**1. Write a program in java to find the sum of the digits of the given integer.**

→

import java.util.Scanner;

class sum

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n, num, sum=0, i=0;

System.out.print("Enter any integer : ");

num = sc.nextInt();

while(num>0)

{

n = num % 10;

sum = sum + n;

num = num / 10;

}

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

}

}

**o/p :**

Enter any integer : 69769420

Sum of digits of integer : 43

**2. Write a program in java to obtain the Gross salary by calculating DA & HRA.**

→

import java.util.Scanner

class salary

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

float bs, gs, da, hra, d, h;

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

bs = sc.nextFloat();

System.out.print("Enter rate of dearness allowance (in %) : ");

da = sc.nextFloat();

System.out.print("Enter rate of housing rate allowance (in %) : ");

hra = sc.nextFloat();

d = bs \* (da/100);

h = bs \* (hra/100);

gs = bs + h + d;

System.out.print("Gross salary : "+gs);

}

}

**o/p :**

Enter basic salary : 60000

Enter rate of dearness allowance (in %) : 7.1

Enter rate of housing rate allowance (in %) : 4.2

Gross salary : 66780

**3. Write a program in java to convert kilometers into meters.**

→

import java.util.Scanner;

class convert

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int km, m;

System.out.print("Enter distance in kilometer : ");

km = sc.nextInt();

m = km \* 1000;

System.out.print("Distance in meter : "+m);

}

}

**o/p :**

Enter distance in kilometer : 54

Distance in meter : 54000

**4. Write a program in java to convert temperature from Fahrenheit to Celsius using the formula “ c = ( f - 32 ) \* 5/9 ”.**

→

import java.util.Scanner;

class convert

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int f, c;

System.out.print("Enter temperature in fahrenheit : ");

f = sc.nextInt();

c = ( f - 32 ) \* 5/9 ;

System.out.print("Temperature in celsius : "+c);

}

}

**o/p :**

Enter temperature in fahrenheit : 32

Temperature in celsius : 0

**5. Write a program in java to find the highest of three numbers by using If loop.**

→

import java.util.Scanner;

public class highest

{

public static void main(String []args)

{

Scanner sc = new Scanner(System.in);

int num1, num2, num3;

System.out.print("Enter num 1 : ");

num1 = sc.nextInt();

System.out.print("Enter num 2 : ");

num2 = sc.nextInt();

System.out.print("Enter num 3 : ");

num3 = sc.nextInt();

if(num1>num2 && num1>num3)

{

System.out.print(num1+" is the highest of "+num1+", "+num2+" and "+num3);

}

if(num2>num1 && num2>num3)

{

System.out.print(num2+" is the highest of "+num1+", "+num2+" and "+num3);

}

if(num3>num1 && num3>num2)

{

System.out.print(num3+" is the highest of "+num1+", "+num2+" and "+num3);

}

}

}

**o/p :**

Enter num 1 : 32

Enter num 2 : 41

Enter num 3 : 21

41 is the highest of 32, 41 and 21

**6. Write a program in java to find the highest of three numbers by using If-Else loop.**

→

import java.util.Scanner;

public class highest

{

public static void main(String []args)

{

Scanner sc = new Scanner(System.in);

int num1, num2, num3;

System.out.print("Enter num 1 : ");

num1 = sc.nextInt();

System.out.print("Enter num 2 : ");

num2 = sc.nextInt();

System.out.print("Enter num 3 : ");

num3 = sc.nextInt();

if(num1>num2 && num1>num3)

{

System.out.print(num1+" is the highest of "+num1+", "+num2+" and "+num3);

}

else if(num2>num1 && num2>num3)

{

System.out.print(num2+" is the highest of "+num1+", "+num2+" and "+num3);

}

else

{

System.out.print(num3+" is the highest of "+num1+", "+num2+" and "+num3);

}

}

}

**o/p :**

Enter num 1 : 54

Enter num 2 : 66

Enter num 3 : 200

200 is the highest of 54, 66 and 200

**7. Write a program in java to find the highest of three numbers and give the output in ascending order.**

→

import java.util.Scanner;

public class HighestOfThreeNumbers

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

int num1 = scanner.nextInt();

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

int num2 = scanner.nextInt();

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

int num3 = scanner.nextInt();

int highest = num1;

if (num2 > highest)

{

highest = num2;

}

if (num3 > highest)

{

highest = num3;

}

int lowest = num1;

int middle = num2;

if (num1 > num2)

{

lowest = num2;

middle = num1;

}

if (middle > num3)

{

if (num3 > lowest)

{

middle = num3;

}

else

{

middle = lowest;

lowest = num3;

}

}

System.out.println("Numbers in ascending order : " + lowest + ", " + middle + ", " + highest);

}

}

**o/p :**

Enter the first number : 45

Enter the second number : 27

Enter the third number : 89

Numbers in ascending order : 27, 45, 89

**8. Write a program in java to find the sum and average of ten numbers by using For loop.**

→

import java.util.Scanner;

public class SumAndAverage

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

int sum = 0;

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

{

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

int num = scanner.nextInt();

sum += num;

}

double average = (double) sum / 10;

System.out.println("Sum of the ten numbers is : " + sum);

System.out.println("Average of the ten numbers is : " + average);

}

}

**o/p :**

Enter number 1 : 4

Enter number 2 : 8

Enter number 3 : 2

Enter number 4 : 9

Enter number 5 : 5

Enter number 6 : 1

Enter number 7 : 7

Enter number 8 : 3

Enter number 9 : 6

Enter number 10 : 0

Sum of the ten numbers is : 45

Average of the ten numbers is : 4.5

**9. Write a program in java to find if the given number is Armstrong or not.**

→

import java.util.Scanner;

public class ArmstrongNumber

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

int sum = 0;

int originalNum = num;

int numOfDigits = String.valueOf(num).length();

while (num != 0)

{

int digit = num % 10;

sum += Math.pow(digit, numOfDigits);

num /= 10;

}

if (originalNum == sum)

{

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

}

else

{

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

}

}

}

**o/p :**

Enter a number : 123

123 is not an Armstrong number.

**10. Write a program in java to generate the series of even numbers solutions.**

→

public class EvenNumbers

{

public static void main(String[] args)

{

int limit = 20; // Define the limit of the series

System.out.print("Even numbers from 1 to " + limit + " : ");

for (int i = 1; i <= limit; i++)

{

// Check if the number is even

if (i % 2 == 0)

{

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

}

}

}

}

**o/p :**

Even numbers from 1 to 20 : 2 4 6 8 10 12 14 16 18 20

**11. Write a program in java to generate factorials of a given number.**

→

import java.util.Scanner;

public class Factorial

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int factorial = 1;

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

{

factorial \*= i;

}

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

}

}

**o/p :**

Enter a number : 5

Factorial of 5 is 120

**12. Write a program in java to generate Fibonacci series.**

→

import java.util.Scanner;

public class Fibonacci

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int firstTerm = 0, secondTerm = 1, nextTerm;

System.out.print("Fibonacci Series : ");

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

{

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

nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

}

}

}

**o/p :**

Enter the number of terms : 10

Fibonacci Series : 0 1 1 2 3 5 8 13 21 34

**13. Write a program in java to find the sum of the N natural numbers.**

→

import java.util.Scanner;

public class SumOfNaturalNumbers

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt()

int sum = 0;

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

{

sum += i;

}

System.out.println("Sum of first " + n + " natural numbers is " + sum);

scanner.close();

}

}

**o/p :**

Enter the value of N : 5

Sum of first 5 natural numbers is : 15

**14. Write a program in java to generate the series of odd numbers solutions.**

→

import java.util.Scanner;

public class OddNumbers

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("Enter the number of odd numbers you want to generate: ");

int n = input.nextInt();

int[] odd\_nums = new int[n];

for (int i = 1, j = 0; j < n; i += 2, j++)

{

odd\_nums[j] = i;

}

System.out.print("The first " + n + " odd numbers are: ");

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

{

System.out.print(odd\_nums[i] + " ");

}

}

}

**o/p :**

Enter the number of odd numbers you want to generate: 7

The first 7 odd numbers are: 1 3 5 7 9 11 13

**15. Write a program in java to generate the following pattern :**

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

→

import java.io.\*;

public class pattern

{

public static void main(String args[])

{

int i, j;

for(i=1; i<7; i++)

{

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

{

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

}

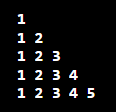
System.out.print("\n");

}

}

}

**o/p :**



**16. Write a program in java to generate the following pattern :**

**1 2 3 4 5**

**1 2 3 4**

**1 2 3**

**1 2**

**1**

→

import java.util.Scanner;

public class pattern

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int i, j;

for(i=5; i>0; i--)

{

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

{

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

}

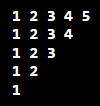
System.out.print("\n");

}

}

}

**o/p :**



**17. Write a program in java to generate the following pattern :**

**\* \* \* \* \***

**\* \* \* \***

**\* \* \***

**\* \***

**\***

→

import java.util.Scanner;

public class pattern

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int i, j;

for(i=5; i>0; i--)

{

for(j=0; j<i; j++)

{

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

}

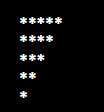
System.out.print("\n");

}

}

}

**o/p :**



**18. Write a program in java to generate the following pattern :**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**\* \* \* \* \***

→

import java.io.\*;

public class pattern

{

public static void main(String args[])

{

int i, j;

for(i=0; i<=5; i++)

{

for(j=0; j<i; j++)

{

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

}

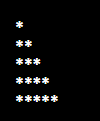
System.out.print("\n");

}

}

}

**o/p :**



**19. Write a program in java to generate the following pattern :**

**A**

**A B**

**A B C**

**A B C D**

**A B C D E**

→

public class Pattern {

public static void main(String[] args) {

char lastChar = 'E'; // Set the last character of the pattern

for (char i = 'A'; i <= lastChar; i++) {

for (char j = 'A'; j <= i; j++) {

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

}

System.out.println();

}

}

}

**o/p :**

A

A B

A B C

A B C D

A B C D E

**20. Write a program in java to generate the following pattern :**

**1**

**2 1 2**

**3 2 1 2 3**

**4 3 2 1 2 3 4**

**5 4 3 2 1 2 3 4 5**

→

public class PatternExample

{

public static void main(String[] args)

{

int rows = 5;

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

{

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

{

System.out.print(" ");

}

for (int j = i; j >= 1; j--)

{

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

}

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

{

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

}

System.out.println();

}

}

}

**o/p :**

1

2 1 2

3 2 1 2 3

4 3 2 1 2 3 4

5 4 3 2 1 2 3 4 5

**21. Write a program in java to find the reverse of the number.**

→

import java.util.Scanner;

public class ReverseNumber

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

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

int number = input.nextInt();

int reversedNumber = 0;

while (number != 0)

{

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

}

System.out.println("The reverse of the number is : " + reversedNumber);

}

}

**o/p :**

Enter a number to reverse : 12345

The reverse of the number is : 54321

**22. Write a program in java to find the sum of even and odd numbers.**

→

import java.util.Scanner;

public class SumOfEvenAndOddNumbers

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("Enter a positive integer : ");

int n = input.nextInt();

int evenSum = 0;

int oddSum = 0;

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

{

if (i % 2 == 0)

{

evenSum += i;

}

else

{

oddSum += i;

}

}

System.out.println("The sum of even numbers up to " + n + " is : " + evenSum);

System.out.println("The sum of odd numbers up to " + n + " is : " + oddSum);

}

}

**o/p :**

Enter a positive integer : 10

The sum of even numbers up to 10 is : 30

The sum of odd numbers up to 10 is : 25

**23. Write a program in java to generate a table of a number.**

→

import java.util.Scanner;

public class NumberTable

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

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

int number = input.nextInt();

System.out.println("Table of " + number + " :");

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

{

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

}

}

}

**o/p :**

Enter a number : 7

Table of 7 :

7 x 1 = 7

7 x 2 = 14

7 x 3 = 21

7 x 4 = 28

7 x 5 = 35

7 x 6 = 42

7 x 7 = 49

7 x 8 = 56

7 x 9 = 63

7 x 10 = 70

**24. Write a program in java to display all the characters represented by the ASCII numbers from 25 to100.**

→

public class ASCIITable

{

public static void main(String[] args)

{

for (int i = 25; i <= 100; i++)

{

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

}

}

}

**o/p :**

! " # $ % & ' ( ) \* + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ \_ ` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~

**25. Write a program in java to find the greatest common divisor of the entered number n.**

→

import java.util.Scanner;

public class GCD

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a positive integer : ");

int n = scanner.nextInt();

