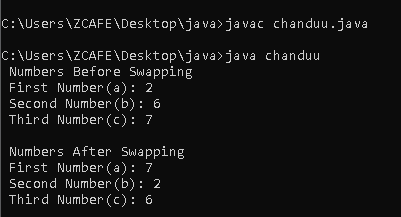
System.out.println(" First Number(a): "+a);

System.out.println(" Second Number(b): "+b);

System.out.println(" Third Number(c): "+c);

}}

**OUTPUT**



**Assignment – 5**

* **Write a program to compute the volume of cone.**

**INPUT**

import java.util.Scanner;

class Cone {

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.print("Enter the radius of the cone: ");

double radius = sc.nextDouble();

System.out.print("Enter the height of the cone: ");

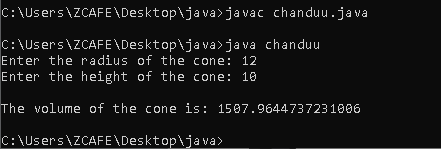
double height = sc.nextDouble();

double volume = (1.0 / 3) \* Math.PI \* radius \* radius \* height;

System.out.println("\nThe volume of the cone is: " + volume);

}}

**OUTPUT**



**Assignment – 6**

* **Write a program to print total and average marks of 5 students.**

**INPUT**

import java.util.Scanner;

class mark{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int s1,s2,s3,s4,s5,total,avg;

System.out.print("Enter marks for student 1:");

s1 = sc.nextInt();

System.out.print("Enter marks for student 2:");

s2 = sc.nextInt();

System.out.print("Enter marks for student 3:");

s3 = sc.nextInt();

System.out.print("Enter marks for student 4:");

s4 = sc.nextInt();

System.out.print("Enter marks for student 5:");

s5 = sc.nextInt();

total = s1+s2+s3+s4+s5;

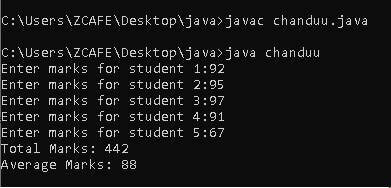
avg = total/5;

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

System.out.println("Average Marks: " + avg);

}}

**OUTPUT**



**Assignment – 7**

* **Write a program to demonstrate scope of variables:**

**• Class instance variable**

**• Local variable**

**• Static variable.**

**INPUT**

class variables{

int b = 20;

static int c = 30;

public static void main(String[] args){

int a = 10;

variables v = new variables();

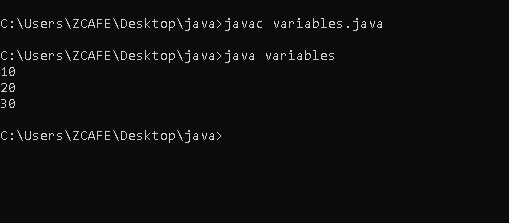
System.out.println(a);

System.out.println(v.b);

System.out.println(variables.c);

}}

**OUTPUT**



**Assignment – 8**

* **Write a program to demonstrate decision statements in Java:**

**1) Else-if ladder**

**2) Nested if**

**INPUT**

import java.util.Scanner;

class chandu{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

if(num > 0){

System.out.println("The number is positive.");}

else if(num < 0){

System.out.println("The number is negative.");}

else{

System.out.println("The number is zero.");}

**2.**

System.out.println("\nEnter your age: ");

int age = sc.nextInt();

if(age >= 18){

System.out.println("You are an Adult.");

if (age >= 60) {

System.out.println("You are a Senior Citizen.");}

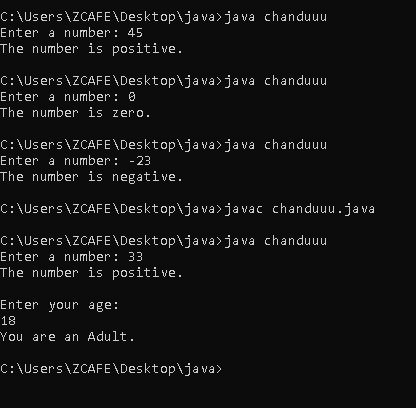
}

else{

System.out.println("You are a Kid.");}

}}

**OUTPUT**



**Assignment – 9**

* **Write a program to demonstrate the control flow-**

**1) to print the sum of 10 natural number (using for, while & do while)**

**2) to determine the odd and even number (using for & while).**

**INPUT**

**1.**

import java.util.Scanner;

class control{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

// sum using for loop

int sum = 0;

int i;

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

sum = sum + i;}

System.out.println("The sum using for loop: "+sum);

//sum using while loop

sum = 0;

i = 1;

while(i <= 10){

sum = sum + i;

i++;}

System.out.println("The sum using while loop: "+sum);

//for loop to determine odd and even numbers

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

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

if (i % 2 == 0) {

System.out.println(i + " is Even");}

else{

System.out.println(i + " is Odd");}

}

//while loop to determine odd and even numbers

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

i = 1;

while (i <= 10){

if(i % 2 == 0) {

System.out.println(i + " is Even");

}

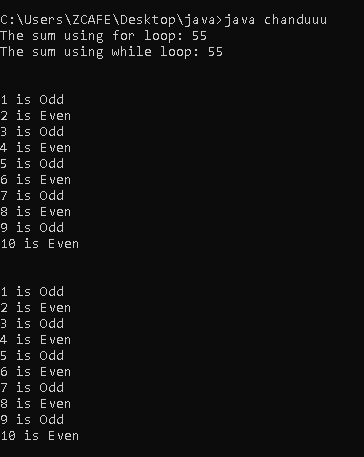
else{

System.out.println(i + " is Odd");}

i++;}

}}

**OUTPUT**



**Assignment – 10**

* **Write a program to demonstrate switch and extended switch.**

**INPUT**

import java.util.Scanner;

class swit{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter a day between 1-7");

int day = sc.nextInt();

switch(day){

case 1:

System.out.println("Monday");

break;

case 2:

System.out.println("Tuesday");

break;

case 3:

System.out.println("Wednesday");

break;

case 4:

System.out.println("Thursday");

break;

case 5:

System.out.println("Friday");

break;

case 6:

System.out.println("Saturday");

break;

case 7:

System.out.println("Sunday");

break;

default:

System.out.println("Wrong Value. Enter between 1-7");

