**ASSIGNMENT 1**

**1.Write a Java program to change temperature from Celsius to Fahrenheit and vice versa**.(07 Nov)

import java.util.Scanner;

public class tempConv {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Temperature Converter");

System.out.print("Enter temperature (F or C): ");

char unit = scanner.next().toUpperCase().charAt(0);

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

double temperature = scanner.nextDouble();

double convertedTemperature;

if (unit == 'F') {

convertedTemperature = celsiusFromFahrenheit(temperature);

System.out.printf("%.2f°F is equal to %.2f°C\n", temperature, convertedTemperature);

} else if (unit == 'C') {

convertedTemperature = fahrenheitFromCelsius(temperature);

System.out.printf( "%.2f°C is equal to %.2f°F\n", temperature, convertedTemperature);

} else {

System.out.println("Invalid unit. Please enter 'F' or 'C'.");

}

scanner.close();

}

private static double celsiusFromFahrenheit(double fahrenheit) {

return (fahrenheit - 32) \* 5 / 9;

}

private static double fahrenheitFromCelsius(double celsius) {

return (celsius \* 9 / 5) + 32;

}

}

Output:

Temperature Converter

Enter temperature (F or C): C

Enter temperature value: 36

36.00°C is equal to 96.80°F

**2.Write a Java Program to check if a number is Positive or Negative.**

(07 Nov)

import java.util.Scanner;

public class posNeg {

@SuppressWarnings("resource")

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

if(num >0){

System.out.println("It is positive.");

}

else if (num==0){

System.out.println("It is neither positive nor negative");

}

else{

System.out.println("It is negative.");

}

}

}

Output:

Enter the number :

-7

It is negative.

**3.Write a Java program to find maximum of three numbers.**

**(09 Sept)**

import java.util.Scanner;

public class MaximunOfThreeNumbers {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter three numbers:");

int n1=scan.nextInt();

int n2=scan.nextInt();

int n3=scan.nextInt();

int num=(n1>n2)?(n1>n3?n1:n2):(n2>n3?n2:n3);

System.out.println("Greatest among all is:"+num);

}

}

Output:

Enter three numbers:

2

4

5

Greatest among all is:5

**4.Write a Java program to swap two numbers(09 Sept)**

import java.util.Scanner;

public class SwappingOfTwoNumbers {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter two numbersL:");

int n1=scan.nextInt();

int n2=scan.nextInt();

int n3;

n3=n1;

n1=n2;

n2=n3;

System.out.println("Value of n1 is:"+n1);

System.out.println("Value of n2 is :"+n2);

}

}

Output:

Enter two numbers:

3

4

Value of n1 is:4

Value of n2 is :3

**5.Write a Java program to convert miles to kilometers (07 Nov)**

import java.util.Scanner;

public class miletokilo{

@SuppressWarnings("resource")

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the distance in mile");

float m = sc.nextFloat();

float k = (float)(m\*1.609);

System.out.println("The distance in kilometer is :"+ k);

}

}

Output:

Enter the distance in mile

45

The distance in kilometer is :72.405

**6.Write a Java program to check whether a year is leap year or not.**

**(07 Nov)**

import java.util.Scanner;

public class leapyear {

@SuppressWarnings("resource")

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int year = sc.nextInt();

if(year%4==0){

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

}

else{

System.out.println("It is not");

}

}

}

Output:

Enter the year:

2005

It is not

**7.Write a Java program for following grading system. Note: Percentage>=90% : Grade A Percentage>=80% : Grade B Percentage>=70% : Grade C Percentage>=60% : Grade D Percentage>=40% : Grade E Percentage (07 Nov)**

import java.util.Scanner;

public class gradingSystem {

@SuppressWarnings("resource")

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the percentage of the student :");

float p = sc.nextFloat();

if(p>=90) System.out.println("Grade A");

else if(p>=80) System.out.println("Grade B");

else if(p>=70 ) System.out.println("Grade C");

else if(p>=60) System.out.println("Grade D");

else if(p>=50) System.out.println("Grade E");

else {System.out.println("Grade F");};

}

}

Output:

Enter the percentage of the student :

75

Grade C

**8.Write a Java program to check whether a number is divisible by a number given by user.(09 Sept)**

import java.util.Scanner;

public class Divisible {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter number to be divided:");

int n1=scan.nextInt();

System.out.println("Enter number with which you want division:");

int n2=scan.nextInt();

if(n1%n2==0)

{

System.out.println("Number is divisible");

}

else{

System.out.println("Number is not divisible");

}

}

}

Output:

Enter number to be divided:

6

Enter number with which you want division:

3

Number is divisible

**9.Write a Java program to calculate factorial of 12.(09 Sept)**

import java.util.Scanner;

public class Factorial {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter number of which you want to calculate factorial:");

int n=scan.nextInt();

int temp=1;

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

{

temp=temp\*i;

}

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

}

}

Output:

Enter number of which you want to calculate factorial:

5

Factorial of given number is:120

**10.Write a Java program for Fibonacci series (07 Nov)**

import java.util.Scanner;

public class fibonacci {

@SuppressWarnings("resource")

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter upto which you want fibonacci:");

int n=scan.nextInt();

int n1=0;

int n2=1;

System.out.println(n1);

System.out.println(n2);

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

{

int n3=n1+n2;

n1=n2;

n2=n3;

System.out.print(" "+n3+" ");

}

}

}

Output:

Enter upto which you want fibonacci:

5

0

1

1. 2 3 5 8

**11.Write a Java program to reverse a number. (09 Sept)**

import java.util.Scanner;

public class ReverseANumber {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter number to be reversed:");

int n=scan.nextInt();

int temp=0;

while(n!=0)

{

int a=n%10;

temp=temp\*10+a;

n=n/10;

}

System.out.println(temp);

}

}

Output:

Enter number to be reversed:

