

OBJECT ORIENTED PROGRAMMING (3ITRC2)

JAVA ASSIGNMENT

Submitted by

Santul Verma

23I3064

Information Technology A

Submitted to

Er. Aditya Makwe

Institute of Engineering and Technology

Devi Ahilya Vishwavidhyalaya, Indore (M.P.) India

Session July-Dec, 2024

ASSIGNMENT -1

24-11-2024

1. Write a Java program to change temperature from Celsius to Fahrenheit and vice versa.

```
import java.util.Scanner ;

class convertor{

    float f;

    float c ;

    float conf( float fa , float ce){

        f = fa ;

        c = ce;

        f = (c*(9/5))+32;

        return f;

    }

    float conc(float fa , float ce){

        f = fa ;

        c = ce;

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

        return c;

    }

}

public class tempConv {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        convertor c1 = new convertor();

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the temp in faranheit");

        float f1 = sc.nextFloat();

        System.out.println("Enter the temp in celcius");
```

```

float c2= sc.nextFloat();

System.out.println("The temp convertor is :");

c1.conc(f1,c2);

c1.conf(f1,c2);

}

}

```

24-11-2024

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

```

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.");

        }

    }

}

```

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

```
import java.util.Scanner;

public class maxOfThree {

    @SuppressWarnings("resource")
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter num1 :");

        int num1 = sc.nextInt();

        System.out.println("Enter num2 :");

        int num2 = sc.nextInt();

        System.out.println("Enter num3 :");

        int num3 = sc.nextInt();

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

            System.out.println("Number 1 is greater");

        }

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

            System.out.println("Number 2 is greater");

        }

        if((num3>num2)&&(num1<num3)){

            System.out.println("Number 3 is greater");

        }

    }

}
```

4. Write a Java program to swap two numbers.

```
import java.util.Scanner ;

public class swapping {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        int temp;

        Scanner sc = new Scanner(System.in) ;

        System.out.println("Enter num1 :");

        int num1 = sc.nextInt();

        System.out.println("Enter num2 :");

        int num2 = sc.nextInt();

        System.out.println("numbers before swapping :"+ num1 +"," +num2);

        temp = num1;

        num1 = num2 ;

        num2 = temp;

        System.out.println("Numbers after swapping :"+ num1+" ,"+num2);

    }

}
```

5. Write a Java program to convert miles to kilometres.

```
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);

}

}

```

24-11-2024

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

```

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");

        }

    }

}

```

24-11-2024

7. Write a Java program for following grading system.

Note:

Percentage \geq 90% : Grade A

Percentage \geq 80% : Grade B

Percentage \geq 70% : Grade C

Percentage \geq 60% : Grade D

Percentage \geq 40% : Grade E

Percentage $<$ 40% : Grade F

```
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 $\geq$ 90) System.out.println("Grade A");

        else if(p $\geq$ 80) System.out.println("Grade B");

        else if(p $\geq$ 70 ) System.out.println("Grade C");

        else if(p $\geq$ 60) System.out.println("Grade D");

        else if(p $\geq$ 50) System.out.println("Grade E");

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

    }

}
```

24-11-2024

8. Write a Java program to check whether a number is divisible by a number given by user.

```
import java.util.Scanner;
```

```

public class divisibility {

    @SuppressWarnings("resource")
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        System.out.println("Enter the number by which it is divided:");

        int n1 = sc.nextInt();

        if(n%n1==0){

            System.out.println(n+" is divisible by "+n1);

        }

    }

}

```

24-11-2024

9. Write a Java program to calculate factorial of 12.

```

class fact{

    int fa(int n ){

        if(n==1){

            return 1;

        }

        else{

            return n*fa(n-1);

        }

    }

}

public class factorial {

    public static void main(String[] args) {

```



```

        fact f = new fact();

        System.out.println("The factorial of 12 is:");

        System.out.println(f.fa(12));;

    }

}

```

24-11-2024

10. Write a Java program for Fibonacci series.

```

import java.util.Scanner;

class test{

    int fib(int n){

        if(n<=1){

            return n;

        }

        else{

            return fib(n-1)+fib(n-2);

        }

    }

}

public class fibonacciSeries {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        test f1 = new test();

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the num to print fibonacci series");

        int n = sc.nextInt();

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

            System.out.println(f1.fib(i));

        } } }