}

// Extended Switch

System.out.println("\nEnter a day between 1-7:");

int din = sc.nextInt();

switch(din){

case 1 -> System.out.println("Monday");

case 2 -> System.out.println("Tuesday");

case 3 -> System.out.println("Wednesday");

case 4 -> System.out.println("Thursday");

case 5 -> System.out.println("Friday");

case 6 -> System.out.println("Saturday");

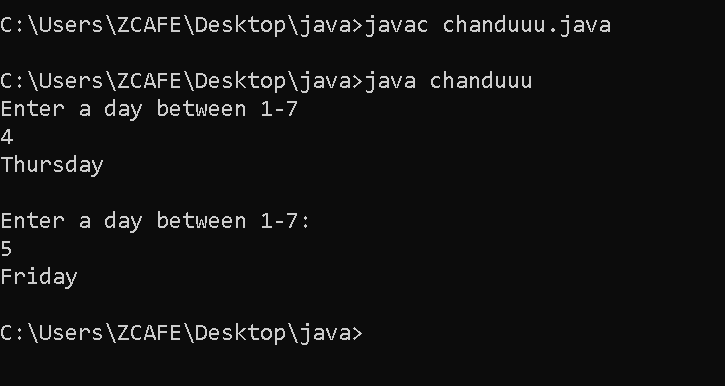
case 7 -> System.out.println("Sunday");

default -> System.out.println("Invalid");

};

}}

**OUTPUT**



**Assignment – 11**

* **Write a program to find the sum of 10 number using array.**

**INPUT**

import java.util.Scanner;

class chanduuu{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int sum = 0;

int[] arr = new int[10];

System.out.println("Enter elements for array:");

for(int i=0;i<10;i++){

arr[i] = sc.nextInt();

}

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

for(int i=0;i<10;i++){

System.out.print(arr[i]+",");

}

for(int i=0;i<10;i++){

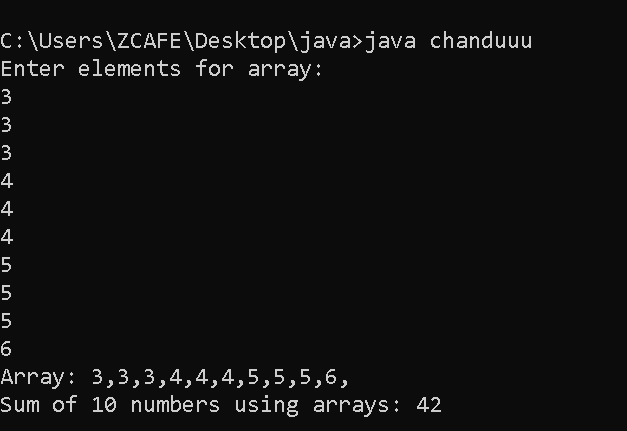
sum = sum + arr[i];

}

System.out.println("\nSum of 10 numbers using arrays: "+sum);

}}

**OUTPUT**



**Assignment – 12**

* **Write a program to generate the following pattern using for loop:**

1. **\***

**\*\***

**\*\*\***

1. **\***

**\* \***

**\* \* \***

**\* \***

**\***

**INPUT**

class star{

public static void main(String[] args){

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

for(int i=1;i<=3;i++){

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

System.out.print("\*");

}

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

}

System.out.println("\nPattern 2");

//upper part

int rows = 3;

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

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

System.out.print(" ");

}

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

System.out.print("\* ");

}

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

}

//lower part

for(int i=1;i<=rows-1;i++){

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

System.out.print(" ");

}

for(int k=i;k<=rows-1;k++){

System.out.print("\* ");

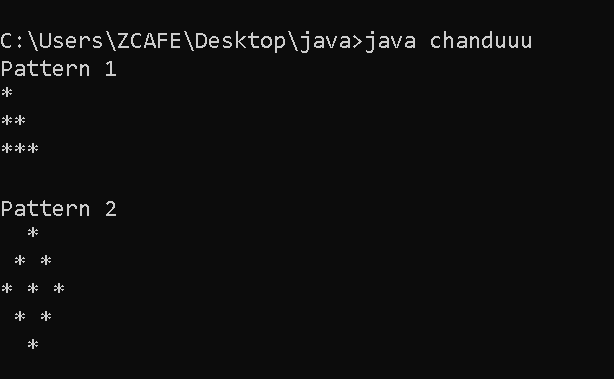
}

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

}

}}

**OUTPUT**



**Assignment – 13**

* **Write a program to check if a number is positive, negative, or zero, and further checks if the positive number is even or odd using nested if and else if ladder.**

**INPUT**

import java.util.Scanner;

class posnev{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

if(num>0){

System.out.println("The Number is Positive");

if(num%2==0){

System.out.println("The Number is Even");}

else{

System.out.println("The Number is Odd");}

}

else if(num<0){

System.out.println("The Number is Negative");

}

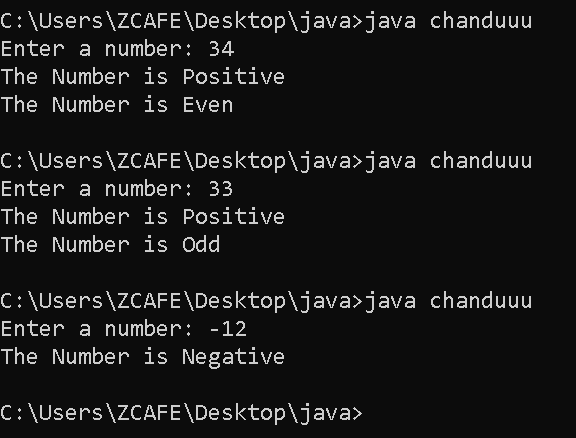
else{

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

}

}}

**OUTPUT**



**Assignment – 14**

* **Write a program to assign a grade based on the marks obtained by a student using nested if and else if ladder.**

**INPUT**

import java.util.Scanner;

class grade{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter marks:");

int marks = sc.nextInt();

if(marks>90)

System.out.println("Grade = A");

else if(marks>80 )

System.out.println("Grade = B");

else if(marks>70)

System.out.println("Grade = C");

else if(marks>60)

System.out.println("Grade = D");

else if(marks>50)