456

654

**12.Admission to a professional course is subject to the following conditions: (a) marks in Mathematics >= 60 (b) marks in Physics >=50 (c) marks in Chemistry >=40 (d) Total in all 3 subjects >=200 (Or) Total in Maths & Physics>=150 Given the marks in the 3 subjects of n (user input)(07 Nov)**

import java.util.Scanner;

class marks{

int p , c , m , total;

void setmarks( int physics , int chem, int math, int t){

p = physics;

c = chem;

m = math;

total = t;

}

String getMarks(){

if((p>=50)&&(c>=40)&&(m>=60)&&(total>=200)){

return "Eligible";

}

else if ((m+p>=150)){

return "Eligible";

}

else{

return "Not Eligible";

}

}

}

public class eligibilty {

@SuppressWarnings("resource")

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int p1 = sc.nextInt();

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

int c1 = sc.nextInt();

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

int m1 = sc.nextInt();

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

int t1 = sc.nextInt();

marks m2 = new marks();

m2.setmarks(p1,c1,m1,t1);

m2.getMarks();

}

}

**13.Write a Java program to calculate the sum of natural numbers up to a certain range.(09 Sept)**

import java.util.Scanner;

public class SumOfNaturalNumbers {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter Range to find sum:");

int n1=scan.nextInt();

int n2=scan.nextInt();

int count=0;

for(int i=n1;i<=n2;i++)

{

count=count+i;

}

System.out.println("Sum is: "+count);

}

}

Output:

Enter Range to find sum:

10

20

Sum is: 165

**14**. **Write a Java program to print all multiple of 10 between a given interval.(09 Sept)**

import java.util.Scanner;

public class MutlipleOfTen {

public static void main(String []args) {

Scanner sc= new Scanner(System.in);

System.out.println("Enter the Starting Integer");

int a=sc.nextInt();

System.out.println("Enter the end");

int b=sc.nextInt();

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

{

if(i%10==0)

{

System.out.println(i);

}

}

}

}

Output:

Enter the Starting Integer

20

Enter the end

45

20

30

40

**15.Write a Java program to generate multiplication table.(09 Sept)**

import java.util.Scanner;

public class Table {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter Number of which you want to generate table:");

int n=scan.nextInt();

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

{

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

}

}

}

Output:

Enter Number of which you want to generate table:

5

5 \* 1 = 5

5 \* 2 = 10

5 \* 3 = 15

5 \* 4 = 20

5 \* 5 = 25

5 \* 6 = 30

5 \* 7 = 35

5 \* 8 = 40

5 \* 9 = 45

5 \* 10 = 50

**16. Write a Java program to find HCF of two Numbers.(07 Nov)**

import java.util.Scanner;

public class HCF {

public static void main(String []args)

{

Scanner scan = new Scanner(System.in);

int n1,n2;

System.out.println("Enter two numbers of which you want HCF");

n1=scan.nextInt();

n2=scan.nextInt();

int num=(n1>n2)?n1:n2;

int i=2;

int num1=1;

while(i<=num)

{

if(n1%i==0 && n2%i==0)

{

num1=num1\*i;

n1=n1/i;

n2=n2/i;

}

else if(n1%i==0)

{

n1=n1/i;

}

else if(n2%i==0)

{

n2=n2/i;

}

i++;

}

System.out.println(num1);

}

}

Output:

Enter two numbers of which you want HCF

6

4

2

**17. Write a Java program to find LCM of two Numbers.(07 Nov)**

import java.util.Scanner;

public class lcm {

static int hcf(int a , int b){

while(b!=0){

int temp = b;

b = a%b;

a = temp;

}

return a ;

}

static int LCM( int a , int b , int hcf){

return (a\*b)/hcf;

}

@SuppressWarnings("resource")

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the first number");

int a1 = sc.nextInt();

System.out.println("Enter the second number ");

int b1 = sc.nextInt();

System.out.println("The LCM of two number is ");

System.out.println(LCM(a1,b1,hcf(a1, b1)));

}

}

Output:

Enter the first number

4

Enter the second number

7

The LCM of two number is

28

**18. Write a Java program to count the number of digits of an integer**

**(09 Sept)**

import java.util.Scanner;

public class NumberOfDigitsInInteger {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

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

int n=scan.nextInt();

int count=0;

while(n!=0)

{

n=n/10;

count++;

}

System.out.println("Number of digits:"+count);

}

}

Output:

Enter Number:

12345

Number of digits:5

**19. Write a Java program to check whether a number is palindrome or not.(09 Sept)**

import java.util.Scanner;

public class PalindromeOrNot {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

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

int n=scan.nextInt();

String s=String.valueOf(n);

String s1="";

for(int i=(s.length()-1);i>=0;i--)

{

s1=s1+s.charAt(i);

}

if(s1.equals(s))

{

System.out.println("Number is palindrome.");

}

else{

System.out.println("Number is not palindrome.");

}

}

Output:

Enter Number:

456

Number is not palindrome.

**20. Write a Java program to check whether a number is prime or not**.

**(09 Sept)**

import java.util.Scanner;

public class PrimeorNot {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

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

int n=scan.nextInt();

boolean a=false;

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

{

if(n%i==0)

{

a=true;

}

}

if(a==true)

{

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

}

else

{

System.out.println("prime");

}

}

}

Output:

Enter Number:

456

Entered number is not prime

**21. Write a Java program to convert a Binary Number to Decimal and Decimal to Binary**.**(07 Nov)**

public class convertor {

// decimal to binary

static void decimaltoBinary(int n){

int[] binaryNum = new int[1000];

int i = 0;

while (n > 0)

{

binaryNum[i] = n % 2;

n = n / 2;

i++;

}

for (int j = i - 1; j >= 0; j--)

System.out.print(binaryNum[j]);

}

static void binarytoDecimal(int n){

int base = 1;

int num = n;

int temp = num;

int dec = 0;

while(temp>0){

int last\_digit = temp%10;

temp = temp/10;

dec = dec + last\_digit\*base;

base = base\*2;

}

System.out.println(dec);