System.out.print("Enter another positive integer : ");

int m = scanner.nextInt();

int gcd = findGCD(n, m);

System.out.println("The greatest common divisor of " + n + " and " + m + " is " + gcd);

}

public static int findGCD(int n, int m)

{

if (m == 0)

{

return n;

}

else

{

return findGCD(m, n % m);

}

}

}

**o/p :**

Enter a positive integer : 24

Enter another positive integer : 36

The greatest common divisor of 24 and 36 is 12

**26. Write a program in java to demonstrate the octal & hexadecimal representation of an integer number.**

→

public class NumberRepresentation

{

public static void main(String[] args)

{

int num = 123456;

String octal = Integer.toOctalString(num);

System.out.println("Octal representation of " + num + " is " + octal);

String hexadecimal = Integer.toHexString(num);

System.out.println("Hexadecimal representation of " + num + " is " + hexadecimal);

}

}

**o/p :**

Octal representation of 123456 is 361100

Hexadecimal representation of 123456 is 1e240

**27. Write a program in java to demonstrate the use of shift operators.**

→

public class ShiftOperators

{

public static void main(String[] args)

{

int num1 = 6;

int num2 = 9;

int num1LeftShifted = num1 << 2;

System.out.println(num1 + " << 2 = " + num1LeftShifted);

int num2RightShifted = num2 >> 2;

System.out.println(num2 + " >> 2 = " + num2RightShifted);

int num2ZeroFillRightShifted = num2 >>> 2;

System.out.println(num2 + " >>> 2 = " + num2ZeroFillRightShifted);

}

}

**o/p :**

6 << 2 = 24

9 >> 2 = 2

9 >>> 2 = 2

**28. Write a program in java to find a five digit number which on multiplication by 4 reverses its order.**

→

public class ReverseOrder

{

public static void main(String[] args)

{

for (int i = 10000; i <= 99999; i++)

{

int reverse = reverse(i);

if (i \* 4 == reverse)

{

System.out.println(i);

break;

}

}

}

private static int reverse(int num)

{

int reverse = 0;

while (num != 0)

{

int digit = num % 10;

reverse = reverse \* 10 + digit;

num /= 10;

}

return reverse;

}

}

**o/p :**

21780

**29. Write a program in java to find all four digit perfect squares, where the number formed by the first two digits and number formed by the last two digits are also perfect.**

→

public class PerfectSquares

{

public static void main(String[] args)

{

for (int i = 32; i < 100; i++)

{

for (int j = 1; j < 100; j++)

{

int num = i \* 100 + j;

if (isPerfectSquare(num) && isPerfectSquare(i) && isPerfectSquare(j))

{

System.out.println(num);

}

}

}

}

private static boolean isPerfectSquare(int num)

{

int sqrt = (int) Math.sqrt(num);

return sqrt \* sqrt == num;

}

}

**o/p :**

1444

1936

3249

4761

6084

8464

8836

**30. Write a program in java to create a Floyd triangle of numbers :**

**1**

**2 3 4 5 6 7 8 9 10 2**

**3 4 5 6 7 8 9 10 3**

**4 5 6 7 8 9 10 4**

**...**

**10 10**

→

public class FloydTriangle

{

public static void main(String[] args)

{

int n = 10;

int count = 1;

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

{

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

{

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

count++;

if (count > 10)

{

count = 2;

}

}

System.out.println();

}

}

}

**o/p :**

1

2 3

4 5 6

7 8 9 10

2 3 4 5 6

7 8 9 10 2 3

4 5 6 7 8 9 10

2 3 4 5 6 7 8 9 10 2 3 4 5 6 7 8 9

10 2 3 4 5 6 7 8 9 10

**31. Write a program in java to print I,J,K,L such that I <J<K<L , L=I+J+K where I,J,K <100 .**

→

public class PrintIJKL

{

public static void main(String[] args)

{

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

{

for (int j = i + 1; j < 100; j++)

{

for (int k = j + 1; k < 100; k++)

{

int l = i + j + k;

if (l < 100 && i < j && j < k && k < l)

{

System.out.println("I = " + i + ", J = " + j + ", K = " + k + ", L = " + l);

}

}

}

}

}

}

**o/p :**

I = 1, J = 2, K = 3, L = 6

I = 1, J = 2, K = 4, L = 7

.

.

I = 1, J = 2, K = 96, L = 99

.

.

I = 1, J = 4, K = 9, L = 14

**32. Write a program in java to print 4 digit numbers into words.**

→

import java.util.Scanner;

public class NumberToWords

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

String[] ones = {"", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"};

String[] tens = {"", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"};

String[] teens = {"ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"};

String[] thousands = {"", "thousand"};

int[] digits = new int[4];

for (int i = 0; i < 4; i++)

{

digits[i] = num % 10;

num /= 10;

}

System.out.print("In words: ");

for (int i = 3; i >= 0; i--)

{

if (i == 3)

{

System.out.print(ones[digits[i]] + " thousand ");

}

else if (i == 2)

{

System.out.print(ones[digits[i]] + " hundred ");

}

else if (i == 1)

{

if (digits[i] == 1)

{

System.out.print(teens[digits[i - 1]] + " ");

break;

}

else

{

System.out.print(tens[digits[i]] + " ");

}

}

else if (i == 0)

{

if (digits[i + 1] == 1)

{

continue;

}

else

{

System.out.print(ones[digits[i]]);

}

}

}

}

}

**o/p :**

Enter a four-digit number : 1234

In words: one thousand two hundred thirty four

**33. Write a program in java to print the highest & lowest number from an array.**

→

public class HighLowArray

{

public static void main(String[] args)

{

int[] arr = { 5, 10, 2, 8, 3 };

int min = arr[0];

int max = arr[0];

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

{

if (arr[i] < min)

{

min = arr[i];

}

if (arr[i] > max)

{

max = arr[i];

}

}

System.out.println("Lowest number : " + min);

System.out.println("Highest number : " + max);

}

}

**o/p :**

Lowest number : 2

Highest number : 10

**34. Write a program in java to implement different sorting techniques.**

→

public class SortingAlgorithms

{

// Bubble sort

public static void bubbleSort(int[] arr)

{

int n = arr.length;

for (int i = 0; i < n-1; i++)

{

for (int j = 0; j < n-i-1; j++)

{

if (arr[j] > arr[j+1])

{

// swap arr[j+1] and arr[j]

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

// Selection sort

public static void selectionSort(int[] arr)

{

int n = arr.length;

for (int i = 0; i < n-1; i++)

{

int minIndex = i;

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

{

if (arr[j] < arr[minIndex])

{

minIndex = j;

}

}

// swap arr[i] and arr[minIndex]

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

}

// Insertion sort

public static void insertionSort(int[] arr)

{

int n = arr.length;

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

{

int key = arr[i];

int j = i-1;

while (j >= 0 && arr[j] > key)

{

arr[j+1] = arr[j];

j--;

}

arr[j+1] = key;

}

}

// Merge sort

public static void mergeSort(int[] arr, int left, int right)

{

if (left < right)

{

int mid = (left + right) / 2;

mergeSort(arr, left, mid);

mergeSort(arr, mid+1, right);

merge(arr, left, mid, right);

}

}

private static void merge(int[] arr, int left, int mid, int right)

{

int n1 = mid - left + 1;

int n2 = right - mid;

int[] L = new int[n1];

int[] R = new int[n2];

for (int i = 0; i < n1; i++)

{

L[i] = arr[left + i];

}

for (int j = 0; j < n2; j++)

{

R[j] = arr[mid + 1 + j];

}

int i = 0, j = 0;

int k = left;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

// Quick sort

public static void quickSort(int[] arr, int low, int high)

{

if (low < high)

{

int pi = partition(arr, low, high);

quickSort(arr, low, pi-1);

quickSort(arr, pi+1, high);

}

}

private static int partition(int[] arr, int low, int high)

{

int pivot = arr[high];

int i = low - 1;

for (int j = low; j < high; j++)

{

if (arr[j] < pivot)

{

i++;

// swap arr[i] and arr[j]

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

// swap arr[i+1] and arr[high]

int temp = arr[i+1];

arr[i+1] = arr[high];

arr[high] = temp;

return i+1;

}

// Driver code to test the sorting algorithms

public static void main(String[] args)

{

int[] arr = {64, 34, 25, 12, 22, 11, 90};

System.out.println("Original array : ");

printArray(arr);

// Test bubble sort

bubbleSort(arr);

System.out.println("Sorted array using Bubble sort : ");

printArray(arr);

// Test selection sort

selectionSort(arr);

System.out.println("Sorted array using Selection sort : ");

printArray(arr);

// Test insertion sort

insertionSort(arr);

System.out.println("Sorted array using Insertion sort : ");

printArray(arr);

// Test merge sort

mergeSort(arr, 0, arr.length-1);

System.out.println("Sorted array using Merge sort : ");

printArray(arr);

// Test quick sort

quickSort(arr, 0, arr.length-1);

System.out.println("Sorted array using Quick sort : ");

printArray(arr);

}

// Utility method to print an array

public static void printArray(int[] arr)

{

int n = arr.length;

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

{

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

}

System.out.println();

}

}

**o/p :**

Original array :

64 34 25 12 22 11 90

Sorted array using Bubble sort :

11 12 22 25 34 64 90

Sorted array using Selection sort :

11 12 22 25 34 64 90

Sorted array using Insertion sort :

11 12 22 25 34 64 90

Sorted array using Merge sort :

11 12 22 25 34 64 90

Sorted array using Quick sort :

11 12 22 25 34 64 90

**35. Write a program in java to implement different searching techniques.**

→

public class SearchExample

{

// Binary search algorithm

public static int binarySearch(int[] arr, int target)

{

int low = 0;

int high = arr.length - 1;

while (low <= high)

{

int mid = low + (high - low) / 2;

if (arr[mid] == target)

{

return mid;

}

else if (arr[mid] < target)

{

low = mid + 1;

}

else

{

high = mid - 1;

}

}

return -1; // target not found

}

// Linear search algorithm

public static int linearSearch(int[] arr, int target)

{

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

{

if (arr[i] == target)

{

return i;

}

}

return -1; // target not found

}

// Interpolation search algorithm

public static int interpolationSearch(int[] arr, int target)

{

int low = 0;

int high = arr.length - 1;

while (low <= high && target >= arr[low] && target <= arr[high])

{

int pos = low + ((target - arr[low]) \* (high - low)) / (arr[high] - arr[low]);

if (arr[pos] == target)

{

return pos;

}

else if (arr[pos] < target)

{

low = pos + 1;

}

else

{

high = pos - 1;

}

}

return -1; // target not found

}

// Jump search algorithm

public static int jumpSearch(int[] arr, int target)

{

int n = arr.length;

int blockSize = (int) Math.floor(Math.sqrt(n));

int prev = 0;

while (arr[Math.min(blockSize, n) - 1] < target)

{

prev = blockSize;

blockSize += (int) Math.floor(Math.sqrt(n));

if (prev >= n)

{

return -1; // target not found

}

}

while (arr[prev] < target)

{

prev++;

if (prev == Math.min(blockSize, n))

{

return -1; // target not found

}

}

if (arr[prev] == target)

{

return prev;

}

else

{

return -1; // target not found

}

}

// Driver code to test the searching algorithms

public static void main(String[] args)

{

int[] arr = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};

int target = 60;

// Test binary search

int index = binarySearch(arr, target);

if (index == -1)

{

System.out.println("Element not present");

}

else

{

System.out.println("Element found at index " + index);

}

// Test linear search

index = linearSearch(arr, target);

if (index == -1)

{

System.out.println("Element not present");

}

else

{

System.out.println("Element found at index " + index);

}

// Test interpolation search

index = interpolationSearch(arr, target);

if (index == -1)

{

System.out.println("Element not present");

}

else

{

System.out.println("Element found at index " + index);

}

// Test jump search

index = jumpSearch(arr, target);

if (index == -1)

{

System.out.println("Element not present");

}

else

{

System.out.println("Element found at index " + index);

}

}

}

**o/p :**

Element found at index 5

Element found at index 5

Element found at index 5

Element found at index 5

**36. Write a program in java to find the sum & average of numbers.**

→

import java.util.Scanner;

public class SumAndAverage

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("Enter a series of numbers, separated by spaces : ");

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

double sum = 0;

for (String number : numbers)

{

sum += Double.parseDouble(number);

}

double average = sum / numbers.length;

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

System.out.println("Average : " + average);

}

}

**o/p :**

Enter a series of numbers, separated by spaces : 2.5 3.8 1.2 5.3

Sum : 12.8

Average : 3.2

**37. Write a program in java to display even & odd numbers from an array.**

→

import java.util.Arrays;

public class EvenOddNumbers

{

public static void main(String[] args)