```

11. Write a Java program to reverse a number.

```
public class reverseNum{
    static int reverse(int n ){
        int rem ;
        int rev = 0;
        while( n >0){
            rem = n%10;
            rev = (rev*10)+rem;
            n = n/10;
        }
        return rev ;
    }
    public static void main(String[] args) {
        int n = 5432;
        System.out.println("the reversed number is :"+reverse(n));
    }
}
```

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) students, write a program to process the applications to list the eligible candidates.

```

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();
    }
}

```

24-11-2024

3. Write a Java program to calculate the sum of natural numbers up to a certain range.

```

import java.util.Scanner;

public class sumOfnumber {

    @SuppressWarnings("resource")
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the range of natural number:");
        int n = sc.nextInt();
        int numb = (n*(n+1))/2;
        System.out.println("The sum of natural numbers is :"+numb);
    }
}

```

24-11-2024

14. Write a Java program to print all multiple of 10 between a given interval.

```
import java.util.Scanner;

public class mulitple10 {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the range of integers");

        int n= sc.nextInt();

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

            i = i*10;

            System.out.println(i);

        }

    }

}
```

24-11-2024

15. Write a Java program to generate multiplication table.

```
import java.util.Scanner;

public class table {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        System.out.println("The table of "+ n + " is");

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

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

        }

    }

}
```

16. Write a Java program to find HCF of two Numbers.

```
import java.util.Scanner;

public class hcf {

    static int HCF(int a , int b){

        while(b!=0){

            int temp = b;

            b = a%b;

            a = temp;}

        return a ;

    }

    @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 HCF of two numbers is :");

        System.out.println(HCF(a1,b1));

    }

}
```

17. Write a Java program to find LCM of two Numbers.

```
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)));

} }

```

24-11-2024

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

```

import java.util.*;

public class digitCount {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        int count =0 ;
    }
}

```

```

Scanner sc = new Scanner(System.in);
System.out.println("Enter the number :");
int n = sc.nextInt();
while(n>0){
    n = n/10;
    count = count+1;
}
System.out.println("The number of digits is :"+count);

}
}

```

24-11-2024

19. Write a Java program to check whether a number is palindrome or not.

```

import java.util.Scanner;

public class palindrome {

    @SuppressWarnings("resource")
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number :");
        int n = sc.nextInt();
        int temp =n;
        int rem ;
        int rev = 0;
        while( n >0){
            rem = n%10;
            rev = (rev*10)+rem;
            n = n/10;

```



```

    }

    //System.out.println(rev);

    if(temp == rev){

        System.out.println("The given number is palindrome");

    }

    else{

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

    }

}

}

```

24-11-2024

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

```

import java.util.*;

public class primenumber {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        int count =0;

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

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

            if(n%i==0){

                count++;

            }

        }

    }

}

```

```

    if(count==2){
        System.out.println("It is prime");
    }
    else{
        System.out.println("It is not prime");
    }
}
}
}

```

24-11-2024

21. Write a Java program to convert a Binary Number to Decimal and Decimal to Binary.

```

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);
}
}

```

24-11-2024

22. Write a Java program to find median of a set of numbers.

```

import java.util.Arrays;

public class median {
    public static void main(String[] args) {

```

```

int Median1 =0 ;
int Median2 = 0;
int m1[] ={44,31,6};
int m2[] = {12,33,2,11,45,3};
Arrays.sort(m1);
Arrays.sort(m2);
    if((m1.length)%2!=0){
        Median1 = m1[Math.floorDiv(m1.length,2)];
    }
    System.out.println("The median is :"+Median1);
    if((m2.length)%2==0){
        Median2 = m2[(Math.floorDiv(m2.length,2)+(Math.floorDiv(m2.length,2)+1))/2];
    }
    System.out.println("The median is :"+Median2);
    }
}