}

public static void main(String[] args) {

System.out.println("The binary num of 10 is ");

decimaltoBinary(10);

System.out.println();

System.out.println("The decimal of 111 is");

int num1 = 111;

binarytoDecimal(num1);

}

}

Output:

The binary num of 10 is

1010

The decimal of 111 is

7

**22. Write a Java program to find median of a set of numbers.( 09 Sept)**

import java.util.Scanner;

public class MedianOfSetOfNumbers {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter the number elements to be entered on array");

int n=scan.nextInt();

int arr[]=new int[n];

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

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

{

arr[i]=scan.nextInt();

}

int count=0;

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

{

count++;

}

System.out.println(count);

if(n%2!=0)

{

System.out.println("Median of the given data is:"+arr[((count+1)/2)-1]);

}

else{

double medianInCaseOfEvenNumberOfElements=(arr[(count/2)-1]+arr[count/2])/2.0;

System.out.println("Median of the given set of numbers is :"+medianInCaseOfEvenNumberOfElements);

}

}

}

Output:

Enter the number elements to be entered on array

4

Enter the elements in array

1

2

3

4

4

Median of the given set of numbers is :2.5

**23. Write Java programs for the patterns given bellow: (a) 1 2 3 4 5 6 7 8 9**

**(09 Sept)**

import java.util.Scanner;

public class NumberPattern {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter number of rows upto which you want pattern:");

int n=scan.nextInt();

int a=1;

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

{

for(int j=1;j<=2\*i-1;j++)

{

System.out.print(a);

a++;

}

System.out.println();

}

}

}

Output:

Enter number of rows upto which you want pattern:

3

1

234

56789

**24. Write a Java program to calculate Sum & Average of an integer array.**

**(09 Sept)**

import java.util.Scanner;

public class SumAndAverage {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter the number elements to be entered on array");

int n=scan.nextInt();

int arr[]=new int[n];

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

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

{

arr[i]=scan.nextInt();

}

int sum=0;

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

{

sum=arr[i]+sum;

}

System.out.println("Sum of all the elements present in given array:"+sum);

double average;

average=sum/n;

System.out.println("Average of the given array is:"+average);

}

}

Output:

Enter the number elements to be entered on array

4

Enter the elements in array

1

2

3

4

Sum of all the elements present in given array:10

Average of the given array is:2.0

**25. Write a Java program to implement stack using array.(07 Nov)**

public class Stack {

int top = -1;

int [] arr = new int[12];

void push(int a){

if(top>=(arr.length)-1){

System.out.println("Stack overflow.");

}

else{

top++;

arr[top] = a;

}

}

int pop(){

if(top <0){

System.out.println("Stack underflow");

return 0;

}

else{

top--;

int a = arr[top];

return a ;

}

}

void display(){

for(int i = top;i>-1;i--){

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

}

}

public static void main(String[] args) {

Stack s = new Stack();

s.push(11);

s.push(12);

s.push(13);

s.display();

System.out.println("Deleted element :"+s.pop());

s.display();

}

}

Output:

13

12

11

Deleted element :12

12

11

**26. Write a Java program to implement Queue using array.(07 Nov)**

public class Queue {

int f,e,size ;

int[]queue;

Queue(int c ){

f = 0;

size = c;

e=-1;

queue = new int[size];

}

void enqueue(int data){

if(e==size-1){

System.out.println("Queue is full");

}

queue[++e]= data;

}

void dequeue(){

if(f>e){

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

return ;

}

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

queue[i]=queue[i+1];

}

e--;

}

void display(){

for(int i = f;i<=e;i++){

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

}

}

public static void main(String[] args) {

Queue q = new Queue(4);

q.enqueue(20);

q.enqueue(90);

q.display();

}

}

Output:

20 90

**27. Write a Java program to enter n elements in an array and find smallest number among them.(09 Sept)**

import java.util.Scanner;

public class SmallestNumberInGivenArray {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter the number elements to be entered on array");

int n=scan.nextInt();

int arr[]=new int[n];

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

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

{

arr[i]=scan.nextInt();

}

int temp;

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

{

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

{

if(arr[i]>arr[j])

{

temp=arr[j];

arr[j]=arr[i];

arr[i]=temp;

}

}

}

System.out.println("Smallest element is:"+arr[0]);

}

}

Output:

Enter the number elements to be entered on array

6

Enter the elements in array

2 3 4 89 6 7

Smallest element is:2

**28. Write Java program to find the sum of all odd numbers in a array.(09 Sept)**

import java.util.Scanner;

public class SumOfOddNumbersInAnArray {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter the number elements to be entered on array");

int n=scan.nextInt();

int arr[]=new int[n];

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

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

{

arr[i]=scan.nextInt();

}

System.out.println("Odd number present in array:");

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

{

if(arr[i]%2!=0)

{

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

}

}

}

}

Output:

Enter the number elements to be entered on array

6

Enter the elements in array

3 5 4 6 7 8

Odd number present in array:

3 5 7

Sum

15

**29. Write a Java program to find duplicate elements in a 1D array and find their frequency of occurrence.(07 Nov)**

import java.util.Arrays;

public class duplicateElements {

public static void main(String[] args) {

int[] array = { 2, 3, 5, 4, 3, 1, 3, 2, 1, };

Arrays.sort(array);

int i,j,frequency;

System.out.println("These elements are repeated along with its frequency-");

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

frequency = 1;

for(j=i+1; j<array.length; j++){

if(array[j] == array[i]){

frequency++;

} else {

break;

}

}

i=j-1;

if(frequency > 1){

System.out.println(array[i] + " --> " + frequency);

}

}

}

}

Output:

These elements are repeated along with its frequency-

1 --> 2

2 --> 2

3 --> 3

**30. Write a Java program to print every alternate number of a given array(09 Sept)**

import java.util.Scanner;

public class AlternateElementFromTheGivenArray {

public static void main(String []args)

{

Scanner scan=new Scanner(System.in);