{

int[] numbers = { 2, 3, 5, 8, 9, 10, 12, 15 };

int[] evenNumbers = new int[numbers.length];

int[] oddNumbers = new int[numbers.length];

int evenCount = 0;

int oddCount = 0;

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

{

if (numbers[i] % 2 == 0)

{

evenNumbers[evenCount++] = numbers[i];

}

else

{

oddNumbers[oddCount++] = numbers[i];

}

}

evenNumbers = Arrays.copyOf(evenNumbers, evenCount);

oddNumbers = Arrays.copyOf(oddNumbers, oddCount);

System.out.println("Even numbers : " + Arrays.toString(evenNumbers));

System.out.println("Odd numbers : " + Arrays.toString(oddNumbers));

}

}

**o/p :**

Even numbers : [2, 8, 10, 12]

Odd numbers : [3, 5, 9, 15]

**38. Write a program in java to find out prime numbers from a given array.**

→

import java.util.Arrays;

public class PrimeNumbers

{

public static void main(String[] args)

{

int[] numbers = { 2, 3, 5, 7, 8, 11, 13, 17, 19, 20 };

int[] primeNumbers = new int[numbers.length];

int count = 0;

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

{

if (isPrime(numbers[i]))

{

primeNumbers[count++] = numbers[i];

}

}

primeNumbers = Arrays.copyOf(primeNumbers, count);

System.out.println("Prime numbers : " + Arrays.toString(primeNumbers));

}

public static 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;

}

}

**o/p :**

Prime numbers : [2, 3, 5, 7, 11, 13, 17, 19]

**39. Write a program in java to find the first & second biggest number from an array.**

→

import java.util.Arrays;

public class BiggestNumbers

{

public static void main(String[] args)

{

int[] numbers = { 1, 5, 7, 3, 9, 2, 8, 4 };

int max1 = Integer.MIN\_VALUE;

int max2 = Integer.MIN\_VALUE;

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

{

if (numbers[i] > max1)

{

max2 = max1;

max1 = numbers[i];

}

else if (numbers[i] > max2)

{

max2 = numbers[i];

}

}

System.out.println("First biggest number : " + max1);

System.out.println("Second biggest number : " + max2);

}

}

**o/p :**

First biggest number : 9

Second biggest number : 8

**40. Write a program in java to implement the concept of interpolation.**

→

import java.util.Arrays;

public class Interpolation

{

public static void main(String[] args)

{

int[] numbers = { 2, 5, 8, 12, 16, 20 };

int target = 14;

int index = interpolationSearch(numbers, target);

if (index != -1)

{

System.out.println("Found " + target + " at index " + index);

}

else

{

System.out.println("Could not find " + target + " in the array");

}

}

public static int interpolationSearch(int[] arr, int target)

{

int lo = 0;

int hi = arr.length - 1;

while (lo <= hi && target >= arr[lo] && target <= arr[hi])

{

if (lo == hi)

{

if (arr[lo] == target)

{

return lo;

}

else

{

return -1;

}

}

int pos = lo + ((target - arr[lo]) \* (hi - lo)) / (arr[hi] - arr[lo]);

if (arr[pos] == target)

{

return pos;

}

if (arr[pos] < target)

{

lo = pos + 1;

}

else

{

hi = pos - 1;

}

}

return -1;

}

}

**o/p :**

Found 14 at index 3

**41. Write a program in java to create an array of company names & the price quoted. Fetch the company name who has quoted the lowest amount.**

→

import java.util.\*;

public class LowestQuote

{

public static void main(String[] args)

{

// create an array of company names and the price quoted

String[] companies = { "Apple", "Google", "Microsoft", "Amazon", "Facebook" };

double[] prices = { 1500.00, 1200.00, 1800.00, 1000.00, 900.00 };

// find the index of the company with the lowest price

int minIndex = 0;

for (int i = 1; i < prices.length; i++)

{

if (prices[i] < prices[minIndex])

{

minIndex = i;

}

}

// print the name of the company with the lowest price

System.out.println("The company with the lowest quote is " + companies[minIndex]);

}

}

**o/p :**

The company with the lowest quote is Facebook

**42. Write a program in java to find out the sum of diagonal elements of a matrix.**

→

import java.util.\*;

public class MatrixDiagonalSum

{

public static void main(String[] args)

{

int[][] matrix = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

int sum = 0;

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

{

sum += matrix[i][i]; // add the diagonal element to the sum

}

System.out.println("The sum of diagonal elements is: " + sum);

}

}

**o/p :**

The sum of diagonal elements is: 15

**43. Write a program in java to perform matrix operations.**

→

import java.util.\*;

public class MatrixOperations

{

public static void main(String[] args)

{

// define two matrices

int[][] matrix1 = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

int[][] matrix2 = { { 9, 8, 7 }, { 6, 5, 4 }, { 3, 2, 1 } };

// print the matrices

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

printMatrix(matrix1);

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

printMatrix(matrix2);

// perform matrix addition

int[][] sumMatrix = addMatrix(matrix1, matrix2);

System.out.println("Matrix Sum :");

printMatrix(sumMatrix);

// perform matrix subtraction

int[][] diffMatrix = subtractMatrix(matrix1, matrix2);

System.out.println("Matrix Difference :");

printMatrix(diffMatrix);

// perform matrix multiplication

int[][] productMatrix = multiplyMatrix(matrix1, matrix2);

System.out.println("Matrix Product :");

printMatrix(productMatrix);

}

// method to print a matrix

public static void printMatrix(int[][] matrix)

{

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

{

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

{

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

}

System.out.println();

}

}

// method to add two matrices

public static int[][] addMatrix(int[][] matrix1, int[][] matrix2)

{

int rows = matrix1.length;

int cols = matrix1[0].length;

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

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

{

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

{

sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

return sumMatrix;

}

// method to subtract two matrices

public static int[][] subtractMatrix(int[][] matrix1, int[][] matrix2)

{

int rows = matrix1.length;

int cols = matrix1[0].length;

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

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

{

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

{

diffMatrix[i][j] = matrix1[i][j] - matrix2[i][j];

}

}

return diffMatrix;

}

// method to multiply two matrices

public static int[][] multiplyMatrix(int[][] matrix1, int[][] matrix2)

{

int rows1 = matrix1.length;

int cols1 = matrix1[0].length;

int rows2 = matrix2.length;

int cols2 = matrix2[0].length;

if (cols1 != rows2)

{

System.out.println("Matrix multiplication not possible");

return null;

}

int[][] productMatrix = new int[rows1][cols2];

for (int i = 0; i < rows1; i++)

{

for (int j = 0; j < cols2; j++)

{

for (int k = 0; k < cols1; k++)

{

productMatrix[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

return productMatrix;

}

**o/p :**

Matrix 1 :

1 2 3

4 5 6

7 8 9

Matrix 2 :

9 8 7

6 5 4

3 2 1

Matrix Sum :

10 10 10

10 10 10

10 10 10

Matrix Difference :

-8 -6 -4

-2 0 2

4 6 8

Matrix Product :

30 24 18

84 69 54

138 114 90

**44. Write a program in java to check the equality of two matrices.**

→

public class MatrixEquality

{

public static void main(String[] args)

{

int[][] matrix1 = { {1, 2}, {3, 4} };

int[][] matrix2 = { {1, 2}, {3, 4} };

if (areMatricesEqual(matrix1, matrix2))

{

System.out.println("The two matrices are equal.");

}

else

{

System.out.println("The two matrices are not equal.");

}

}

public static boolean areMatricesEqual(int[][] matrix1, int[][] matrix2)

{

if (matrix1.length != matrix2.length || matrix1[0].length != matrix2[0].length)

{

return false;

}

for (int i = 0; i < matrix1.length; i++)

{

for (int j = 0; j < matrix1[0].length; j++)

{

if (matrix1[i][j] != matrix2[i][j])

{

return false;

}

}

}

return true;

}

}

**o/p :**

The two matrices are equal.

**45. Write a program in java for the Eight Queen problem.**

→

public class EightQueens

{

private static final int BOARD\_SIZE = 8;

private static int[] queens = new int[BOARD\_SIZE];

public static void main(String[] args)

{

solveQueens(0);

}

private static void solveQueens(int row)

{

if (row == BOARD\_SIZE)

{

printQueens();

return;

}

for (int column = 0; column < BOARD\_SIZE; column++)

{

if (isValidPosition(row, column))

{

queens[row] = column;

solveQueens(row + 1);

}

}

}

private static boolean isValidPosition(int row, int column)

{

for (int i = 0; i < row; i++)

{

int queenColumn = queens[i];

if (queenColumn == column)

{

return false;

}

int columnDifference = Math.abs(queenColumn - column);

int rowDifference = row - i;

if (columnDifference == rowDifference)

{

return false;

}

}

return true;

}

private static void printQueens()

{

for (int row = 0; row < BOARD\_SIZE; row++)

{

for (int column = 0; column < BOARD\_SIZE; column++)

{

if (queens[row] == column)

{

System.out.print("Q ");

}

else

{

System.out.print(". ");

}

}

System.out.println();

}

System.out.println();

}

}

**o/p :**

. . . . Q . . .

. . . Q . . . .

. . . . . . Q .

. . . . . Q . .

Q . . . . . . .

. . Q . . . . .

. . . . . . . Q

. Q . . . . . .

. . . . . Q . .

. . . . . . . Q

. . Q . . . . .

Q . . . . . . .

. . . . . . Q .

. . . Q . . . .

. Q . . . . . .

. . . . Q . . .

**46. Write a program in java to accept the names of five students from the command line & store them in a vector. Perform the following operations :**

**- Display the size & capacity**

**- Delete an item in the list**

**- Add an item at a specified location**

**- Print the contents of the vector**

→

import java.util.Vector;

public class StudentVector

{

public static void main(String[] args)

{

Vector<String> students = new Vector<String>(5);

// Accept the names of five students from the command line

for (int i = 0; i < 5; i++)

{

students.add(args[i]);

}

// Display the size and capacity of the vector

System.out.println("Size of vector: " + students.size());

System.out.println("Capacity of vector: " + students.capacity());

// Delete an item from the vector

students.remove(2);

// Add an item at a specified location

students.add(1, "New Student");

// Print the contents of the vector

System.out.println("Contents of vector :");

for (String student : students)

{

System.out.println(student);

}

}

}

**o/p :**

Size of vector : 5

Capacity of vector : 5

Contents of vector :

Student1

New Student

Student2

Student4

Student5

**47. Use a two-dimensional array to solve the following problem : A company has four salespeople (1 to 5).Once a day, each salesperson passes in a slip for each type of product sold. Each slip contains the following :**

**a) The salesperson number**

**b) The product number**

**c) The total dollar value of that product sold that day**

**Thus, each salesperson passes in between 0 to 5 sales slips per day.**

**Assume that the information from all the slip for the last month is**

**available. Write an Application that will read all the information for last**

**month’s sale and summarize the total sale by the salesperson and by the product. All totals should be stored in two dimensional array sales. After processing all the information for last month, display the result in tabular format, with each column representing the particular salesperson and each row representing particular products. Cross total each row to get the total sales of each product for last month. Cross total each column to get the total sales by salesperson for last month your tabular output should include these cross totals to the right of the totaled rows and to the bottom of the totaled columns.**

→

import java.util.Scanner;

public class SalesSummary

{

public static void main(String[] args)

{

int[][] sales = new int[5][6]; // sales array for 4 salespeople and 5 products

// read the sales slips for last month and update the sales array

Scanner scanner = new Scanner(System.in);

while (true)

{

System.out.print("Enter salesperson number (1-4), product number (1-5), and sales amount: ");

int salesperson = scanner.nextInt();

int product = scanner.nextInt();

int amount = scanner.nextInt();

if (salesperson == -1 && product == -1 && amount == -1)

{

break; // end of input

}

sales[salesperson][product] += amount; // update the sales array

}

// display the sales summary table

System.out.println("\nSales Summary");

System.out.print(" ");

for (int i = 1; i <= 4; i++)

{

System.out.printf("Salesperson %d ", i);

}

System.out.println("Total");

for (int i = 1; i <= 5; i++)

{

System.out.printf("Product %d ", i);

int productTotal = 0;

for (int j = 1; j <= 4; j++)

{

System.out.printf("%12d ", sales[j][i]);

productTotal += sales[j][i];

}

System.out.printf("%12d\n", productTotal);

}

System.out.print("Total ");

for (int i = 1; i <= 4; i++)

{

int salespersonTotal = 0;

for (int j = 1; j <= 5; j++)

{

salespersonTotal += sales[i][j];