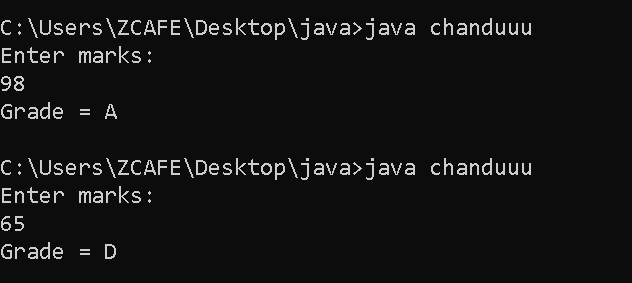
System.out.println("Grade = E");

else

System.out.println("Fail");

}}

**OUTPUT**



**Assignment – 15**

* **Write a program to performs basic arithmetic operations (addition, subtraction, multiplication, division) based on user input using switch case.**

**INPUT**

import java.util.Scanner;

public class operations{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

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

double num1 = sc.nextDouble();

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

double num2 = sc.nextDouble();

System.out.println("Choose operation: +, -, \*, /");

char operator = sc.next().charAt(0);

switch(operator){

case '+':

double add = num1 + num2;

System.out.println("Addition: " +add);

break;

case '-':

double sub = num1 - num2;

System.out.println("Subtraction: " +sub);

break;

case '\*':

double mul = num1 \* num2;

System.out.println("Multiplication: " +mul);

break;

case '/':

if(num2 != 0){

double div = num1 / num2;

System.out.println("Division: " +div);}

else{

System.out.println("Error! Division by zero is not allowed.");

}

break;

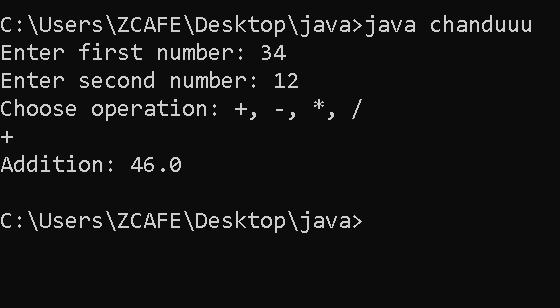
default:

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

}

}}

**OUTPUT**



**Assignment – 16**

* **Write a program to determine the number of days in a given month using enhanced switch case.**

**INPUT**

import java.util.Scanner;

class chanduuu{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.print("Enter a month number(1-12): ");

int month = sc.nextInt();

switch(month){

case 1, 3, 5, 7, 8, 10, 12 -> System.out.println("This month has 31 days");

case 4, 6, 9, 11 -> System.out.println("This month has 30 days");

case 2 -> {

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

int year = sc.nextInt();

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

System.out.println("This month has 29 days");}

else{

System.out.println("This months has 28 days");}

}

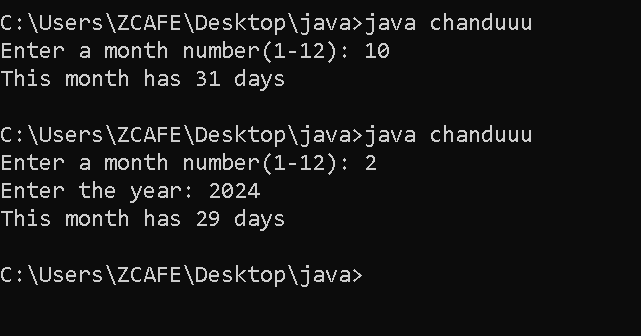
default -> {

System.out.println("Invalid month!");}

}

}}

**OUTPUT**



**Assignment – 17**

* **Write a program that prints all characters in a string except vowels using continue and break statement.**

**INPUT**

import java.util.Scanner;

class vowel{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

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

String s = sc.nextLine();

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

char ch = s.charAt(i);

if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' || ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')

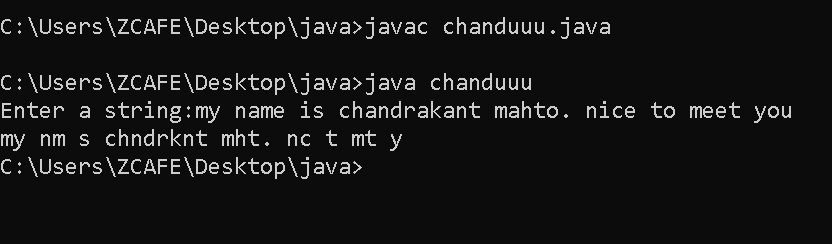
continue;

System.out.print(ch);

}

}}

**OUTPUT**



**Assignment – 18**

* **Write a program that prints numbers from 1 to 100 but skips multiples of 5 using continue.**

**INPUT**

class continu{

public static void main(String[] args){

int i ;

System.out.println("Natural Numbers from 1 - 100 without multiples of 5");

for(i=1;i<=100;i++){

if(i%5==0)

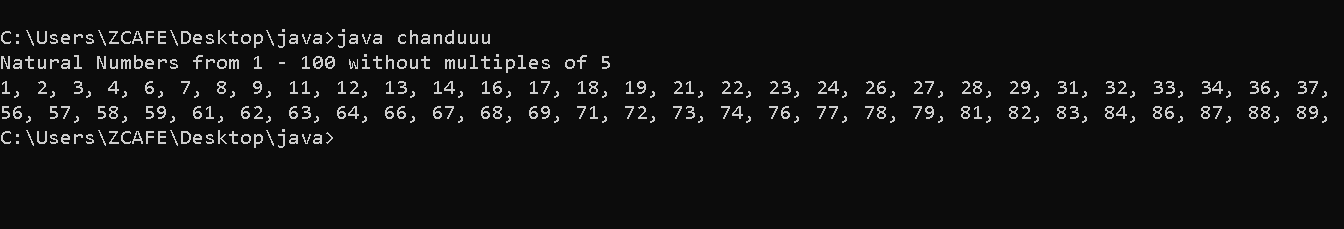
continue;

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

}

}}

**OUTPUT**



**Assignment – 19**

* **Write a program to demonstrate array.fill() method in java.**

**INPUT**

import java.util.Arrays;

class fill{

public static void main(String[] args){

int[] arr = new int[5];

Arrays.fill(arr,3);

System.out.println("Array:");

for(int e : arr)