```

24-11-2024

23. Write Java programs for the patterns given below:

(a) 1

2 3 4

5 6 7 8 9

(b) 1

2 1 2

3 2 1 2 3

4 3 2 1 2 3 4

```

public class pattern {

    public static void main(String[] args) {

        /*
        1
        2 3 4
        5 6 7 8 9 8*/
        int c = 1;
        for(int i = 1; i<=3;i++){
            for( int j = 1; j<=(2*i-1);j++)
            {
                System.out.print(c + " ");
                c++;
            }
            System.out.println();
        }
        /*
        1
        2 1 2
        3 2 1 2 3
        4 3 2 1 2 3 4
        */
        for(int i = 1;i<=4;i++){
            for( int j = 1;j<=i-1;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();
}

```

24-11-2024

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

```

public class sum_AvgArr{
    public static void main(String[] args) {
        int arr[] = { 1,2,3,9,5};
        int sum = 0;
        int size = arr.length;
        for(int i =0;i<5;i++){
            sum = sum+arr[i];
        }
        System.out.println("The sum of elements of array is :"+sum);
        int avg = sum/size;
        System.out.println("The average of array is :"+avg);
    }
}

```

24-11-2024

25. Write a Java program to implement stack using array.

```

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();

}

}

```

24-11-2024

26. Write a Java program to implement Queue using array.

```

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();

}
}

```

27. Write a Java program to enter n elements in an array and find smallest number among them.

```
public class smallestArray {  
    public static void main(String[] args) {  
        int arr[]={11,37,44,56,11,2};  
        int min = arr[0];  
        for( int i =0;i<6;i++){  
            if(min>arr[i]){  
                min = arr[i];  
            }  
        }  
        System.out.println("The smallest element in array is :"+ min);  
    }  
}
```

28. Write Java program to find the sum of all odd numbers in a array.

```
public class SumOfOdd {  
    public static void main(String[] args) {  
        int arr[]={1,23,3,4,11,15};  
        int sum =0 ;  
        for(int i =0;i<6;i++){  
            if(arr[i]%2!=0){  
                sum = sum+arr[i];  
            }  
        }  
    }  
}
```

```

        System.out.println("The sum of all odd numbers in a given array is :"+sum);
    }
}

```

24-11-2024

29. Write a Java program to find duplicate elements in a 1D array and find their frequency of occurrence.

```

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);
    }
}
}
}

```

24-11-2024

30. Write a Java program to print every alternate number of a given array

```

public class alternateNumber {
    public static void main(String[] args) {
        int arr[] = new int[]{12,33,89,67};
        for(int i =0;i<4;i++){
            if(i%2==0){
                System.out.print(arr[i]+" ");
            }
        }
    }
}

```

24-11-2024

31. Write a Java program to show 0-arguments constructor.

```

class Abc{
    Abc(){
        System.out.println("0-Arguments constructor");
    }
}

```

```

public class zeroAurguments {
    public static void main(String[] args) {
        new Abc();
    }
}

```

24-11-2024

32. Write a Java program to show parameterized constructor.

```

class MM{
    String name;
    int age;
    MM(String name , int age){
        this.name = name ;
        this.age = age;
    }
}

```

```

public class parameterizedConstructor {
    public static void main(String[] args) {
        MM a1 = new MM("Santul", 19);
        System.out.println("The name of student is :"+a1.name);
        System.out.println("The age is :"+a1.age);
    }
}

```

24-11-2024

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"

```
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());
    }
}
```

ASSIGNMENT -2

24-11-2024

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.

```
import java.util.Arrays;

public class sortedArray {

    public static void main(String[] args) {

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

        int b[] = { 6,7,8,9};

        int l1 = a.length;

        int l2 = b.length;

        int c[] = new int[l1+l2];

        int i = 0 ;int k = 0;

        int j = 0;

        while(i<l1){

            c[k++]=a[i++];

        }

        while(j<l2){

            c[k++]=b[j++];

        }

        Arrays.sort(c);

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

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

        }

    }

}
```

2. Write a Java program to show 0-arguments constructor.

```
class Abc{
    Abc(){
        System.out.println("0-Arguments constructor");
    }
}

public class zeroAurguments {

    public static void main(String[] args) {
        new Abc();
    }
}
```