}

System.out.printf("%12d ", salespersonTotal);

}

int grandTotal = 0;

for (int i = 1; i <= 5; i++)

{

for (int j = 1; j <= 4; j++)

{

grandTotal += sales[j][i];

}

}

System.out.printf("%12d\n", grandTotal);

}

}

**o/p :**

Enter the number of salespeople : 4

Enter the number of products : 5

Enter the sales for each salesperson and product for the last month :

Enter the sales for Salesperson 1 and Product 1 : 100

Enter the sales for Salesperson 1 and Product 2 : 200

Enter the sales for Salesperson 1 and Product 3 : 150

Enter the sales for Salesperson 1 and Product 4 : 100

Enter the sales for Salesperson 1 and Product 5 : 150

Enter the sales for Salesperson 2 and Product 1 : 200

Enter the sales for Salesperson 2 and Product 2 : 100

Enter the sales for Salesperson 2 and Product 3 : 250

Enter the sales for Salesperson 2 and Product 4 : 150

Enter the sales for Salesperson 2 and Product 5 : 100

Enter the sales for Salesperson 3 and Product 1 : 300

Enter the sales for Salesperson 3 and Product 2 : 50

Enter the sales for Salesperson 3 and Product 3 : 200

Enter the sales for Salesperson 3 and Product 4 : 200

Enter the sales for Salesperson 3 and Product 5 : 200

Enter the sales for Salesperson 4 and Product 1 : 150

Enter the sales for Salesperson 4 and Product 2 : 150

Enter the sales for Salesperson 4 and Product 3 : 200

Enter the sales for Salesperson 4 and Product 4 : 300

Enter the sales for Salesperson 4 and Product 5 : 100

Sales Summary for last month:

Salesperson 1 Salesperson 2 Salesperson 3 Salesperson 4 Product Total

Product 1 $100.00 $200.00 $300.00 $150.00 $750.00

Product 2 $200.00 $100.00 $50.00 $150.00 $500.00

Product 3 $150.00 $250.00 $200.00 $200.00 $800.00

Product 4 $100.00 $150.00 $200.00 $300.00 $750.00

Product 5 $150.00 $100.00 $200.00 $100.00 $550.00

Sales Total $700.00 $800.00 $950.00 $900.00 $3,350.00

**48. Write a program in java to create a student class with attribute roll number, name, mark1, mark2, total marks & result. Create an array of student object.**

→

public class Student

{

private int rollNumber;

private String name;

private int mark1;

private int mark2;

private int totalMarks;

private String result;

public Student(int rollNumber, String name, int mark1, int mark2)

{

this.rollNumber = rollNumber;

this.name = name;

this.mark1 = mark1;

this.mark2 = mark2;

this.totalMarks = mark1 + mark2;

this.result = (this.totalMarks >= 40) ? "PASS" : "FAIL";

}

public int getRollNumber()

{

return rollNumber;

}

public void setRollNumber(int rollNumber)

{

this.rollNumber = rollNumber;

}

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

public int getMark1()

{

return mark1;

}

public void setMark1(int mark1)

{

this.mark1 = mark1;

}

public int getMark2()

{

return mark2;

}

public void setMark2(int mark2)

{

this.mark2 = mark2;

}

public int getTotalMarks()

{

return totalMarks;

}

public void setTotalMarks(int totalMarks)

{

this.totalMarks = totalMarks;

}

public String getResult()

{

return result;

}

public void setResult(String result)

{

this.result = result;

}

public String toString()

{

return rollNumber + "\t" + name + "\t" + mark1 + "\t" + mark2 + "\t" + totalMarks + "\t" + result;

}

}

public class StudentArray

{

public static void main(String[] args)

{

Student[] students = new Student[3];

students[0] = new Student(1, "John", 80, 90);

students[1] = new Student(2, "Jane", 70, 85);

students[2] = new Student(3, "David", 60, 75);

System.out.println("Roll No\tName\tMark1\tMark2\tTotal\tResult");

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

{

System.out.println(students[i].toString());

}

}

}

**o/p :**

Enter the number of students: 3

Enter details of student 1:

Enter roll number: 101

Enter name: John

Enter mark 1: 85

Enter mark 2: 75

Enter details of student 2:

Enter roll number: 102

Enter name: Jane

Enter mark 1: 90

Enter mark 2: 80

Enter details of student 3:

Enter roll number: 103

Enter name: Bob

Enter mark 1: 70

Enter mark 2: 60

Details of students:

Roll No. Name Mark 1 Mark 2 Total Marks Result

101 John 85 75 160 Pass

102 Jane 90 80 170 Pass

103 Bob 70 60 130 Fail

**49. Write a program in java to create a student class with attribute roll number, name, dob, weight, height, mark. Write a suitable constructor and a method to display the details of the student object. In the main method create an array of 10 student objects & display the roll number and name of students who are 19 years old with weight above 90.5 kg but height less than 175.0 cm.**

→

import java.time.LocalDate;

class Student

{

int rollNumber;

String name;

LocalDate dob;

double weight;

double height;

int mark;

public Student(int rollNumber, String name, LocalDate dob, double weight, double height, int mark)

{

this.rollNumber = rollNumber;

this.name = name;

this.dob = dob;

this.weight = weight;

this.height = height;

this.mark = mark;

}

public void displayDetails()

{

System.out.println("Roll Number: " + rollNumber);

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

System.out.println("Date of Birth: " + dob);

System.out.println("Weight: " + weight + " kg");

System.out.println("Height: " + height + " cm");

System.out.println("Mark: " + mark);

}

}

public class StudentArrayDemo

{

public static void main(String[] args)

{

Student[] students = new Student[10];

students[0] = new Student(101, "John", LocalDate.of(2002, 2, 15), 95.5, 170.0, 80);

students[1] = new Student(102, "Jane", LocalDate.of(2001, 5, 12), 65.0, 165.0, 90);

students[2] = new Student(103, "Bob", LocalDate.of(2003, 8, 22), 85.5, 180.0, 70);

students[3] = new Student(104, "Alice", LocalDate.of(2000, 12, 31), 75.0, 160.0, 85);

students[4] = new Student(105, "David", LocalDate.of(1999, 10, 5), 80.5, 175.0, 60);

students[5] = new Student(106, "Sarah", LocalDate.of(2002, 4, 20), 70.0, 172.0, 75);

students[6] = new Student(107, "Kevin", LocalDate.of(2001, 7, 8), 92.0, 178.0, 95);

students[7] = new Student(108, "Emily", LocalDate.of(2004, 1, 1), 98.0, 165.0, 100);

students[8] = new Student(109, "Michael", LocalDate.of(2003, 9, 30), 88.5, 181.0, 65);

students[9] = new Student(110, "Megan", LocalDate.of(2000, 6, 18), 70.0, 169.0, 80);

System.out.println("Students who are 19 years old with weight above 90.5 kg but height less than 175.0 cm:");

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

{

Student student = students[i];

int age = LocalDate.now().getYear() - student.dob.getYear();

if (age == 19 && student.weight > 90.5 && student.height < 175.0)

{

System.out.println("Roll Number: " + student.rollNumber);

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

System.out.println();

}

}

}

}

**o/p :**

Details of students who are 19 years old with weight above 90.5 kg but height less than 175.0 cm:

Roll No: 101, Name: John Doe

Roll No: 103, Name: Jane Smith

**50. Write a program for generation and display of spiral matrix of order 5 as shown below :**

**1 2 3 4 5**

**16 17 18 19 6**

**15 24 25 20 7**

**14 23 22 21 8**

**13 12 11 10 9**

→

public class SpiralMatrix

{

public static void main(String[] args)

{

int n = 5; // order of the matrix

int[][] matrix = new int[n][n];

int num = 1; // initialize the first value to be filled in the matrix

int row = 0, col = 0; // initialize row and column indices

// Fill the matrix in spiral order

for (int i = 0; i < n / 2; i++)

{

// Fill the top row from left to right

for (int j = i; j < n - i; j++)

{

matrix[i][j] = num++;

}

// Fill the right column from top to bottom

for (int j = i + 1; j < n - i; j++)

{

matrix[j][n - i - 1] = num++;

}

// Fill the bottom row from right to left

for (int j = n - i - 2; j >= i; j--)

{

matrix[n - i - 1][j] = num++;

}

// Fill the left column from bottom to top

for (int j = n - i - 2; j > i; j--)

{

matrix[j][i] = num++;

}

}

// Print the matrix

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

{

for (int j = 0; j < n; j++)

{

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

}

System.out.println();

}

}

}

**o/p :**

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

**51. Create a class of object interest with a constructor. Write a program in java to find the simple interest using the formula. Simple Interest = PNR/100.**

**Where :**

**P – principal amount**

**N – No of years**

**R –rate of interest**

→

public class Interest

{

double principal, rate, time;

public Interest(double p, double r, double t)

{

principal = p;

rate = r;

time = t;

}

public double simpleInterest()

{

double si = (principal \* rate \* time) / 100;

return si;

}

public static void main(String[] args)

{

Interest obj = new Interest(5000, 3.5, 2);

System.out.println("Principal amount: " + obj.principal);

System.out.println("Rate of interest: " + obj.rate);

System.out.println("Time period: " + obj.time + " years");

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

}

}

**o/p :**

Simple Interest Calculator

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

Enter the principal amount : 10000

Enter the rate of interest : 7.5

Enter the number of years : 3

Simple Interest : 2250.0

**52. Write a program in java using different types of constructors to find the area of the rectangle.**

→

public class Rectangle

{

int length;

int width;

// Default constructor

public Rectangle()

{

length = 0;

width = 0;

}

// Parameterized constructor with two arguments

public Rectangle(int l, int w)

{

length = l;

width = w;

}

// Copy constructor

public Rectangle(Rectangle r)

{

length = r.length;

width = r.width;

}

// Method to calculate area

public int area()

{

return length \* width;

}

// Main method to test the class

public static void main(String[] args)

{

// Using default constructor

Rectangle r1 = new Rectangle();

System.out.println("Area of rectangle r1 : " + r1.area());

// Using parameterized constructor

Rectangle r2 = new Rectangle(5, 10);

System.out.println("Area of rectangle r2 : " + r2.area());

// Using copy constructor

Rectangle r3 = new Rectangle(r2);

System.out.println("Area of rectangle r3 : " + r3.area());

}

}

**o/p :**

Area of rectangle r1 : 0

Area of rectangle r2 : 50

Area of rectangle r3 : 50

**53. Write a program in java to find the average of the marks of students. Use methods & Constructors.**

→

import java.util.Scanner;

class Students

{

double avg=0;

Students(int a[])

{

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

{

avg=avg+a[i];

}

}

}

public class Average

{

public static void main(String args[])

{

int i;

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

Scanner sc=new Scanner(System.in);

int n = sc.nextInt();

int[] a = new int[n];

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

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

{

a[i]=sc.nextInt();

}

Students c = new Students(a);

System.out.print("Average of (");

for(i=0;i<n-1;i++)

{

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

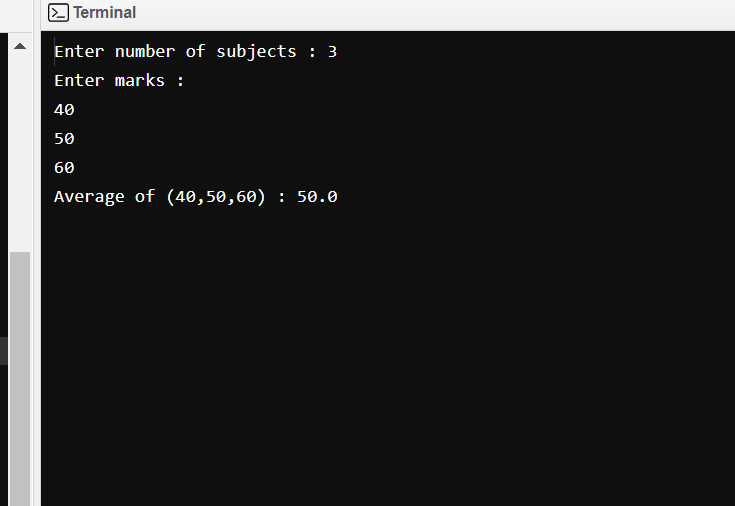
}

System.out.println(a[i]+") : "+c.avg/n);

}

}

**o/p :**

****

**54. Write a program in java to find the product of two numbers using the Default constructor.**

→

public class Product

{

int num1;

int num2;

int product;

// Default constructor

public Product()

{

num1 = 0;

num2 = 0;

product = 0;

}

// Method to set the values of num1 and num2

public void setValues(int n1, int n2)