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

}}

**OUTPUT**



**Assignment – 20**

* **Write a program to print first 8 values with their index position using for loop in java.**

**INPUT**

import java.util.Scanner;

class index{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int[] arr = new int[8];

System.out.println("Enter the elements of array:");

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

arr[i] = sc.nextInt();

}

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

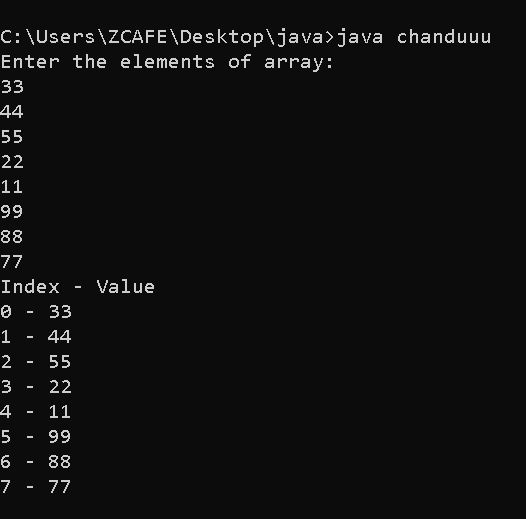
for(int i = 0; i < arr.length; i++){

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

}

}}

**OUTPUT**



**Assignment – 21**

* **Write a program to add two matrices and stores the result in a third matrix.**

**INPUT**

public class matrix{

public static void main(String[] args){

int mat1[][] = {{1,2,3},

{4,5,6},

{7,8,9}};

System.out.println("Matrix-1");

for(int i=0; i<mat1.length; i++){

for(int j=0; j<mat1[i].length; j++){

System.out.print(mat1[i][j]+" ");

}

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

}

System.out.println("");

int mat2[][] = {{10,20,30},

{40,50,60},

{70,80,90}};

System.out.println("Matrix-2");

for(int i=0; i<mat2.length; i++){

for(int j=0; j<mat2[i].length; j++){

System.out.print(mat2[i][j]+" ");

}

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

}

int[][] sum = new int[3][3];

System.out.println("");

System.out.println("Sum of Mat1 and Mat2:");

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

for(int j=0; j<sum[i].length; j++){

sum[i][j] = mat1[i][j] + mat2[i][j];

}

}

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

for(int j=0; j<sum[i].length; j++){

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

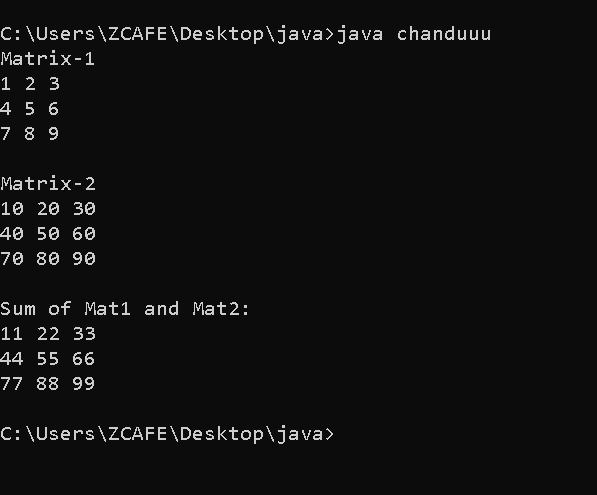
}

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

}

}}

**OUTPUT**



**Assignment – 22**

* **Write a program to search for a specific value in a 2D array.**

**INPUT**

import java.util.Scanner;

class mat1{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int rows, cols, val;

System.out.print("Enter the no. of rows in a matrix:");

rows = sc.nextInt();

System.out.print("Enter the no. of columns in a matrix:");

cols = sc.nextInt();

int[][] mat = new int[rows][cols];

System.out.println("Enter the elements in array: ");

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

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

mat[i][j] = sc.nextInt();

}

}

System.out.println("The Matrix is: ");

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

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

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

}

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

}

System.out.println();

System.out.print("Enter the value to be searched:");

val = sc.nextInt();

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

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

if(val == mat[i][j]){

System.out.println(val+" is found at "+ (i+1) +" row and "+ (j+1) +" column");

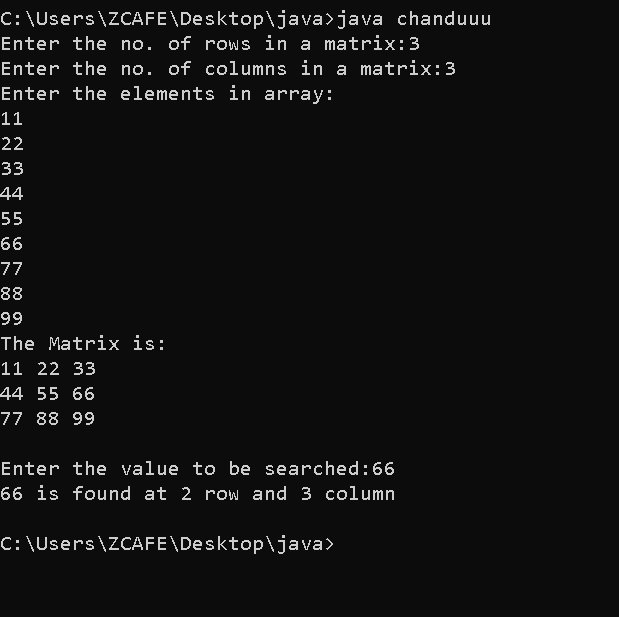
}

}

}

}}

**OUTPUT**



**Assignment – 23**

* **Write a program to find the transpose of a matrix (rows become columns and vice versa).**

**INPUT**

import java.util.Scanner;

public class mat2{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int rows,cols;

System.out.print("Enter the no. of rows in a matrix:");

rows = sc.nextInt();

System.out.print("Enter the no. of columns in a matrix:");

cols = sc.nextInt();

int[][] matrix = new int[rows][cols];

int[][] transpose = new int[cols][rows];

System.out.println("Enter elements of the matrix:");

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

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

matrix[i][j] = sc.nextInt();

}

}

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

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

transpose[j][i] = matrix[i][j];

}

}

System.out.println("Transpose of the matrix:");