System.out.println("Enter the number elements to be entered on array");

int n=scan.nextInt();

int arr[]=new int[n];

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

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

{

arr[i]=scan.nextInt();

}

System.out.println("Alternate Elements of provided array:");

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

{

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

}

}

}

Output:

Enter the number elements to be entered on array

5

Enter the elements in array

1 2 3 4 5

Alternate Elements of provided array:

1 3 5

**31. Write a Java program to show 0-arguments constructor.(09 Sept)**

class Demo

{

private int a;

private int b;

public Demo()

{

System.out.println("Default Constructor is called");

a=10;

b=20;

}

public void disp()

{

System.out.println("Value of integer a is :"+a);

System.out.println("Value of integer b is :"+b);

}

}

public class DefaultConstructor {

public static void main(String []args)

{

Demo d=new Demo();

d.disp();

}

}

Output:

Default Constructor is called

Value of integer a is :10

Value of integer b is :20

**32. Write a Java program to show parameterized constructor.(09 Sept)**

class Demo{

private int a;

private int b;

public Demo(int a,int b)

{

this.a=a;

this.b=b;

System.out.println("Parmeterized Constructor is called.");

}

public void disp()

{

System.out.println("Value of first integer is:"+a);

System.out.println("Value of second integer is:"+b);

}

}

public class ParameterizedConstructor {

public static void main(String []args)

{

Demo d=new Demo(4,5);

d.disp();

}

}

Output:

Parmeterized Constructor is called.

Value of first integer is:4

Value of second integer is:5

**33. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission.Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message “Invalid Input**”**(07 Nov)**

import java.util.Scanner;

class demo{

int sales ;

demo( int x ){

sales = x;

}

int com( ){

int t = (int)(0.1\*sales);

return t;

}

}

public class comission{

@SuppressWarnings("resource")

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the sales ");

int s = sc.nextInt();

if(s<0){

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

}

demo c1 = new demo(s);

System.out.println("The comission is of Rs."+c1.com());

}

}

Output:

Enter the sales

2300

The comission is of Rs.230

**ASSIGNMENT 2**

**1. Given are two one-dimensional arrays A & B, which are sorted in ascending order. Write a Java program to merge them into single sorted array C that contains every item from arrays A & B, in ascending order(16 Nov)**

import java.util.Arrays;

public class MergeSortedArrays {

public static int[] mergeArrays(int[] A, int[] B) {

int[] C = new int[A.length + B.length];

int i = 0, j = 0, k = 0;

while (i < A.length && j < B.length) {

if (A[i] <= B[j]) {

C[k] = A[i];

i++;

} else {

C[k] = B[j];

j++;

}

k++;

}

while (i < A.length) {

C[k] = A[i];

i++;

k++;

}

while (j < B.length) {

C[k] = B[j];

j++;

k++;

}

return C;

}

public static void main(String[] args) {

int[] A = {1, 3, 5, 7};

int[] B = {2, 4, 6, 8};

int[] C = mergeArrays(A, B);

System.out.println("Merged Sorted Array: " + Arrays.toString(C));

}

}

**Output:**

Merged Sorted Array: [1, 2, 3, 4, 5, 6, 7, 8]

**2. Write a Java program to show 0-arguments constructor. (16 Nov)**

class ZeroArgumentConstructor

{

public ZeroArgumentConstructor()

{

System.out.println("ZERO AGRUMENT CONSTRUCTOR IS CALLED");

}

}

public class Constructor {

public static void main(String []args)

{

ZeroArgumentConstructor a= new ZeroArgumentConstructor();

}

}

**Output:**

ZERO AGRUMENT CONSTRUCTOR IS CALLED

**3. Write a Java program to show parameterized constructor**. **(16 Nov)**

class ParameterizedConstructorImplementation

{

public ParameterizedConstructorImplementation(int x,int y)

{

int z=x+y;

System.out.println("PARAMETRIZED CONSTRUCTOR IS CALLED.");

System.out.println("SUM OF INTEGERS PASSED IS:"+z);

}

}

public class Main {

public static void main(String []args)

{

ParameterizedConstructorImplementation a= new ParameterizedConstructorImplementation(4,5);

}

}

**OUTPUT:**

PARAMETRIZED CONSTRUCTOR IS CALLED.

SUM OF INTEGERS PASSED IS:9

**4. Write a Java program to show constructor overloading. (16 Nov)**

class StudentData

{ private String name;

private int age;

public StudentData()

{

System.out.println("Default constructor is called");

name="Rahul";

age=18;

System.out.println("Name of the Student is:"+name);

System.out.println("Age of the Student is:"+age);

}

public StudentData(String Name)

{

System.out.println(" Constructor having one argumnet is called");

this.name=Name;

age=20;

System.out.println("Name of the Student is:"+Name);

System.out.println("Age of the Student is:"+age);

}

public StudentData(String Name , int Age)

{

System.out.println(" Constructor having two argumnets is called");

this.name=Name;

this.age=Age;

System.out.println("Name of the Student is:"+Name);

System.out.println("Age of the Student is:"+Age);

}

}

public class Main {

public static void main(String []args)

{

StudentData S1= new StudentData();

StudentData S2= new StudentData("Nitin");

StudentData S3= new StudentData("Raman",21);

}

}

**Output:**

Default constructor is called

Name of the Student is:Rahul

Age of the Student is:18

Constructor having one argumnet is called

Name of the Student is:Nitin

Age of the Student is:20

Constructor having two argumnets is called

Name of the Student is:Raman

Age of the Student is:21

**5. Write a Java program to implement the concept of inheritance. (16 Nov)**

class ParentClass

{

void Method1()

{

System.out.println("METHOD OF PARENT CLASS");

}

}

class ChildClass extends ParentClass

{

void Method2()

{

System.out.println("METHOD OF CHILD CLASS");

}

}

public class Inheritance {

public static void main(String []args)

{

ChildClass obj= new ChildClass();

obj.Method1();//We created the object of child class and accessing the method of parent class.