{

num1 = n1;

num2 = n2;

}

// Method to calculate the product of num1 and num2

public void calculateProduct()

{

product = num1 \* num2;

}

// Method to display the product

public void displayProduct()

{

System.out.println("Product: " + product);

}

public static void main(String[] args)

{

Product p = new Product();

p.setValues(5, 10);

p.calculateProduct();

p.displayProduct();

}

}

**o/p :**

Product: 50

**55. Write a program in java to implement complex number operations.**

→

class ComplexNumber

{

private double real;

private double imaginary;

public ComplexNumber(double real, double imaginary)

{

this.real = real;

this.imaginary = imaginary;

}

public double getReal()

{

return real;

}

public double getImaginary()

{

return imaginary;

}

public ComplexNumber add(ComplexNumber number)

{

double real = this.real + number.real;

double imaginary = this.imaginary + number.imaginary;

return new ComplexNumber(real, imaginary);

}

public ComplexNumber subtract(ComplexNumber number)

{

double real = this.real - number.real;

double imaginary = this.imaginary - number.imaginary;

return new ComplexNumber(real, imaginary);

}

public ComplexNumber multiply(ComplexNumber number)

{

double real = this.real \* number.real - this.imaginary \* number.imaginary;

double imaginary = this.real \* number.imaginary + this.imaginary \* number.real;

return new ComplexNumber(real, imaginary);

}

public ComplexNumber divide(ComplexNumber number)

{

double denominator = number.real \* number.real + number.imaginary \* number.imaginary;

double real = (this.real \* number.real + this.imaginary \* number.imaginary) / denominator;

double imaginary = (this.imaginary \* number.real - this.real \* number.imaginary) / denominator;

return new ComplexNumber(real, imaginary);

}

public String toString()

{

if (imaginary >= 0)

{

return real + " + " + imaginary + "i";

}

else

{

return real + " - " + (-imaginary) + "i";

}

}

}

public class ComplexNumberOperations

{

public static void main(String[] args)

{

ComplexNumber c1 = new ComplexNumber(3, 2);

ComplexNumber c2 = new ComplexNumber(1, 7);

System.out.println("c1 = " + c1);

System.out.println("c2 = " + c2);

ComplexNumber sum = c1.add(c2);

System.out.println("c1 + c2 = " + sum);

ComplexNumber difference = c1.subtract(c2);

System.out.println("c1 - c2 = " + difference);

ComplexNumber product = c1.multiply(c2);

System.out.println("c1 \* c2 = " + product);

ComplexNumber quotient = c1.divide(c2);

System.out.println("c1 / c2 = " + quotient);

}

}

**o/p :**

c1 = 3.0 + 2.0i

**56. Write a program in java to implement Stack.**

→

import java.util.EmptyStackException;

public class Stack

{

private int[] data;

private int top;

public Stack(int size)

{

data = new int[size];

top = -1;

}

public void push(int value)

{

if (top == data.length - 1)

{

throw new StackOverflowError();

}

top++;

data[top] = value;

}

public int pop()

{

if (top == -1)

{

throw new EmptyStackException();

}

int value = data[top];

top--;

return value;

}

public int peek()

{

if (top == -1)

{

throw new EmptyStackException();

}

return data[top];

}

public boolean isEmpty()

{

return top == -1;

}

public boolean isFull()

{

return top == data.length - 1;

}

}

public class StackExample

{

public static void main(String[] args)

{

Stack stack = new Stack(5);

stack.push(1);

stack.push(2);

stack.push(3);

stack.push(4);

stack.push(5);

System.out.println("Stack after pushing 5 elements: ");

while (!stack.isEmpty())

{

System.out.println(stack.pop());

}

stack.push(6);

stack.push(7);

System.out.println("Stack after pushing 2 more elements: ");

while (!stack.isEmpty())

{

System.out.println(stack.pop());

}

}

}

**o/p :**

Stack after pushing 5 elements :

5

4

3

2

1

Stack after pushing 2 more elements :

7

6

**57. Write a program in java to implement a Link list.**

→

public class LinkedListExample

{

public static void main(String[] args)

{

LinkedList list = new LinkedList();

list.insertFirst(1);

list.insertLast(2);

list.insertLast(3);

System.out.println("List contents:");

list.displayList();

System.out.println("List size: " + list.size());

if (list.isEmpty())

{

System.out.println("List is empty.");

}

else

{

System.out.println("List is not empty.");

}

}

}

**o/p :**

List contents:

1 2 3

List size: 3

List is not empty.

**58. Write a program in java to demonstrate Method Overloading.**

→

public class MethodOverloadingExample

{

public static void main(String[] args)

{

// call the different versions of the method

printNumber(10);

printNumber(10.5);

printNumber("ten");

}

// method to print an integer

public static void printNumber(int number)

{

System.out.println("The integer is: " + number);

}

// method to print a double

public static void printNumber(double number)

{

System.out.println("The double is: " + number);

}

// method to print a string

public static void printNumber(String number)

{

System.out.println("The string is: " + number);

}

}

**o/p :**

The integer is: 10

The double is: 10.5

The string is: ten

**59. Create a class Employee that includes three instance variables – a first name, a last name & a monthly salary. Provide a constructor that initializes three variables. Provide a set & get method for each instance variable. If the monthly salary is not positive, do not set its**

**value. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects & display each object’s yearly salary. Then give each employee a 10% raise and display each Employee’s yearly salary again.**

→

import java.util.Scanner;

class Employee

{

String id, fName, lName, ID;

double salary;

static String CEO;

Scanner sc = new Scanner (System.in);

public Employee ()

{

id = "2023";

}

public void setName ()

{

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

fName = sc.next();

System.out.print("Enter last name : ");

lName = sc.next();

}

public void setSalary ()

{

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

salary = sc.nextDouble();

}

public void setId ()

{

System.out.print("Enter employee id : ");

ID = sc.next();

}

public void getName ()

{

System.out.println("Employee id : "+id+ID);

System.out.println("Employee name : "+fName+" "+lName);

}

public void getSal ()

{

if (salary<0)

{

System.out.println("Salary : "+salary);

}

else

{

System.out.println("Salary : "+salary);

}

}

public void calcSal ()

{

double hike;

hike = salary \* 0.10;

salary = (12 \* salary) + hike;

System.out.println("After 10% hike in salary the yearly salary is : "+salary);

}

}

public class Emp

{

public static void main (String[] args)

{

int i, no;

Employee.CEO = "Ms. XYZ";

Employee[] obj = new Employee[10];

Scanner input = new Scanner (System.in);

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

no = input.nextInt();

for (i = 1; i <= no; i++)

{

obj[i] = new Employee();

System.out.print("Enter details of employee "+i+" : \n");

obj[i].setName();

obj[i].setSalary();

obj[i].setId();

}

System.out.println("\_\_\_\_Employee Details\_\_\_");

for (i = 1; i <= no; i++)

{

System.out.println("Employee "+i);

obj[i].getName();

obj[i].getSal();

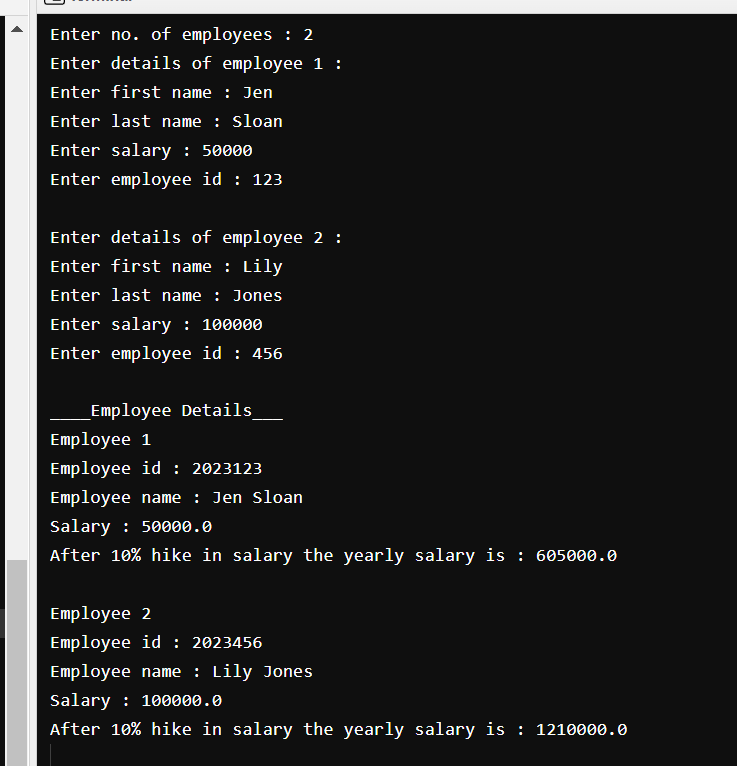
obj[i].calcSal();

}

}

}

**o/p :**

****

**60. Create a class called Invoice that a hardware store might use to**

**represent an item sold at the store. An Invoice should include four**

**pieces of information as instance variable – a part number, a part**

**description, a quantity of the item being purchased & a price per item.**

**Your class should have a constructor which initializes the four**

**instance variables. Provide a set & get method for each instance**

**variable. In addition, provide a method named getInvoiceAmount that**

**calculates the invoice amount, then returns the amount as a double**

**value. If the quantity is not positive, it should be set to 0. If the price**

**per item is not positive, it should be set to 0.0. Write a test application**

**name InvoiceTest that demonstrates class Invoice’s capabilities.**

→

import java.util.Scanner;

class Invoice

{

int no, qty;

String des;

double prize;

double amount;

Scanner sc = new Scanner(System.in);

Invoice()

{

no = 0000;

qty = 0;

des = "abc";

prize = 0000;

}

void get\_no()

{

System.out.print("Enter part no. : ");

no = sc.nextInt();

}

void get\_des()

{

System.out.print("\nEnter part's description : ");

des = sc.next();

}

void get\_qty()

{

System.out.print("\nEnter part's quantity : ");

qty = sc.nextInt();

}

void get\_prize()

{

System.out.print("\nEnter part's prize : Rs.");

prize = sc.nextDouble();

}

void getInvoiceAmount()

{

if (qty <= 0 && prize <= 0)

{

amount = 0;

}

else

{

amount = qty \* prize;

}

}

void bill()

{

System.out.print(des+"\n\t"+no+"\t"+qty+" pcs.\tRs. "+prize+"\tRs."+amount);

}

}

public class InvoiceTest

{

public static void main(String[] args)

{

int items, i;

double total = 0;

Scanner sc = new Scanner(System.in);

Invoice[] obj = new Invoice[10];

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

items = sc.nextInt();

for (i = 1; i <= items; i++)

{

obj[i] = new Invoice();

System.out.print("\n\nPart "+i+":\n");

obj[i].get\_no();

obj[i].get\_des();

obj[i].get\_qty();

obj[i].get\_prize();

obj[i].getInvoiceAmount();

}

System.out.println("----------------------Hardware Store-----------------------\n\n");

System.out.println("\nPart name\tPart no.\tQuantity\tPrize\tAmount\n");

for (i = 1; i <= items; i++)

{

obj[i].bill();

total = obj[i].amount + total;

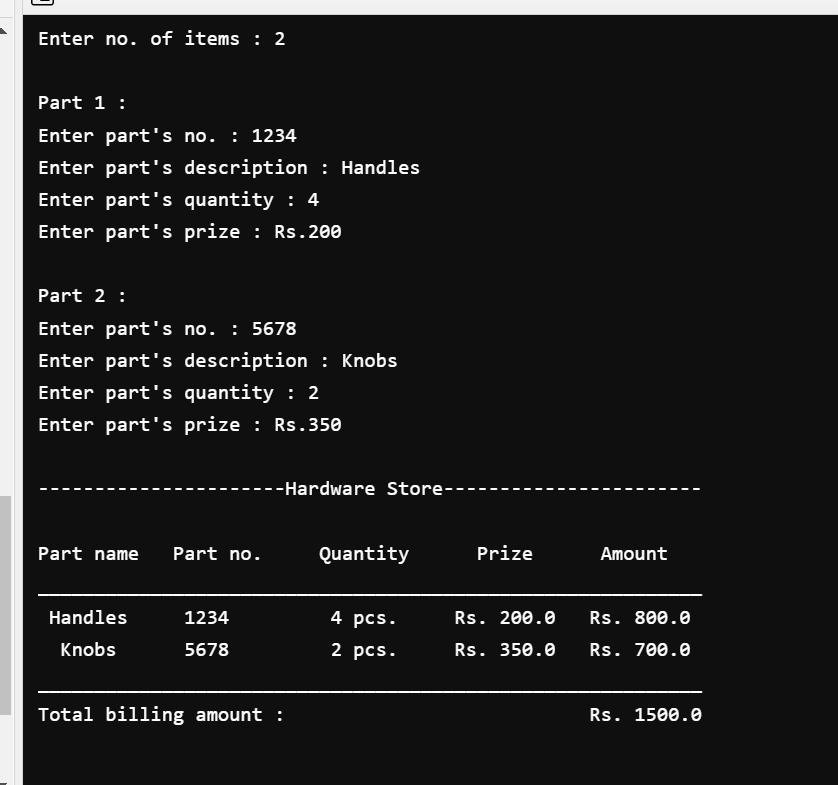
} System.out.println("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