3. Write a Java program to show parameterized constructor.

```
class MM{
    String name;
    int age;
    MM(String name , int age){
        this.name = name ;
        this.age = age;
    }
}

public class parameterizedConstructor {
    public static void main(String[] args) {
```



```

        MM a1 = new MM("Santul", 19);

        System.out.println("The name of student is :"+a1.name);

        System.out.println("The age is :"+a1.age);

    }
}

```

24-11-2024

4. Write a Java program to show constructor overloading.

```

class Add{
    Add(int a){
        a = a+10;

        System.out.println("The value of a is :"+ a);
    }

    Add(int a , int b , int c){
        a = a*b;

        c = a+134;

        b = a%c;

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

        System.out.println("The value of b is :"+b);

        System.out.println("The value of c is :"+ c);
    }
}

public class constructorOverloading {
    public static void main(String[] args) {
        Add x = new Add(12);

        Add y = new Add(13);

        Add z = new Add(122, 11, 5);
    }
}

```

```
}
```

```
}
```

24-11-2024

5. Write a Java program to implement the concept of inheritance.

```
class Animal{  
    void eat(){  
        System.out.println("eating...");  
    }  
}  
  
class Dog extends Animal{  
    void bark(){  
        System.out.println("barking...");  
    }  
}
```

```
public class singleInheritance {  
    public static void main(String[] args) {  
        Dog d = new Dog();  
        d.bark();  
        d.eat();  
    }  
}
```

24-11-2024

6. Write a Java program to show method overloading.

```

public class methodOverloading {

    static int Area(int a ){

        return a*a ;

    }

    static int Area(int a , int b){

        return a* b;

    }


    public static void main(String[] args) {

        System.out.println("The area of Square is :"+Area(25));

        System.out.println("The area of Rectangle is :"+ Area(12, 11));

    }

}

```

24-11-2024

7. Write a Java program to show method overriding.

```

//example 1

class Vehicle{

    public void k(){

        System.out.println("vehicle is running safely... ");

    }

}

class Bike extends Vehicle{

    public void k(){

        System.out.println("Bike is running safely...");

    }

}

```

```

}

// example 2

class Bank{

    int rateOfInterest(){

        return 0;

    }

}

class SBI extends Bank{

    int rateOfInterest(){

        return 8;

    }

}

class Axis extends Bank{

    int rateOfInterest(){

        return 9;

    }

}

class ICICI extends Bank{

    int rateOfInterest(){

        return 7;

    }

}

public class methodOverriding {

    public static void main(String[] args) {

        Bike ob = new Bike();

        ob.k();

        SBI s = new SBI();

        Axis a = new Axis();

        ICICI i = new ICICI();

```

```
        System.out.println(s.rateOfInterest());  
        System.out.println(a.rateOfInterest());  
        System.out.println(i.rateOfInterest());  
    }  
}
```

24-11-2024

8. Write a Java program to show method hiding.

```
class Human{  
    public static void Smile(){  
        System.out.println("The human smile's glows up the face of him");  
    }  
}  
  
class Baby extends Human{  
    public static void Smile(String msg){  
        System.out.println("The human greets:"+msg);  
    }  
}  
  
public class methodHiding {  
    public static void main(String[] args) {  
        Human h = new Human();  
        h.Smile();  
        Baby b = new Baby();  
        b.Smile();  
        b.Smile("Hello with a smile");  
    }  
}
```

```
}
```

```
}
```

24-11-2024

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.

```
import java.util.Scanner;
```

```
abstract class ThreeDObject{  
    abstract double surfaceArea();  
    abstract double volume();  
}
```

```
class Box extends ThreeDObject{  
    private double length, width, height;  
  
    public Box(double length, double width, double height) {  
        this.length = length;  
        this.width = width;  
        this.height = height;  
    }  
  
    public double surfaceArea() {  
        return 2 * (length * width + width * height + height * length);  
    }  
}
```

```

    }

    public double volume() {
        return length * width * height;
    }
}

class Cube extends Box{
    public Cube(double side) {
        super(side, side, side);
    }
}

class Cylinder extends ThreeDObject {
    private double radius, height;

    public Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    public double surfaceArea() {
        return 2 * Math.PI * radius * (radius + height);
    }

    public double volume() {
        return Math.PI * radius * radius * height;
    }
}

class Cone extends ThreeDObject {