//This is possible because of the concept of inheritance.

obj.Method2();

}

}

**Output:**

METHOD OF PARENT CLASS

METHOD OF CHILD CLASS

**6. Write a Java program to show method overloading. (16 Nov)**

class SumCalculator

{

int add(int a,int b)

{

return a+b;

}

int add(int a, int b , int c)

{

return a+b+c;

}

}

public class MethodOverloading {

public static void main(String []args)

{

SumCalculator obj1 = new SumCalculator();

System.out.println(obj1.add(4,5));

SumCalculator obj2 = new SumCalculator();

System.out.println(obj2.add(4, 5,6));

}

};

**Output:**

9

15

**7. Write a Java program to show method overriding. (16 Nov)**

class Human

{

void sleep()

{

System.out.println("Humans need a good sleep for healthy body");

}

}

class Student extends Human

{

void sleep()

{

System.out.println("Each and every student should sleep for 8 hours");

}

}

public class Main {

public static void main(String []args)

{

Student S1 = new Student();

Human H1 = new Human();

S1.sleep();

H1.sleep();

}

}

**Output:**

Each and every student should sleep for 8 hours

Humans need a good sleep for healthy body

**8. Write a Java program to show method hiding. (16 Nov)**

class Parent

{

static void display() {

System.out.println("Method of parent class");

}

}

class Child extends Parent

{

static void display()

{

System.out.println("Display method of child class");

}

}

public class MethodHiding {

public static void main(String []args)

{

Parent P1 = new Parent();

Parent obj= new Child();

Child C1 =new Child();

P1.display();

obj.display();

C1.display();

}

}

**Output:**

Method of parent class

Method of parent class

Display method of child class

**9. Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea ( ) and volume (). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes. (16 Nov)**

import java.util.Scanner;

class ThreeDObject

{

double wholeSurfaceArea()

{

System.out.println("This method will return the total Surface area of the object");

return 0;

}

double volume()

{

System.out.println("This method will return the volume of the object");

return 0;

}

}

class Box extends ThreeDObject

{

double length;

double breadth;

double height;

public Box(double a,double b,double c)

{

length=a;

breadth=b;

height=c;

}

double wholeSurfaceArea()

{

double TSA=2\*(length\*breadth+breadth\*height+height\*length);

return TSA;

}

double volume()

{

double v=length\*breadth\*height;

return v;

}

}

class Cube extends ThreeDObject

{

double side;

public Cube(double a)

{

side=a;

}

double wholeSurfaceArea()

{

double TSA=6\*side\*side;

return TSA;

}

double volume()

{

double v=side\*side\*side;

return v;

}

}

class Cylinder extends ThreeDObject

{

double radius;

double height;

public Cylinder(double a,double b)

{

radius=a;

height=b;

}

double wholeSurfaceArea()

{

double TSA=2\*3.14\*radius\*(radius+height);

return TSA;

}

double volume()

{

double v=3.14\*radius\*radius\*height;

return v;

}

}

class Cone extends ThreeDObject

{

double radius;

double height;

double slantheight;

public Cone(double a,double b, double c)

{

radius=a;

height=b;

slantheight=c;

}

double wholeSurfaceArea()

{

double TSA=3.14\*radius\*(radius+slantheight);

return TSA;

}

double volume()

{

double v=3.14\*radius\*radius\*slantheight;

return v;

}

}

public class MainClass {

public static void main(String []args)

{

Scanner scan = new Scanner(System.in);

System.out.println("Enter Dimensions of a box:");

double a=scan.nextDouble();

double b=scan.nextDouble();

double c=scan.nextDouble();

Box B1 = new Box(a,b,c);

System.out.println("Total surface area of the box is"+B1.wholeSurfaceArea());

System.out.println("Volume of the box is:"+B1.volume());

System.out.println("Enter Dimensions of a cube:");

double obj=scan.nextDouble();

Cube C1 = new Cube(obj);

System.out.println("Total surface area of the cube is"+C1.wholeSurfaceArea());

System.out.println("Volume of the cube is:"+C1.volume());

System.out.println("Enter Dimensions of a Cylinder:");

double obj1=scan.nextDouble();

double obj2=scan.nextDouble();

Cylinder C2 = new Cylinder(obj1,obj2);

System.out.println("Total surface area of the cyclinder is"+C2.wholeSurfaceArea());

System.out.println("Volume of the cyclinder is:"+C2.volume());

System.out.println("Enter Dimensions of a Cone:");

double obj3=scan.nextDouble();

double obj4=scan.nextDouble();

double obj5= scan.nextDouble();

Cone C3 = new Cone(obj3,obj4,obj5);

System.out.println("Total surface area of the cone is"+C3.wholeSurfaceArea());

System.out.println("Volume of the cone is:"+C3.volume());

}

}

**Output:**

Enter Dimensions of a box:

12

23

34

Total surface area of the box is2932.0

Volume of the box is:9384.0

Enter Dimensions of a cube:

5

Total surface area of the cube is150.0

Volume of the cube is:125.0

Enter Dimensions of a Cylinder:

3

4

Total surface area of the cyclinder is131.88

Volume of the cyclinder is:113.03999999999999

Enter Dimensions of a Cone:

5

6

7

Total surface area of the cone is188.4

Volume of the cone is:549.5

**10. Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car respectively with content of the super class’s showData ( ) method. (16 Nov)**

class Vehicle {

protected String regnNumber;

protected int speed;

protected String color;

protected String ownerName;

public Vehicle(String regnNumber, int speed, String color, String ownerName) {

this.regnNumber = regnNumber;

this.speed = speed;

this.color = color;

this.ownerName = ownerName;

}

void showData() {

System.out.println("This is a vehicle class.");

System.out.println("Registration Number: " + regnNumber);

System.out.println("Speed: " + speed + " km/h");

System.out.println("Color: " + color);

System.out.println("Owner Name: " + ownerName);