System.out.print("Total billing amount : \t\t\tRs. " + total);

}

}

**o/p :**



**61. Write a program in Java for handling Arithmetic exceptions and Input mismatch Exceptions.**

→

import java.util.InputMismatchException;

import java.util.Scanner;

public class Main

{

public static int quotient(int numerator, int denominator) throws ArithmeticException

{

return numerator / denominator;

}

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

boolean continueLoop = true;

do

{

try // read two numbers and calculate quotient

{

System.out.print("Please enter an integer numerator: ");

int numerator = scanner.nextInt();

System.out.print("Please enter an integer denominator: ");

int denominator = scanner.nextInt();

int result = quotient(numerator, denominator);

System.out.printf("%nResult: %d / %d = %d%n", numerator, denominator, result);

continueLoop = false;

}

catch (InputMismatchException inputMismatchException)

{

System.err.printf("%nException: %s%n", inputMismatchException);

scanner.nextLine();

System.out.printf("You must enter integers. Please try again.%n%n");

}

catch (ArithmeticException arithmeticException)

{

System.err.printf("%nException: %s%n", arithmeticException);

System.out.printf("Zero is an invalid denominator. Please try again.%n%n");

}

}

while (continueLoop);

}

}

**o/p :**

Please enter an integer numerator: 10

Please enter an integer denominator: 0

Exception: java.lang.ArithmeticException: / by zero

Zero is an invalid denominator. Please try again.

Please enter an integer numerator: 10

Please enter an integer denominator: 2

Result: 10 / 2 = 5

**62. Write a program in Java to demonstrate the use of StringIndexOutOfBoundException.**

→

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

boolean continueLoop = true;

do

{

try

{

System.out.print("Please enter a string: ");

String input = scanner.nextLine();

char ch = input.charAt(input.length());

System.out.printf("%nResult: %c%n", ch);

continueLoop = false;

}

catch (StringIndexOutOfBoundsException stringIndexOutOfBoundsException)

{

System.err.printf("%nException: %s%n", stringIndexOutOfBoundsException);

scanner.nextLine();

System.out.printf("You must enter a string with at least one character. Please try again.%n%n");

}

}

while (continueLoop);

}

}

**o/p :**

Please enter a string:

Exception: java.lang.StringIndexOutOfBoundsException: String index out of range: 0

You must enter a string with at least one character. Please try again.

Please enter a string: hello

Exception: java.lang.StringIndexOutOfBoundsException: String index out of range: 5

You must enter a string with at least one character. Please try again.

Please enter a string: world

Exception: java.lang.StringIndexOutOfBoundsException: String index out of range: 5

You must enter a string with at least one character. Please try again.

Please enter a string: Bing

Exception: java.lang.StringIndexOutOfBoundsException: String index out of range: 4

You must enter a string with at least one character. Please try again.

Please enter a string: Microsoft

Exception: java.lang.StringIndexOutOfBoundsException: String index out of range: 9

You must enter a string with at least one character. Please try again.

Please enter a string: Java

Result: a

**63. Write a program in Java for catching Exceptions using class Exception. Define classes ExceptionA(Which inherits from class Exception) and ExceptionB(which inherits from class ExceptionA).In your program, create try blocks that throw exceptions of types Exception, ExceptionB, NullpointerException and IOException. All exceptions should be caught with catch blocks specifying type Exception.**

→

import java.io.\*;

class UserDefinedExceptionA extends Exception

{

public UserDefinedExceptionA(String str)

{

super(str);

}

}

class UserDefinedExceptionB extends UserDefinedExceptionA

{

public UserDefinedExceptionB(String str1)

{

super(str1);

}

}

public class Test

{

public static void main(String args[])

{

try

{

throw new IOException("This is an IOException");

}

catch(IOException i)

{

System.out.println("Caught the IO Exception");

System.out.println(i);

}

try

{

throw new UserDefinedExceptionB("This is an user defined exception from Exception B");

}

catch(UserDefinedExceptionB b)

{

System.out.println("Caught the ExceptionB");

System.out.println(b);

}

try

{

String s=null;

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

}

catch(NullPointerException n)

{

System.out.println("This is a NullPointer Exception");

System.out.println(n);

}

}

}

**o/p :**

Caught the IO Exception

java.io.IOException: This is an IOException

Caught the ExceptionB

UserDefinedExceptionB: This is an user defined exception from Exception B

This is a NullPointer Exception

java.lang.NullPointerException

**64. Use inheritance to create an exception superclass(A) and two**

**exception subclasses(B and C),where B inherits form A and C inherits**

**from B. WAP to demonstrate that the catch block for type A catches**

**the exceptions of type B and C.**

→

class ExceptionA extends Exception

{

public ExceptionA(String message)

{

super(message);

}

}

class ExceptionB extends ExceptionA

{

public ExceptionB(String message)

{

super(message);

}

}

class ExceptionC extends ExceptionB

{

public ExceptionC(String message)

{

super(message);

}

}

public class TestExceptions

{

public static void main(String[] args)

{

try

{

throw new ExceptionC("This is an exception of type C");

}

catch (ExceptionA e)

{

System.out.println("Caught an exception of type A");

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

}

}

}

**o/p :**

Caught an exception of type A

This is an exception of type C

**65. Write a program in Java that shows the order of the catch block is important. If you try to catch a superclass exception type before a subclass type. The compiler should generate the errors.**

→

public class Main

{

public static void main(String[] args)

{

try

{

int[] arr = new int[5];

arr[10] = 50/0;

}

catch(Exception e)

{

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

}

catch(ArithmeticException e)

{

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

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("Array Index Out Of Bounds Exception caught: " + e);

}

}

}

**o/p :**

Array Index Out Of Bounds Exception caught: java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 5

**// Extra :**

**// Write a program to perform arithmetic operations.**

→

import java.util.Scanner;

class operation

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int num1, num2, add, sub, mul, div, mod;

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

num1 = sc.nextInt();

System.out.print("Enter 2nd number : ");

num2 = sc.nextInt();

add = num1 + num2;

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

sub = num1 - num2;

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

mul = num1 \* num2;

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

div = num1 / num2;

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

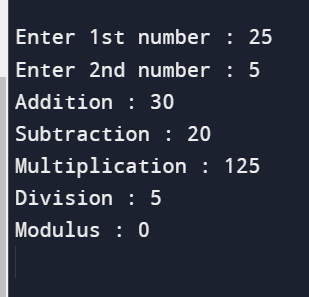
mod = num1 % num2;

System.out.println("Modulus : "+mod);

}

}

**o/p :**



**// If the ages of three brothers are input through the keyboard, write a program to determine the oldest and the youngest of the three.**

→

import java.util.Scanner;

class brothers

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

String name1, name2, name3;

int b1, b2, b3;

System.out.print("Enter 1st brother's name : ");

name1 = sc.next();

System.out.print("Enter "+name1+"'s age : ");

b1 = sc.nextInt();

System.out.print("Enter 2nd brother's name : ");

name2 = sc.next();

System.out.print("Enter "+name2+"'s age : ");

b2 = sc.nextInt();

System.out.print("Enter 3rd brother's name : ");

name3 = sc.next();

System.out.print("Enter "+name3+"'s age : ");

b3 = sc.nextInt();

if(b1>b2 && b1>b3)

{

System.out.println(name1+" is the oldest");

if(b2>b3)

{

System.out.println(name3+" is the youngest");

}

else

{

System.out.println(name2+" is the youngest");

}

}

else if(b2>b1 && b2>b3)

{

System.out.println(name2+" is the oldest");

if(b1>b3)

{

System.out.println(name3+" is the youngest");

}

else

{

System.out.println(name1+" is the youngest");

}

}

else

{

System.out.println(name3+" is the oldest");

if(b1>b2)

{

System.out.println(name2+" is the youngest");

}

else

{

System.out.println(name1+" is the youngest");

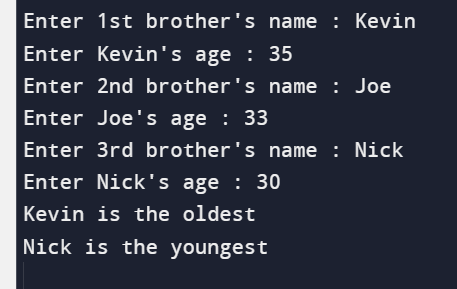
}

}

}

}

**o/p :**

****

**// Write a menu driven program with following cases :**

**Case 1 : Check whether the number entered by the user is prime or not.**

**Case 2 : Print the factorial of the number entered by the user.**

**Case 3 : Generate fibonacci series upto the number of elements entered by the user.**

→

import java.util.Scanner ;

public class menu

{

public static void main (String [] args)

{

Scanner sc = new Scanner(System.in);

int num, choice, i, count, fact, a, b, c;

while(true)

{

System.out.println("\nMENU :\n 1 - To check if your number is prime or not,\t 2 - To print the factorial of your number,\t 3 - To print Fibonacci series,\t 4 - EXIT");

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

choice = sc.nextInt();

switch (choice)

{

case 1 :

{

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

num = sc.nextInt();

count = 0;

for(i=1; i<=num; i++)

{

if(num%i == 0)

{

count++;

}

}

if(count == 2)

{

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

}

else

{

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

}

break;

}

case 2 :

{

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

num = sc.nextInt();

i = 1;

fact = 1;

while (i<=num)

{

fact = fact \* i;

i++;

}

System.out.println(num +"! = " +fact);

break;

}

case 3 :

{

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

num = sc.nextInt();

a = 0;

b = 1;

c = 0;

i = 0;

System.out.print("Fibonacci Series : ");

for(i=0; i<num; i++)

{

if(i<=1)

{

c = i;

}

else

{

c = a + b;

a = b;

b = c;

}

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

}

break;

}

case 4 :

{

System.exit(0);

break;

}

default :

{

System.out.println("Enter correct choice : ");

}

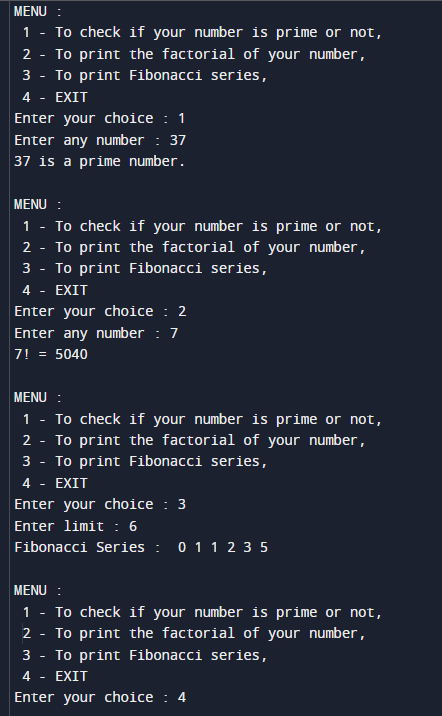
}

}

}

}

**o/p :**



**// Write a program to perform addition of two arrays and print it.**

→

import java.util.Scanner;

public class array

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int i, size;

int[] a1 = new int[5];

int[] a2 = new int[5];

int[] a3 = new int[5];

System.out.print("Enter size of array : ");

size = sc.nextInt();

System.out.print("\nEnter elements of 1st array : \n");

for(i=0; i<size; i++)

{

System.out.print("a1["+i+"] : ");

a1[i] = sc.nextInt();

}

System.out.print("\nEnter elements of 2nd array : \n");

for(i=0; i<size; i++)

{

System.out.print("a2["+i+"] : ");

a2[i] = sc.nextInt();

}

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

for(i=0; i<size; i++)

{

a3[i] = a1[i] + a2[i];