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

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

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

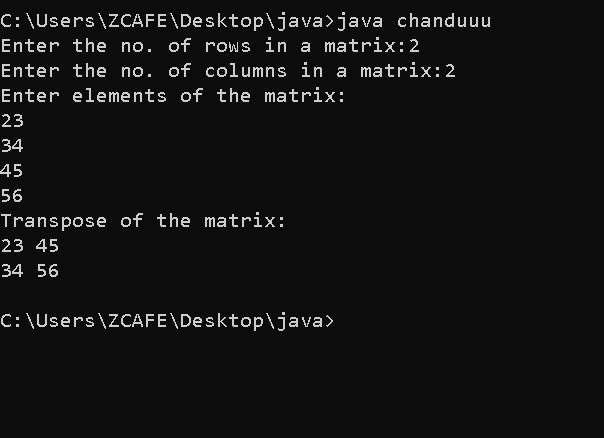
}

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

}

}}

**OUTPUT**



**Assignment – 24**

* **Write a program to create a class, methods, variables and access the variables, methods using object and class**

**Input**

class student{

String name;

int age;

static String college1="MSI College";

void info(){

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

System.out.println("Age:"+age);

}

static void college(){

System.out.println("College: "+college1);

}

}

public class Teststudent{

public static void main(String[] args){

//Creating object

student s1=new student();

s1.name="shinchan";

s1.age=5;

s1.info();

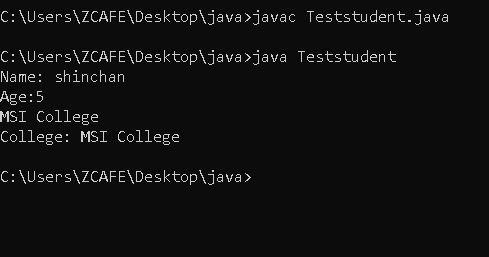
System.out.println(student.college1);

student.college();

}

}

**OUTPUT**



**Assignment – 25**

* **Write a program to demonstrate static variables and non-static variables.**

**Input**

class employee{

String name; //non-static

static String company; //static

}

class TestEmployee1{

public static void main(String[] args){

employee.company="Z-cafe";

employee e1=new employee();

employee e2=new employee();

e1.name="nobita";

e2.name="doraemon";

System.out.println(e1.name);

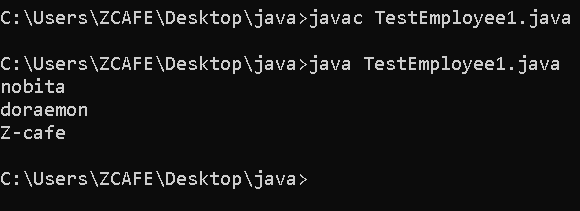
System.out.println(e2.name);

System.out.println(employee.company);

}

}

**OUTPUT**



**Assignment – 26**

* **Write a program to demonstrate static methods and non-static methods.**

**Input**

class Example{

static void staticMethod(){

System.out.println("This is a static method.");

}

void nonstatic(){

System.out.println("This is a non-static method.");

}

}

public class test{

public static void main(String[] args){

Example.staticMethod();

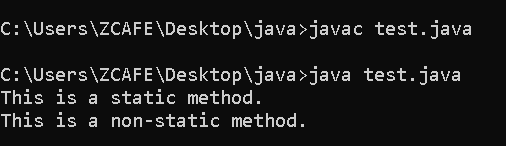
Example e=new Example();

e.nonstatic();

}

}

**OUTPUT**



**Assignment – 27**

* **Write a program to demonstrate static block.**

**INPUT**

class Try1 {

// Static block

static {

System.out.println("Static block is executed before main method.");

}

public static void main(String[] args) {

System.out.println("Main method is executed.");

}

}

**OUTPUT**



**Assignment – 28**

**Write a program to declare two methods and calculate**

**• Area of circle**

**• Area of square**

**INPUT**

class calculator{

void circle(double r){

double area = Math.PI\*r\*r;

System.out.println("Area of Circle="+area);

}

void square(double s){

double area=s\*s;

System.out.println("Area of Square"+area);

}

}

public class TestArea{

public static void main(String[] args){

calculator calc= new calculator();

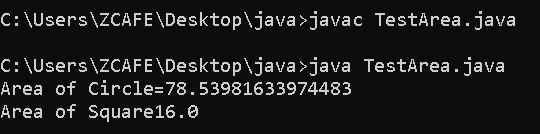
calc.circle(5.0);

calc.square(4.0);

}

}

**OUTPUT**



**Assignment – 29**

* **Write a program to display all the types of inheritance such as single, multilevel and hierarchical.**

**Input**

class Animal{

void eat(){

System.out.println("Animal eats food.");

}

}

class Dog extends Animal{

void bark(){

System.out.println("Dog barks.");

}

}

class puppy extends Dog{

void weep(){

System.out.println("puppy weep");

}

}

class Cat extends Animal{

void meow(){

System.out.println("Cat meows.");

}

}

public class Inheritance{

public static void main(String[] args){

Dog d=new Dog();

d.eat();

d.bark();

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

puppy p=new puppy();

p.eat();

p.bark();

p.weep();

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

Cat c=new Cat();

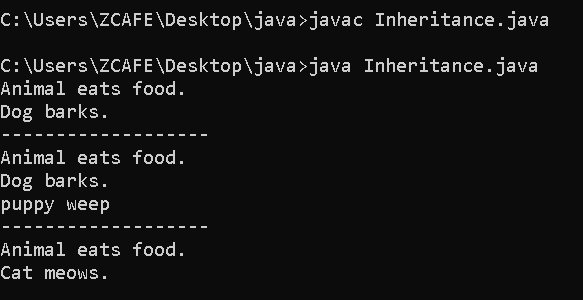
c.eat();

c.meow();

}

}

**OUTPUT**



**Assignment – 30**

* **Write a program to demonstrate final method and class.**

**INPUT**

final class vehicle{

final void display(){

System.out.println("This is a final method inside a final class.");

}

}

//class car extends vehicle{}//Error

public class TestFinal{

public static void main(String[] args){

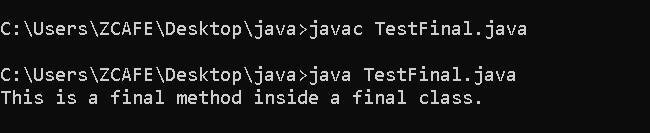
vehicle v =new vehicle();

v.display();