}

}

class Bus extends Vehicle {

private int routeNumber;

public Bus(String regnNumber, int speed, String color, String ownerName, int routeNumber) {

super(regnNumber, speed, color, ownerName);

this.routeNumber = routeNumber;

}

@Override

public void showData() {

super.showData(); // Calling the parent class showData

System.out.println("Route Number: " + routeNumber);

}

}

class Car extends Vehicle {

private String manufacturerName;

public Car(String regnNumber, int speed, String color, String ownerName, String manufacturerName) {

super(regnNumber, speed, color, ownerName);

this.manufacturerName = manufacturerName;

}

@Override

public void showData() {

super.showData();

System.out.println("Manufacturer Name: " + manufacturerName);

}

}

public class Main {

public static void main(String[] args) {

Bus bus = new Bus("KA01-1234", 60, "Yellow", "John Doe", 101);

System.out.println("\nBus Details:");

bus.showData();

Car car = new Car("MH12-5678", 120, "Red", "Alice Smith", "Toyota");

System.out.println("\nCar Details:");

car.showData();

}

}

**Output:**

Color: Yellow

Owner Name: John Doe

Route Number: 101

Car Details:

This is a vehicle class.

Registration Number: MH12-5678

Speed: 120 km/h

Color: Red

Owner Name: Alice Smith

Manufacturer Name: Toyota

**11. Write a Java program which creates a base class Num and contains an integer number along with a method shownum() which displays the number. Now create a derived class HexNum which inherits Num and overrides shownum() which displays the hexadecimal value and octal value of the number. Demonstrate the working of the classes. (16 Nov)**

import java.util.\*;

class Num

{

int a;

void shownum(int x)

{ a=x;

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

}

}

class HexNum extends Num

{

void shownum(int a)

{

String str=Integer.toHexString(a);

String str1=Integer.toOctalString(a);

System.out.println("The number in Hexa decimal form: "+str);

System.out.println("The number in octal number System is: "+str1);

}

}

public class MainClass11 {

public static void main(String []args)

{

HexNum A1= new HexNum();

A1.shownum(70);

}

}

**Output:**

The number in Hexa decimal form: 46

The number in octal number System is: 106

**12.Create a base class Distance which stores the distance between two locations in miles and a method travelTime(). The method prints the time taken to cover the distance when the speed is 60 miles per hour. Now in a derived class DistanceMKS, override travelTime() so that it prints the time assuming the distance is in kilometers and the speed is 100 km per second. Demonstrate the working of the classes. (16 Nov)**

class Distance

{

double distanceInMiles;

void travelTime(double x)

{ distanceInMiles=x;

double time=distanceInMiles/60;

System.out.println("The time taken to cover "+ distanceInMiles+"miles is "+time+" hours");

}

}

class DistanceMKS extends Distance

{

void travelTime(double x)

{ distanceInMiles=x;

double time=distanceInMiles\*1.609/100;

System.out.println("The time taken to cover "+ distanceInMiles\*1.609+ " km is"+ time+" hours when speed is 100 km per hour");

}

}

public class MainClass12 {

public static void main(String []args)

{

Distance D1 = new Distance();

DistanceMKS D2 = new DistanceMKS();

D1.travelTime(100);

D2.travelTime(100);

}

}

**Output:**

The time taken to cover 100.0miles is 1.6666666666666667 hours

The time taken to cover 160.9 km is1.609 hours when speed is 100 km per hour

**12.Write a Java program to explain “multilevel inheritance.” (16 Nov)**

//Program to demonstrate multilevel inheritance

class Animals

{

void Method1()

{

System.out.println("There are wide variety of animal species on planet earth");

}

}

class Cat extends Animals

{

void Method2()

{

System.out.println("Cats are the animals that live on land");

}

}

class Tiger extends Cat

{

void Method3()

{

System.out.println("Tiger belongs to cat family");

}

}

public class MainClass13 {

public static void main(String []args)

{

Tiger T1 = new Tiger();

T1.Method1();//Method of animal class.

T1.Method2();//Method of cat class.

T1.Method3();//Method of Tiger class.

}

}

**Output:**

There are wide variety of animal species on planet earth

Cats are the animals that live on land

Tiger belongs to cat family

**14. Write a program to define a class Employee to accept emp\_id, emp \_name, basic\_salary from the user and display the gross\_salary.**

**(16 Nov)**

import java.util.Scanner;

class Employee

{

int emp\_id;

String emp\_name;

double basic\_salary;

Employee(int a , String b , double c)

{

emp\_id=a;

emp\_name=b;

basic\_salary=c;

}

void gross\_salary()

{

double a= basic\_salary-(18\*basic\_salary)/100;

System.out.println("Gross salary after deducting 18 percent gst: "+ a);

}

}

public class MainClass14 {

public static void main(String []args)

{ Scanner scan = new Scanner(System.in);

System.out.println("Enter Employee id ,employee name and salary");

int a=scan.nextInt();

String b= scan.next();

double c=scan.nextDouble();

Employee E1 = new Employee(a,b,c);

E1.gross\_salary();

}

}

**Output:**

Enter Employee id ,employee name and salary

12

Harsh

4500

Gross salary after deducting 18 percent gst: 3690.0

**15. Write a program to demonstrate use of 'this' keyword. (16 Nov)**

class Animal

{

public Animal()

{

System.out.println("Default constructor of animal class.");

}

public Animal(int a)

{ this();

System.out.println("Parametized constructor of animal class.");

}

}

public class MainClass15 {

public static void main(String []args)

{

Animal A1=new Animal();

Animal A2=new Animal(4);

}

}

**Output:**

Default constructor of animal class.

Default constructor of animal class.