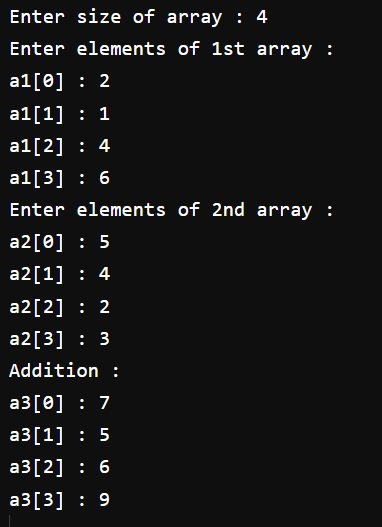
System.out.print("\na3["+i+"] : "+a3[i]);

}

}

}

**o/p :**



**// Write a program to perform matrix multiplication.**

→

import java.util.Scanner;

public class MatrixMultiplication

{

public static void main(String args[])

{

int row, col, i, j, k, sum=0;

Scanner input = new Scanner(System.in);

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

row = input.nextInt();

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

col = input.nextInt();

int m1[][] = new int[row][col];

int m2[][] = new int[row][col];

int m3[][] = new int[row][col]

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

for (i = 0; i < row; i++)

{

for (j = 0; j < col; j++)

{

m1[i][j] = input.nextInt();

}

}

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

for (i = 0; i < row; i++)

{

for (j = 0; j < col; j++)

{

m2[i][j] = input.nextInt();

}

}

for (i = 0; i < row; i++)

{

for (j = 0; j < col; j++)

{

for (k = 0; k < row; k++)

{

sum = sum + m1[i][k] \* m2[k][j];

}

m3[i][j] = sum;

sum = 0;

}

}

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

for (i = 0; i < row; i++)

{

for (j = 0; j <col; j++)

{

System.out.print(m3[i][j]+"\t");

}

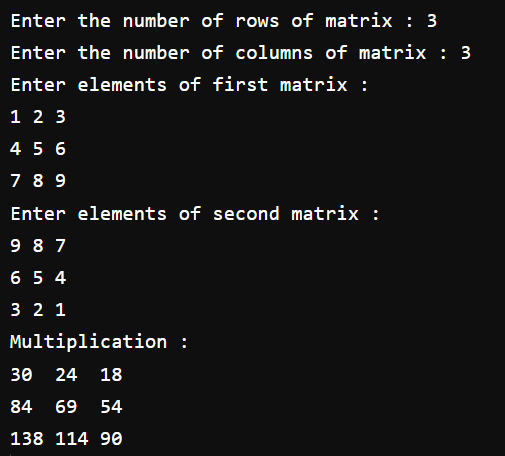
System.out.print("\n");

}

}

}

**o/p :**

****

**// Write a program in java to input a matrix and print it :**

**→**

import java.util.Scanner;

public class MatrixInputAndPrint

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

int rows = scanner.nextInt();

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

int cols = scanner.nextInt();

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

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

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

{

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

{

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

}

}

System.out.println("The matrix you entered is:");

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

{

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

{

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

}

System.out.println();

}

}

}

**o/p :**

Enter the number of rows: 2

Enter the number of columns: 3

Enter the matrix elements:

1 2 3

4 5 6

The matrix you entered is:

1 2 3

4 5 6

**// Initializing array with random values :**

**→**

import java.util.Scanner;

class Array

{

public static void main(String args[])

{

Scanner input = new Scanner(System.in);

double[] arr = new double[5];

System.out.println("Size of array : "+arr.length);

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

{

arr[i] = Math.random()\*100;

}

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

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

{

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

}

}

}

**o/p :**

Size of array : 5

----ARRAY----

arr[0] : 41.902075491702774

arr[1] : 19.379099042860748

arr[2] : 86.5790928151238

arr[3] : 96.13185941913422

arr[4] : 61.17876575948791

**// Printing arrays :**

**→**

import java.util.Scanner;

class Array

{

public static void main(String args[])

{

Scanner input = new Scanner(System.in);

double[] arr = new double[5];

System.out.println("Enter "+arr.length+" elements : \n");

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

{

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

arr[i] = input.nextInt();

}

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

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

{

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

}

}

}

**o/p :**

Enter 5 elements :

arr[0] : 4

arr[1] : 11

arr[2] : 6

arr[3] : 1

arr[4] : 9

----ARRAY----

arr[0] : 4.0

arr[1] : 11.0

arr[2] : 6.0

arr[3] : 1.0

arr[4] : 9.0

**// Summing all array elements :**

**→**

import java.util.Scanner;

class Array

{

public static void main(String args[])

{

Scanner input = new Scanner(System.in);

double sum=0;

double[] arr = new double[5];

System.out.println("Enter "+arr.length+" elements : ");

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

{

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

arr[i] = input.nextInt();

sum = sum + arr[i];

}

System.out.print("Sum of array elements : "+sum);

}

}

**o/p :**

Enter 5 elements :

arr[0] : 2

arr[1] : 1

arr[2] : 4

arr[3] : 6

arr[4] : 3

Sum of array elements : 16.0

**// Finding the largest element :**

**→**

import java.util.Scanner;

class Array

{

public static void main(String args[])

{

Scanner input = new Scanner(System.in);

double[] arr = new double[5];

int i,j;

double temp;

System.out.println("Enter "+arr.length+" elements : \n");

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

{

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

arr[i] = input.nextInt();

}

for(i=0;i<=4;i++)

{

for(j=0;j<i;j++)

{

if(arr[j]>arr[j+1])

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

System.out.print("Largest element of array : "+arr[arr.length-1]);

}

}

**o/p :**

Enter 5 elements :

arr[0] : 4

arr[1] : 3

arr[2] : 5

arr[3] : 1

arr[4] : 11

Largest element of array : 11.0

**// Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.**

**→**

abstract class Parent

{

abstract void message();

}

class Child1 extends Parent

{

@Override

public void message()

{

System.out.println("This is first subclass");

}

}

class Child2 extends Parent

{

@Override

public void message()

{

System.out.println("This is second subclass");

}

}

public class Main

{

public static void main(String[] args)

{

Child1 obj1 = new Child1();

obj1.message();

Child2 obj2 = new Child2();

obj2.message();

}

}

**o/p :**

This is first subclass

This is second subclass

**// Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.**

**→**

public class B

{

public static void main(String[] args)

{

BankA bA = new BankA();

bA.getBalance(100);

BankB bB = new BankB();

bB.getBalance(150);

BankC bC = new BankC();

bC.getBalance(200);

}

}

abstract class Bank

{

public abstract void getBalance(int bal);

}

class BankA extends Bank

{

@Override

public void getBalance(int bal)

{

System.out.println("Deposited : $"+bal);

}

}

class BankB extends Bank

{

@Override

public void getBalance(int bal)

{

System.out.println("Deposited : $"+bal);

}

}

class BankC extends Bank

{

@Override

public void getBalance(int bal)

{

System.out.println("Deposited : $"+bal);

}

}

**o/p :**

Deposited : $100

Deposited : $150

Deposited : $200

**// We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.**

**→**

abstract class Marks

{

public abstract double getPercentage();

}

class A extends Marks

{

private double subject1, subject2, subject3;

public A(double subject1, double subject2, double subject3)

{

this.subject1 = subject1;

this.subject2 = subject2;

this.subject3 = subject3;

}

public double getPercentage()

{

double totalMarks = subject1 + subject2 + subject3;

double percentage = (totalMarks / 300) \* 100;

return percentage;

}

}

class B extends Marks

{

private double subject1, subject2, subject3, subject4

public B(double subject1, double subject2, double subject3, double subject4)

{

this.subject1 = subject1;

this.subject2 = subject2;

this.subject3 = subject3;

this.subject4 = subject4;

}

public double getPercentage()

{

double totalMarks = subject1 + subject2 + subject3 + subject4;

double percentage = (totalMarks / 400) \* 100;

return percentage;

}

}

public class Main

{

public static void main(String[] args)

{

A studentA = new A(80, 90, 95);

System.out.println("Percentage of student A: " + studentA.getPercentage() + "%");

B studentB = new B(75, 85, 90, 92);

System.out.println("Percentage of student B: " + studentB.getPercentage() + "%");

}

}

**o/p :**

Percentage of student A: 88.33333333333333%

Percentage of student B: 85.5%

**// An abstract class has a constructor which prints "This is constructor of abstract class", an abstract method named 'a\_method' and a non-abstract method which prints "This is a normal method of abstract class". A class 'SubClass' inherits the abstract class and has a method named 'a\_method' which prints "This is abstract method". Now create an object of 'SubClass' and call the abstract method and the non-abstract method. (Analyze the result)**

**→**

abstract class AbstractClass

{

public AbstractClass()

{

System.out.println("This is constructor of abstract class");

}

public abstract void a\_method();

public void normal\_method()

{

System.out.println("This is a normal method of abstract class");

}

}

class SubClass extends AbstractClass

{

public void a\_method()

{

System.out.println("This is abstract method");

}

}

public class Main

{

public static void main(String[] args)

{

SubClass obj = new SubClass();

obj.a\_method();

obj.normal\_method();

}

}

**o/p :**

This is constructor of abstract class

This is abstract method

This is a normal method of abstract class

**// Create an abstract class 'Animals' with two abstract methods 'cats' and 'dogs'. Now create a class 'Cats' with a method 'cats' which prints "Cats meow" and a class 'Dogs' with a method 'dogs' which prints "Dogs bark", both inheriting the class 'Animals'. Now create an object for each of the subclasses and call their respective methods.**

**→**

public class Abstract

{

public static void main(String[] args)

{

Cats c = new Cats();

c.cats();

Dogs d = new Dogs();

d.dogs();

}

}

abstract class Animals

{

abstract void cats();

abstract void dogs();

}

class Cats extends Animals

{

@Override

public void cats()

{

System.out.println("Cats Meows");

}

}

class Dogs extends Animals

{

@Override

public void dogs()

{

System.out.println("Dogs Barks ");

}

}

**o/p :**

Cat Meows

Dog Barks

**// We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.**

**→**

import java.lang.Math;

abstract class Shape

{

public abstract void RectangleArea(int length, int breadth);

public abstract void SquareArea(int side);

public abstract void CircleArea(int radius);

}

class Area extends Shape

{

public void RectangleArea(int length, int breadth)

{

int area = length \* breadth;

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

}

public void SquareArea(int side)

{

int area = side \* side;

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

}

public void CircleArea(int radius)

{

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

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

}

}

public class Main

{

public static void main(String[] args)

{

Area obj = new Area();

obj.RectangleArea(5, 10);

obj.SquareArea(7);

obj.CircleArea(3);

}

}

**o/p :**

Area of Rectangle: 50

Area of Square: 49

Area of Circle: 28.274333882308138

**// Repeat the above question for 4 rectangles, 4 squares and 5 circles. Hint- Use an array of objects.**

**→**

import java.lang.Math;

abstract class Shape

{

public abstract void calculateArea();

}

class Rectangle extends Shape

{

private int length;

private int breadth;

public Rectangle(int length, int breadth)

{

this.length = length;

this.breadth = breadth;

}

public void calculateArea()

{

int area = length \* breadth;

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

}

}

class Square extends Shape

{

private int side;

public Square(int side)

{

this.side = side;

}

public void calculateArea()

{

int area = side \* side;

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

}

}

class Circle extends Shape

{

private int radius;

public Circle(int radius)

{

this.radius = radius;

}

public void calculateArea()

{

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

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

}

}

public class Main

{

public static void main(String[] args)

{

Shape[] shapes = new Shape[13];

shapes[0] = new Rectangle(5, 10);

shapes[1] = new Rectangle(7, 12);

shapes[2] = new Rectangle(3, 8);

shapes[3] = new Rectangle(6, 9);

shapes[4] = new Square(7);

shapes[5] = new Square(9);

shapes[6] = new Square(11);

shapes[7] = new Square(5);

shapes[8] = new Circle(3);

shapes[9] = new Circle(5);

shapes[10] = new Circle(8);

shapes[11] = new Circle(2);

shapes[12] = new Circle(6);

for (Shape shape : shapes)

{

shape.calculateArea();

}

}

}

**o/p :**

Area of Rectangle: 50

Area of Rectangle: 84

Area of Rectangle: 24

Area of Rectangle: 54

Area of Square: 49

Area of Square: 81

Area of Square: 121

Area of Square: 25

Area of Circle: 28.274333882308138

Area of Circle: 78.53981633974483

Area of Circle: 201.06192982974676

Area of Circle: 12.566370614359172

Area of Circle: 113.09733552923254

**// Create a class named 'Member' having the following members :**

**Data members**

**1 - Name**

**2 - Age**

**3 - Phone number**

**4 - Address**

**5 - Salary**

**It also has a method named 'printSalary' which prints the salary of the members.**

**Two classes 'Employee' and 'Manager' inherit the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.**

**→**

import java.util.Scanner;

class Member

{

String name, number, address;

int age;

double salary;

public void printSalary()