}

}

**OUTPUT**



**Assignment – 31**

* **Write a program to implement package in Java**

**INPUT**

import java.lang.\*;

class abc1

{

public static void main(String[] args)

{

double radius = 5.0;

double area = Math.PI \* Math.pow(radius, 2);

System.out.println("Area: " + area);

String message = "Hello, World!";

int length = message.length();

System.out.println("Length: " + length);

int number = 42;

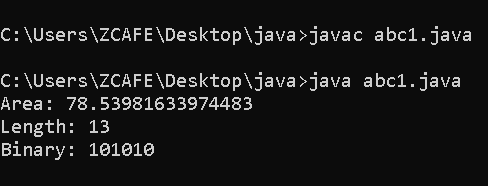
String binary = Integer.toBinaryString(number);

System.out.println("Binary: " + binary);

}

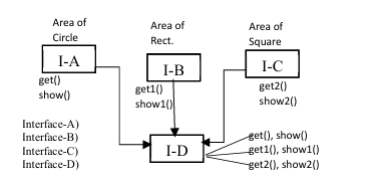
}

**OUTPUT**



**Assignment – 32**

* **Write a program to demonstrate Interface to Interface in java**



interface IA{

void get();

void show();

}

interface IB{

void get1();

void show1();

}

interface IC{

void get2();

void show2();

}

interface ID extends IA,IB,IC{

}

class calculate implements ID{

double radius, length, breadth, side;

public void get(){

radius = 5.0;

}

public void show(){

double area = Math.PI\*radius\*radius;

System.out.println("Area of circle="+area);

}

public void get1(){

length = 4.0;

breadth= 3.0;

}

public void show1(){

double area = length\*breadth;

System.out.println("Area of Rectangle ="+area);

}

public void get2(){

side=6.0;

}

public void show2(){

double area = side \* side;

System.out.println("Area of Square ="+ area);

}

}

public class testing{

public static void main(String[] args){

calculate obj= new calculate();

obj.get();

obj.show();

obj.get1();

obj.show1();

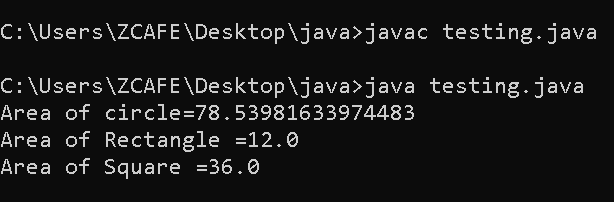
obj.get2();

obj.show2();

}

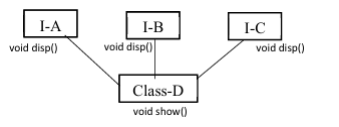
}

**OUTPUT**



**Assignment – 33**

* **Write a program to demonstrate interface to class in java**



interface IA{

void disp();

}

interface IB{

void disp();

}

interface IC{

void disp();

}

class D implements IA,IB,IC{

public void disp(){

System.out.println("display A,B,C");

}

public void show(){

System.out.println("Show D");

}

}

public class ItoC{

public static void main(String[] args){

D obj = new D();

obj.disp();

obj.show();

}

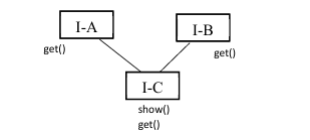
}

**OUTPUT**



**Assignment – 34**

* **Write a program to demonstrate multiple inheritance in java.**



interface IA{

void get();

}

interface IB{

void get();

}

interface IC extends IA,IB{

void show();

}

class abc implements IC{

public void get(){

System.out.println("Get method from interfaces A and B");

}

public void show(){

System.out.println("Show method From interface C");

}

}

public class inheri{

public static void main(String[] args){

abc obj=new abc();

obj.get();

obj.show();

}

}

**OUTPUT**



**Assignment – 35**

* **Write a program to demonstrate compile time and run time polymorphism.**

**INPUT**

class compile{

void add(int a, int b){

System.out.println("Addition of two integers:"+(a+b));

}

void add(double a,double b){

System.out.println("Addition of two doubles:"+(a+b));

}

}

class Animal{

void sound(){

System.out.println("Animal makes a sound");

}

}

class dog extends Animal{

void sound(){

System.out.println("Dog makes a sound");

}

}

public class poly{

public static void main(String[] args){

compile obj = new compile();

obj.add(10,20);

obj.add(3.5,3.5);

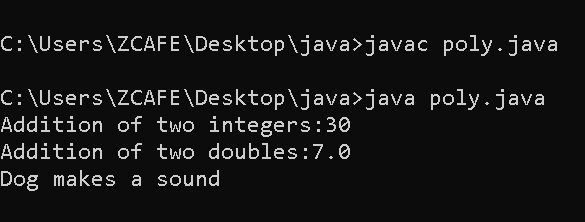
Animal a;

a=new dog();

a.sound();

}}

**OUTPUT**



**Assignment – 36**

* **Write a program to demonstrate instance of operator.**

**INPUT**

class Animal{}

class Dog extends Animal{}

public class inst{

public static void main(String[] args){

Dog d = new Dog();

System.out.println(d instanceof Dog);

System.out.println(d instanceof Animal);

System.out.println(d instanceof Object);

Animal a=new Animal();

System.out.println(a instanceof Dog);

}

}

**OUTPUT**



**Assignment – 37**

**Write a program to create a new file and perform the following methods:**

**a) canread**

**b) delete**

**c) exists**

**d) length**

**e) getname**

**f) get absolute path**

**g) renameto**

**INPUT**

import java.io.\*;

class finfo {

public static void main(String[] args) {

File f = new File("C:\\Users\\ZCAFE\\Desktop\\file.txt");

if (f.exists()) {

System.out.println("File name is: " + f.getName());

System.out.println("File location is: " + f.getAbsolutePath());

System.out.println("File writable: " + f.canWrite());

System.out.println("File readable: " + f.canRead());

System.out.println("File size (bytes): " + f.length());

// Rename the file

File renamedFile = new File("C:\\Users\\ZCAFE\\Desktop\\file06.txt");

if (f.renameTo(renamedFile)) {

System.out.println("File successfully renamed to: " + renamedFile.getName());

// Delete the renamed file

System.out.println("File removed: " + renamedFile.delete());