Parametized constructor of animal class

**16. Write a program to demonstrate use of 'static' keyword. (16 Nov)**

class Counter {

static int count = 0;

static void incrementCount() {

count++;

System.out.println("Count: " + count);

}

}

public class StaticDemo {

public static void main(String[] args) {

System.out.println("First call to incrementCount:");

Counter.incrementCount();

System.out.println("Second call to incrementCount:");

Counter.incrementCount();

System.out.println("Third call to incrementCount:");

Counter.incrementCount();

}

}

**Output:**

First call to incrementCount:

Count: 1

Second call to incrementCount:

Count: 2

Third call to incrementCount:

Count: 3

**17. Write program, which finds the sum of numbers formed by consecutive digits. (16 Nov)**

public class SumOfConsecutiveDigits {

public static void main(String[] args) {

String input = "2415";

int sum = 0;

for (int i = 0; i < input.length() - 1; i++) {

String pair = input.substring(i, i + 2);

sum += Integer.parseInt(pair);

}

System.out.println("The sum of consecutive numbers is: " + sum);

}

}

**Output:**

The sum of consecutive numbers is: 80

**18. Create three interfaces, each with two methods. Inherit a new interface from the three, adding a new method. Create a class by implementing the new interface and also inheriting from a concrete class. Now write four methods, each of which takes one of the four interfaces as an argument. In main ( ), create an object of your class and pass it to each of the methods. (16 Nov)**

interface Interface1 {

void method1();

void method2();

}

interface Interface2 {

void method3();

void method4();

}

interface Interface3 {

void method5();

void method6();

}

interface Interface4 extends Interface1, Interface2, Interface3 {

void method7();

}

class ConcreteClass implements Interface4 {

public void method1() {

System.out.println("Method 1 from Interface1");

}

public void method2() {

System.out.println("Method 2 from Interface1");

}

public void method3() {

System.out.println("Method 3 from Interface2");

}

public void method4() {

System.out.println("Method 4 from Interface2");

}

public void method5() {

System.out.println("Method 5 from Interface3");

}

public void method6() {

System.out.println("Method 6 from Interface3");

}

public void method7() {

System.out.println("Method 7 from Interface4");

}

}

public class MainClass {

public static void useInterface1(Interface1 obj) {

obj.method1();

obj.method2();

}

public static void useInterface2(Interface2 obj) {

obj.method3();

obj.method4();

}

public static void useInterface3(Interface3 obj) {

obj.method5();

obj.method6();

}

public static void useInterface4(Interface4 obj) {

obj.method1();

obj.method2();

obj.method3();

obj.method4();

obj.method5();

obj.method6();

obj.method7();

}

public static void main(String[] args) {

ConcreteClass concreteObj = new ConcreteClass();

System.out.println("Using Interface1:");

useInterface1(concreteObj);

System.out.println("\nUsing Interface2:");

useInterface2(concreteObj);

System.out.println("\nUsing Interface3:");

useInterface3(concreteObj);

System.out.println("\nUsing Interface4:");

useInterface4(concreteObj);

}

}

**Output:**

Using Interface1:

Method 1 from Interface1

Method 2 from Interface1

Using Interface2:

Method 3 from Interface2

Method 4 from Interface2

Using Interface3:

Method 5 from Interface3

Method 6 from Interface3

Using Interface4:

Method 1 from Interface1

Method 2 from Interface1

Method 3 from Interface2

Method 4 from Interface2

Method 5 from Interface3

Method 6 from Interface3

Method 7 from Interface4

**19. Write a Java program to show the use of all keywords for exception handling. (16 Nov)**

import java.io.IOException;

public class ExceptionHandlingDemo {

public static void readFile() throws IOException {

throw new IOException("File not found");

}

public static void main(String[] args) {

try {

System.out.println("Inside try block");

readFile();

} catch (IOException e) {

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

throw new ArithmeticException("Arithmetic Exception thrown from catch block");

} catch (Exception e) {

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

} finally {

System.out.println("Finally block executed, cleanup can be done here");

}

}

}

**Output:**

Inside try block

Caught exception: File not found

Finally block executed, cleanup can be done here

Exception in thread "main" java.lang.ArithmeticException: Arithmetic Exception thrown from catch block

at MainClass16.main(MainClass16.java:19)

**20. Write a Java program using try and catch to generate NegativeArrayIndex Exception and Arithmetic Exception. (16 Nov)**

public class ExceptionDemo {

public static void main(String[] args) {

try {

System.out.println("Attempting to create an array with negative size...");

int size = -5;

int[] arr = new int[size];

} catch (NegativeArraySizeException e) {

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

}

try {

System.out.println("\nAttempting to divide by zero...");

int result = 10 / 0;

} catch (ArithmeticException e) {

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

}

}

}

**Output:**

Attempting to create an array with negative size...

Caught NegativeArraySizeException: -5

Attempting to divide by zero...

Caught ArithmeticException: / by zero

**21. Write a program that outputs the name of the capital of the country entered at the command line. The program should throw a “NoMatchFoundException” when it fails to print the capital of the country entered at the command line. (16 Nov)**

class NoMatchFoundException extends Exception {

public NoMatchFoundException(String message) {

super(message);

}

}

public class CountryCapital {

public static String getCapital(String country) throws NoMatchFoundException {

String[] countries = {"USA", "India", "France", "Germany", "Canada", "Australia", "Japan"};

String[] capitals = {"Washington D.C.", "New Delhi", "Paris", "Berlin", "Ottawa", "Canberra", "Tokyo"};

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

if (countries[i].equalsIgnoreCase(country)) {

return capitals[i];

}

}

throw new NoMatchFoundException("No match found for the country: " + country);

}

public static void main(String[] args) {

if (args.length == 0) {

System.out.println("Please provide a country name as a command line argument.");

return;

}

String country = args[0];

try {

String capital = getCapital(country);

System.out.println("The capital of " + country + " is " + capital + ".");

} catch (NoMatchFoundException e) {

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

}

}

}

Output:

The capital of France is Paris

**22. Write a java program to create an custom Exception that would handle at least 2 kind of Arithmetic Exceptions while calculating a given equation. (16 Nov)**

class ArithmeticOperationException extends Exception {

public ArithmeticOperationException(String message) {

super(message);