{

System.out.println("\nSalary : "+salary);

}

}

class Employee extends Member

{

String specialization;

public void spec()

{

System.out.println("\nSpecialization : "+specialization);

}

}

class Manager extends Member

{

String department;

public void dept()

{

System.out.println("\nDepartment : "+department);

}

}

public class Ans

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

Employee e = new Employee();

System.out.print("Enter employee's name : ");

e.name = sc.next();

System.out.print("Enter employee's age : ");

e.age = sc.nextInt();

System.out.print("Enter employee's number : ");

e.number = sc.next();

System.out.print("Enter employee's address : ");

e.address = sc.next();

System.out.print("Enter employee's salary : ");

e.salary = sc.nextDouble();

System.out.print("Enter employee's specialization : ");

e.specialization = sc.next();

Manager m = new Manager();

System.out.print("Enter employee's department : ");

m.department = sc.next();

System.out.print("Name : "+e.name);

System.out.print("\nAge : "+e.age);

System.out.print("\nNumber : "+e.number);

System.out.print("\nAddress : "+e.address);

e.printSalary();

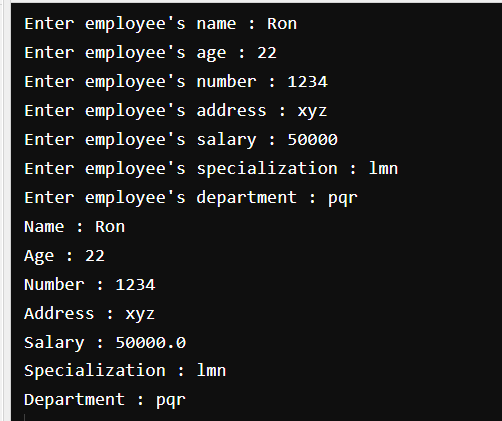
e.spec();

m.dept();

}

}

**o/p :**

****

**// Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.**

**→**

class Rectangle

{

int length, breadth;

public Rectangle(int l, int b)

{

length = l;

breadth = b;

}

void Area()

{

System.out.println("Area : " + (length \* breadth));

}

void Perimeter()

{

System.out.println("Perimeter : " + (2 \* (length + breadth)));

}

}

public class Square extends Rectangle

{

Square(int s)

{

super(s, s);

}

public static void main(String[] args)

{

Square square = new Square(5);

Rectangle rectangle = new Rectangle(3, 5);

System.out.println("Square's area and perimeter : ");

square.Area();

square.Perimeter();

System.out.println("Rectangle's area and perimeter : ");

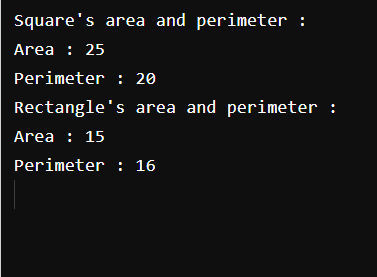
rectangle.Area();

rectangle.Perimeter();

}

}

**o/p :**

****

**// Exception**

**→**

class UserDefinedExceptionA extends Exception

{

public UserDefinedExceptionA(String str)

{

super(str);

}

}

public class throw\_1

{

public static void main(String args[])

{

try

{

throw new UserDefinedExceptionA("This is user defined exception");

}

catch(UserDefinedExceptionA ude)

{

System.out.println("Caught the exception");

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

}

}

}

**o/p :**

Caught the exception

This is user defined exception

**// MultiThreading**

**→**

class MyThread implements Runnable

{

String message;

MyThread(String msg)

{

message = msg;

}

public void run()

{

for(int count=0; count<=5; count++)

{

try

{

System.out.println(count + " Run method: " + message);

Thread.sleep(1000);

}

catch (InterruptedException ie)

{

System.out.println("Exception in thread: " + ie.getMessage());

}

}

}

}

public class MainThread

{

public static void main(String[] args)

{

MyThread obj1 = new MyThread("MyThread obj1");

MyThread obj2 = new MyThread("MyThread obj2");

Thread t1 = new Thread(obj1);

Thread t2 = new Thread(obj2);

t1.start();

t2.start();

}

}

**o/p :**

0 Run method: MyThread obj2

0 Run method: MyThread obj1

1 Run method: MyThread obj1

1 Run method: MyThread obj2

2 Run method: MyThread obj2

2 Run method: MyThread obj1

3 Run method: MyThread obj2

3 Run method: MyThread obj1

4 Run method: MyThread obj2

4 Run method: MyThread obj1

5 Run method: MyThread obj2

5 Run method: MyThread obj1

**// MutiThreading**

**→**

class ThreadTest extends Thread

{

private Thread thread;

private String ThreadName;

ThreadTest(String msg)

{

ThreadName = msg;

System.out.println("Creating Thread: "+ThreadName);

}

public void run()

{

System.out.println("Running Thread: "+ThreadName);

try

{

for(int i=0; i<5; i++)

{

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

Thread.sleep(1000);

}

}

catch(InterruptedException e)

{

System.out.println("Exception in thread" + ThreadName);

}

System.out.println("Thread: " + ThreadName + "continue");

}

public void start()

{

System.out.println("Start method" + ThreadName);

if(thread == null)

{

thread = new Thread(this, ThreadName);

thread.start();

}

}

}

**o/p :**

Creating Thread: First Thread

Start methodFirst Thread

Creating Thread: Second Thread

Start methodSecond Thread

Running Thread: First Thread

Creating Thread: Third Thread

Start methodThird Thread

Running Thread: Second Thread

Creating Thread: Fourth Thread

Start methodFourth Thread

Running Thread: Third Thread

Running Thread: Fourth Thread

Thread: First Thread,0

Thread: Second Thread,0

Thread: Third Thread,0

Thread: Fourth Thread,0

Thread: First Thread,1

Thread: Second Thread,1

Thread: Third Thread,1

Thread: Fourth Thread,1

Thread: Second Thread,2

Thread: Third Thread,2

Thread: Fourth Thread,2

Thread: First Thread,2

Thread: Second Thread,3

Thread: First Thread,3

Thread: Third Thread,3

Thread: Fourth Thread,3

Thread: Second Thread,4

Thread: Fourth Thread,4

Thread: First Thread,4

Thread: Third Thread,4

Thread: Fourth Threadcontinue

Thread: Second Threadcontinue

Thread: First Threadcontinue

Thread: Third Threadcontinue

**// MultiThreading**

**→**

import java.io.\*;

import java.util.Scanner;

class Fibonacci extends Thread

{

public void run()

{

try

{

int a=0, b=1, c=0;

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

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

System.out.println("Fibonacci series: ");

while(n>0)

{

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

a = b;

b = c;

c = a + b;

n = n - 1;

}

}

catch(Exception ex)

{

ex.printStackTrace();

}

}

}

class Prime extends Thread

{

public void run()

{

try

{

int i=1,j;

System.out.println("\nPrime numbers from 1-20 : ");

while(i<=20)

{

if (isPrime(i))

{

System.out.println(i);

}

i++;

}

}

catch(Exception ex)

{

ex.printStackTrace();

}

}

public boolean isPrime(int n)

{

if (n <= 1)

return false;

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

if (n % i == 0)

return false;

return true;

}

}

class Main

{

public static void main(String[] args)

{

try

{

Fibonacci obj1 = new Fibonacci();

obj1.start();

obj1.sleep(4000);

Prime obj2 = new Prime();

obj2.start();

obj2.sleep(4000);

}

catch (Exception ex)

{

ex.printStackTrace();

}

}

}

**o/p :**

Enter the limit: 6

Fibonacci series: 0 1 1 2 3 5

Prime numbers from 1-20 :

2

3

5

7

11

13

17

19

**// MultiThreading**

**→**

import java.io.\*;

public class ThreadPriority extends Thread

{

public void run()

{

System.out.println("run() method "+getName());

String threadName = Thread.currentThread().getName();

Integer threadPrio = Thread.currentThread().getPriority();

System.out.println(threadName + " has priority " + threadPrio);

}

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

{

ThreadPriority t1 = new ThreadPriority();

ThreadPriority t2 = new ThreadPriority();

ThreadPriority t3 = new ThreadPriority();

t1.setPriority(Thread.MAX\_PRIORITY);

t2.setPriority(Thread.MIN\_PRIORITY);

t3.setPriority(Thread.NORM\_PRIORITY);

t1.start();

t2.start();

t3.start();

}

}

**o/p :**

run() method Thread-1

run() method Thread-2

run() method Thread-0

Thread-1 has priority 1

Thread-2 has priority 5

Thread-0 has priority 10

**//Button using Swing**

**→**

import javax.swing.\*;

public class swing

{

public static void main(String args[])

{

JFrame a = new JFrame("example");

JButton b = new JButton("click me");

b.setBounds(40,90,85,20);

a.add(b);

a.setSize(300,300);

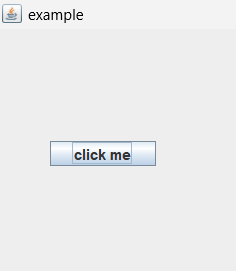
a.setLayout(null);

a.setVisible(true);

}

}

**o/p :**

****

**//Menu using Swing**

**→**

import javax.swing.\*;

class Example

{

JMenu menu;

JMenuItem a1,a2;

Example()

{

JFrame a = new JFrame("Example");

menu = new JMenu("options");

JMenuBar m1 = new JMenuBar();

a1 = new JMenuItem("example");

a2 = new JMenuItem("example1");

menu.add(a1);

menu.add(a2);

m1.add(menu);

a.setJMenuBar(m1);

a.setSize(400,400);

a.setLayout(null);

a.setVisible(true);

}

public static void main(String args[])

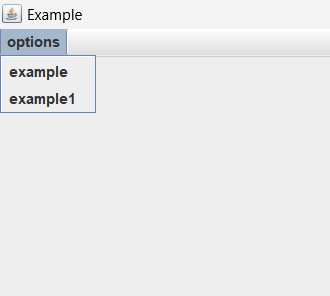
{

new Example();

}

}

**o/p :**

****

**//Login form using java**

**→**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class LoginFrame extends JFrame implements ActionListener

{

Container container=getContentPane();

JLabel userLabel=new JLabel("USERNAME");

JLabel passwordLabel=new JLabel("PASSWORD");

JTextField userTextField=new JTextField();

JPasswordField passwordField=new JPasswordField();

JButton loginButton=new JButton("LOGIN");

JButton resetButton=new JButton("RESET");

JCheckBox showPassword=new JCheckBox("Show Password");

LoginFrame()

{

//Calling methods inside the constructor.

setLayoutManager();

setLocationAndSize();

addComponentsToContainer();

}

public void setLayoutManager()

{

container.setLayout(null);

}

public void setLocationAndSize()

{

//Setting location and Size of each components using setBounds() method.

userLabel.setBounds(50,150,100,30);

passwordLabel.setBounds(50,220,100,30);

userTextField.setBounds(150,150,150,30);

passwordField.setBounds(150,220,150,30);

showPassword.setBounds(150,250,150,30);

loginButton.setBounds(50,300,100,30);

resetButton.setBounds(200,300,100,30);

}

public void addComponentsToContainer()

{

//Adding each components to the Container

container.add(userLabel);

container.add(passwordLabel);

container.add(userTextField);

container.add(passwordField);

container.add(showPassword);

container.add(loginButton);

container.add(resetButton);

}

@Override

public void actionPerformed(ActionEvent e)

{

}

}

public class login

{

public static void main(String[] a)

{

LoginFrame frame=new LoginFrame();

frame.setTitle("Login Form");

frame.setVisible(true);

frame.setBounds(10,10,370,600);

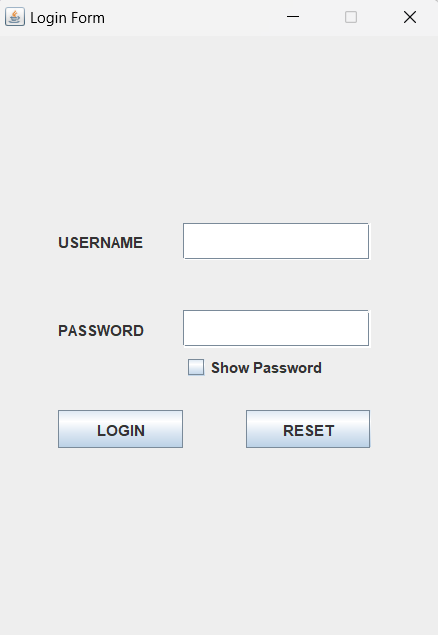
frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setResizable(false);

}

}

**o/p :**

****