} else {

System.out.println("Failed to rename the file.");

}

} else {

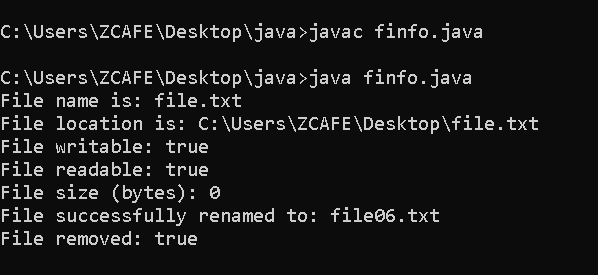
System.out.println("File doesn't exist.");

}

}

}

**OUTPUT**



**Assignment – 38**

**Write a program to implement filereader and filewriter methods in Java.**

**INPUT**

import java.io.\*;

class info {

public static void main(String[] args) {

// Writing to the file

try {

FileWriter f = new FileWriter("C:\\Users\\ZCAFE\\Desktop\\hi.txt");

try {

f.write("Java Programming is the best language....");

} finally {

f.close();

System.out.println("Successfully wrote to the file.");

}

} catch (IOException i) {

System.out.println("Writing Error: " + i);

}

// Reading from the file

try {

FileReader r = new FileReader("C:\\Users\\ZCAFE\\Desktop\\hi.txt");

try {

int i;

while ((i = r.read()) != -1) {

System.out.print((char) i);

}

} finally {

r.close();

System.out.println("\nFile closed.");

}

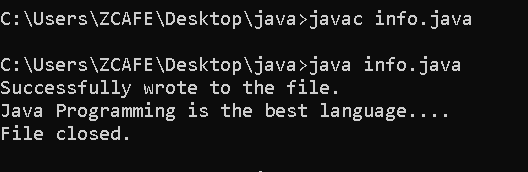
} catch (IOException e) {

System.out.println("Reading Error: " + e);

}

}}

**OUTPUT**



**Assignment – 39**

**Write a program to copy the content of file and paste into another file.**

**INPUT**

import java.io.\*;

class inffo

{

public static void main(String[] args)throws IOException

{

FileInputStream r=new FileInputStream("C:\\Users\\ZCAFE\\Desktop\\hi.txt");

FileOutputStream w=new FileOutputStream("C:\\Users\\ZCAFE\\Desktop\\hi2.txt ");

int i;

while((i=r.read())!=-1)

{

w.write((char)i);

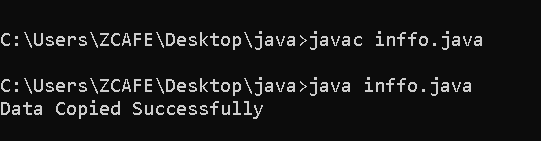
}

System.out.println("Data Copied Successfully ");

}

}

**OUTPUT**



**Assignment – 40**

**Write a program to implement exception handling using try & catch block**

**a) IOException**

**b) Arithmetic Exception**

**c) Null pointer Exception**

import java.io.\*;

class Exception{

public static void main(String[] args) {

// a) IOException Example

try {

FileReader fr = new FileReader("non\_existing\_file.txt"); // file doesn't exist

fr.read();

fr.close();

} catch (IOException e) {

System.out.println("IOException caught: " + e);

}

// b) ArithmeticException Example

try {

int a = 10 / 0; // division by zero

} catch (ArithmeticException e) {

System.out.println("ArithmeticException caught: " + e);

}

// c) NullPointerException Example

try {

String str = null;

System.out.println(str.length()); // null string access

} catch (NullPointerException e) {

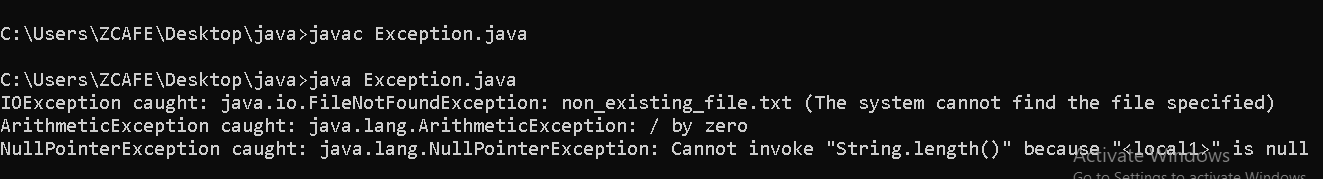
System.out.println("NullPointerException caught: " + e);

}

}

}

**OUTPUT**



**Assignment-41**

**Q. Write a program to demonstrate exception using throws and same can be managed by try, catch, and finally block.**

**Ans.**

import java.io.\*;

public class Throws {

public static void readFile() throws IOException {

FileReader file = new FileReader("data.txt");

BufferedReader reader = new BufferedReader(file);

System.out.println(reader.readLine());

reader.close();

}

public static void main(String[] args) {

try {

readFile();

} catch (IOException e) {

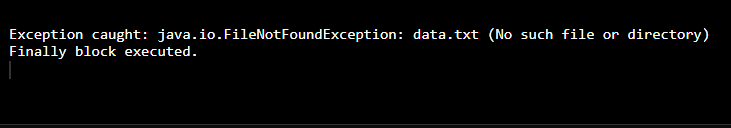
System.out.println("Exception caught: " + e);

} finally {

System.out.println("Finally block executed.");

} } }

**Output**

****

**Assignment-42**

**Q. Write a program to handle exception with multiple try and catch block.**

**Ans.**

public class MultipleTryCatch{

public static void main(String[] args) {

try {

int a = 10, b = 0;

int result = a / b;

System.out.println("Result: " + result);

} catch (ArithmeticException e) {

System.out.println("ArithmeticException caught: " + e);

}

try {

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

System.out.println(arr[5]);

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("ArrayIndexOutOfBoundsException caught: " + e);

}

try {

String str = null;

System.out.println(str.length());

} catch (NullPointerException e) {

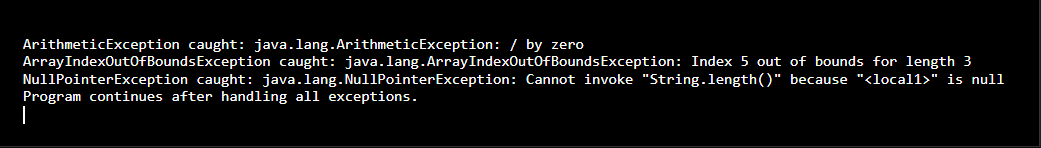
System.out.println("NullPointerException caught: " + e);