}

}

public class CustomArithmeticExceptionDemo {

public static int calculate(String operation, int a, int b) throws ArithmeticOperationException {

int result = 0;

try {

switch (operation) {

case "divide":

if (b == 0) {

throw new ArithmeticOperationException("Error: Division by zero is not allowed.");

}

result = a / b;

break;

case "add":

if ((a > 0 && b > Integer.MAX\_VALUE - a) || (a < 0 && b < Integer.MIN\_VALUE - a)) {

throw new ArithmeticOperationException("Error: Integer overflow occurred during addition.");

}

result = a + b;

break;

default:

throw new ArithmeticOperationException("Unsupported operation.");

}

} catch (ArithmeticOperationException e) {

throw e;

}

return result;

}

public static void main(String[] args) {

try {

System.out.println("Result of 10 / 2: " + calculate("divide", 10, 2));

System.out.println("Result of 10 / 0: " + calculate("divide", 10, 0));

} catch (ArithmeticOperationException e) {

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

}

try {

System.out.println("Result of 2147483647 + 1: " + calculate("add", Integer.MAX\_VALUE, 1));

} catch (ArithmeticOperationException e) {

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

}

}

}

**Output:**

Result of 10 / 2: 5

Error: Division by zero is not allowed.

Error: Integer overflow occurred during addition.

**23. Create two user-defined exceptions named “TooHot” and “TooCold” to check the temperature (in Celsius) given by the user passed through the command line is too hot or too cold. If temperature > 35, throw exception “TooHot”. If temperature <5, throw exception “TooCold”. Otherwise, print “Normal” and convert it to Farenheit. (16 Nov)**

import java.util.Scanner;

class TooHot extends Exception {

public TooHot(String message) {

super(message);

}

}

class TooCold extends Exception {

public TooCold(String message) {

super(message);

}

}

public class TemperatureCheck {

public static void checkTemperature(double tempCelsius) throws TooHot, TooCold {

if (tempCelsius > 35) {

throw new TooHot("The temperature is too hot!");

} else if (tempCelsius < 5) {

throw new TooCold("The temperature is too cold!");

} else {

double tempFahrenheit = (tempCelsius \* 9 / 5) + 32;

System.out.println("The temperature is normal. In Fahrenheit: " + tempFahrenheit + "°F");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

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

double tempCelsius = scanner.nextDouble();

checkTemperature(tempCelsius);

} catch (TooHot hotError) {

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

} catch (TooCold coldError) {

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

} catch (Exception e) {

System.out.println("Invalid input! Please enter a valid number for the temperature.");

} finally {

scanner.close();

}

}

}

**Output**:

Enter the temperature in Celsius: 37

The temperature is too hot!

**24. Consider an Employee recruitment system that prints the candidate name based on the age criteria. The name and age of the candidate are taken as Input.Create two user-defined exceptions named “TooOlder” and “TooYounger” If age>45, throw exception “TooOlder”. If age<20, throw exception “TooYounger”. Otherwise, print “Eligible” and print the name of the** **candidate**. **(16 Nov)**

import java.util.Scanner;

class TooOlder extends Exception {

public TooOlder(String message) {

super(message);

}

}

class TooYounger extends Exception {

public TooYounger(String message) {

super(message);

}

}

public class EmployeeRecruitment {

public static void checkEligibility(String name, int age) throws TooOlder, TooYounger {

if (age > 45) {

throw new TooOlder("The candidate is too older!");

} else if (age < 20) {

throw new TooYounger("The candidate is too younger!");

} else {

System.out.println("Eligible");

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

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

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

String name = scanner.nextLine();

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

int age = scanner.nextInt();

checkEligibility(name, age);

} catch (TooOlder olderError) {

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

} catch (TooYounger youngerError) {

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

} catch (Exception e) {

System.out.println("Invalid input! Please enter valid data.");

} finally {

scanner.close();

}

}

}

**Output:**

Enter candidate name: Nitin

Enter candidate age: 60

The candidate is too older!

**25. Write a program to raise a user defined exception if username is less than 6 characters and password does not match. (16 Nov)**

import java.util.Scanner;

class InvalidUsername extends Exception {

public InvalidUsername(String message) {

super(message);

}

}

class PasswordMismatch extends Exception {

public PasswordMismatch(String message) {

super(message);

}

}

public class UserRegistration {

public static void validateUser(String username, String password, String confirmPassword) throws InvalidUsername, PasswordMismatch {

if (username.length() < 6) {

throw new InvalidUsername("Username must be at least 6 characters long!");

}

if (!password.equals(confirmPassword)) {

throw new PasswordMismatch("Password and confirmation password do not match!");

}

System.out.println("Registration successful!");

System.out.println("Username: " + username);

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

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

String username = scanner.nextLine();

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

String password = scanner.nextLine();

System.out.print("Confirm password: ");

String confirmPassword = scanner.nextLine();

validateUser(username, password, confirmPassword);

} catch (InvalidUsername usernameError) {

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

} catch (PasswordMismatch passwordError) {

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

} catch (Exception e) {

System.out.println("An error occurred: " + e.getMessage());

} finally {

scanner.close();

}

}

}

**Output:**

Enter username: ayush123

Enter password: 3456

Confirm password: 3456

Registration successful!

Username: ayush123

**26. Write a program to input name and age of a person and throw a user-defined exception, if the entered age is negative. (16 Nov)**

import java.util.Scanner;

class NegativeAgeException extends Exception {

public NegativeAgeException(String message) {

super(message);

}

}

public class PersonInfo {

public static void checkAge(int age) throws NegativeAgeException {

if (age < 0) {

throw new NegativeAgeException("Age cannot be negative!");

}

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

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

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

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

String name = scanner.nextLine();

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

int age = scanner.nextInt();

checkAge(age);

} catch (NegativeAgeException ageError) {

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

} catch (Exception e) {

System.out.println("Invalid input! Please enter valid data.");

} finally {

scanner.close();

}

}

}

**Output:**

Enter your name: Rahul

Enter your age: 37

Name: Rahul

Age: 37