}

System.out.println("Program continues after handling all exceptions.");

} }

**Output**

****

**Assignment-43**

**Q. Write a program to handle exception with one try and multiple catch block: a)nullpointer exception**

**b)Arithmetic exception**

**c)Numberformat exception**

**d)Stringindexoutofbound exception**

**Ans.**

public class MultipleCatchDemo {

public static void main(String[] args) {

try {

// a) NullPointerException

String str = null;

System.out.println(str.length());

// b) ArithmeticException

// int a = 10 / 0;

// c) NumberFormatException

// int num = Integer.parseInt("abc");

// d) StringIndexOutOfBoundsException

// String name = "Java";

// System.out.println(name.charAt(10));

} catch (NullPointerException e) {

System.out.println("Caught NullPointerException: " + e);

} catch (ArithmeticException e) {

System.out.println("Caught ArithmeticException: " + e);

} catch (NumberFormatException e) {

System.out.println("Caught NumberFormatException: " + e);

} catch (StringIndexOutOfBoundsException e) {

System.out.println("Caught StringIndexOutOfBoundsException: " + e);

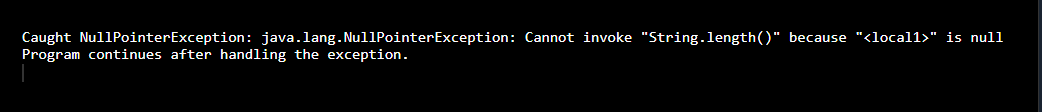
}

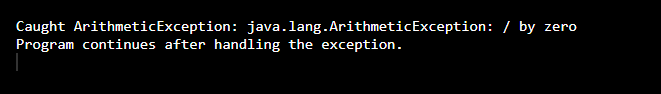
System.out.println("Program continues after handling the exception.");

}

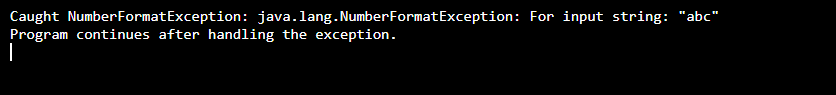
}

**Output**

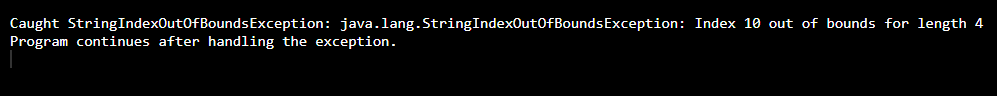
1. **NullPointerExecption-**
2. **ArithmeticException**

****

1. **NumberFormatException**



1. **StringIndexOutOfBoundException**

****

**Assignment-44**

**Q. Write a program to demonstrate the thread class by using extend.**

**Ans.**

class MyThread extends Thread {

public void run() {

for (int i = 1; i <= 3; i++) {

System.out.println("Thread running: " + i);

try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.out.println("Thread interrupted: " + e);

}}}

public class MyMainClass {

public static void main(String[] args) {

MyThread t1 = new MyThread();

t1.start();

for (int i = 1; i <= 3; i++) {

System.out.println("Main thread: " + i);

try {

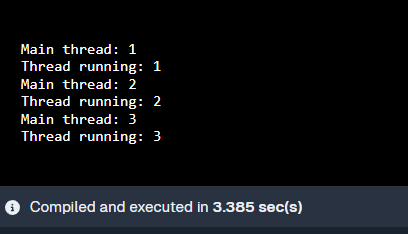
Thread.sleep(500);

} catch (InterruptedException e) {

System.out.println("Main thread interrupted: " + e);

}}}}

**Output**

****

**Assignment-45**

**Q. Write a program to demonstrate extending thread class using throws exception.**

**Ans.**

class MyThread extends Thread {

public void riskyMethod() throws Exception {

throw new Exception("Something went wrong in riskyMethod()");

}

public void run() {

try {

System.out.println("Thread is running...");

riskyMethod();

} catch (Exception e) {

System.out.println("Caught exception: " + e.getMessage());

} } }

public class ThreadWithThrows {

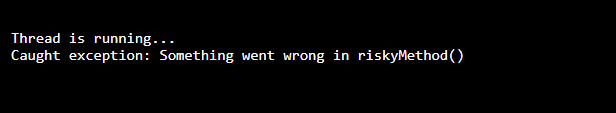
public static void main(String[] args) {

MyThread t1 = new MyThread();

t1.start();

} }

**Output**

****

**Assignment-46**

**Q. Write a program to implementing runnable interface in multithreading.**

**Ans.**

class MyRunnable implements Runnable {

public void run() {

System.out.println("Thread is running...");

}

}

public class Main {

public static void main(String[] args) {

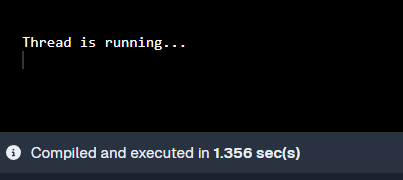
Thread t = new Thread(new MyRunnable());

t.start();

}

}

**Output**

****

**Assignment-47**

**Q. Write a program to create thread in java and run thread parallel.**

**Ans.**

class MyRunnable implements Runnable {

private final String name;

public MyRunnable(String name) {

this.name = name;

}

public void run() {

for (int i = 1; i <= 3; i++) {

System.out.println(name + “ – count: “ + i);

try {

Thread.sleep(200);

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

} } } }

public class ParallelThreads {

public static void main(String[] args) {

Thread t1 = new Thread(new MyRunnable(“Thread A”));

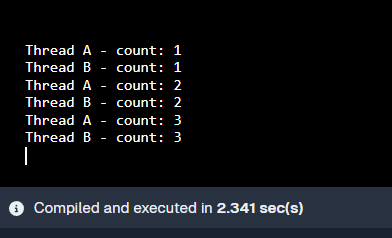
Thread t2 = new Thread(new MyRunnable(“Thread B”));

t1.start();

t2.start();

} }

**Output**

****