```

```

private double radius, height;

public Cone(double radius, double height) {
    this.radius = radius;
    this.height = height;
}

public double surfaceArea() {
    return Math.PI * radius * (radius + Math.sqrt(height * height + radius * radius));
}

public double volume() {
    return (1.0 / 3.0) * Math.PI * radius * radius * height;
}
}

public class menstruation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter length, width, and height of the box: ");
        double length = scanner.nextDouble();
        double width = scanner.nextDouble();
        double height = scanner.nextDouble();
        Box box = new Box(length, width, height);

        System.out.print("Enter side of the cube: ");
        double side = scanner.nextDouble();
        Cube cube = new Cube(side);

        System.out.print("Enter radius and height of the cylinder: ");

```



```
double radius = scanner.nextDouble();
height = scanner.nextDouble();
Cylinder cylinder = new Cylinder(radius, height);

System.out.print("Enter radius and height of the cone: ");
radius = scanner.nextDouble();
height = scanner.nextDouble();
Cone cone = new Cone(radius, height);

System.out.println("Box Surface Area: " + box.surfaceArea());
System.out.println("Box Volume: " + box.volume());

System.out.println("Cube Surface Area: " + cube.surfaceArea());
System.out.println("Cube Volume: " + cube.volume());

System.out.println("Cylinder Surface Area: " + cylinder.surfaceArea());
System.out.println("Cylinder Volume: " + cylinder.volume());

System.out.println("Cone Surface Area: " + cone.surfaceArea());
System.out.println("Cone Volume: " + cone.volume());

scanner.close();
}
}
```

24-11-2024

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.

```
class Vehicle{
    protected int regnNumber;
    protected int speed;
    protected String color ;
    protected String ownerName;
    public Vehicle(int regnNumber,int speed,String color,String ownerName){
        this.color = color;
        this.speed = speed;
        this.ownerName = ownerName;
        this.regnNumber = regnNumber;
    }
    public void showData(){
        System.out.println("This is vehicle class:");
    }
}

class Bus extends Vehicle{
    private int routeNumber;
    public Bus(int regnNumber,int s,String colour,String ownerName,int routeNumber){
        super(regnNumber, s, colour, ownerName);
        this.routeNumber = routeNumber;
    }
}
```

```

    }

    public void showData(){
        super.showData();
        System.out.println("This is routenumber for bus:"+routeNumber);
        System.out.println("Registration Number:"+regnNumber);
        System.out.println("speed"+speed);
        System.out.println("color:"+color);
        System.out.println("ownerName:"+ownerName);
    }

}

class Car extends Vehicle{
    private String manufacturerName;

    public Car(String manufacturerName,int regnNumber,int s,String colour,String
ownerName){
        super(regnNumber, s, colour, ownerName);
        this.manufacturerName = manufacturerName;

    }

    public void showData(){
        super.showData();
        System.out.println("Manufacturer Name :"+manufacturerName);
        System.out.println("Registration Number:"+regnNumber);
        System.out.println("speed"+speed);
        System.out.println("color:"+color);
        System.out.println("ownerName:"+ownerName);
    }
}

```

```

}

public class Main {

    public static void main(String[] args) {

        Bus B = new Bus(8871, 100, "Black", "Raju", 120);

        B.showData();

        Car C = new Car("Maruti Motors", 7623, 70, "White", "Nayanjyoti Sharma");

        C.showData();

    }

}

```

24-11-2024

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.

```

class Num{
    int number;

    public Num(int number){
        this.number = number;
    }

    void showNum(){
        System.out.println("The number is "+ number);
    }
}

class HexNum extends Num{

    public HexNum(int number){

```

```

        super(number);
    }

    String num = Integer.toHexString(number);
    String num1 = Integer.toOctalString(number);

    void showNum(){
        super.showNum();
        System.out.println("The Hexnumber is "+ num);
        System.out.println("The Octnumber is "+ num1);

    }
}

public class HexandOct {
    public static void main(String[] args) {
        HexNum hexNum = new HexNum(12);
        hexNum.showNum();}
}

```

24-11-2024

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.

```

import java.util.Scanner;

class Distance{
    double distance;

    int speed = 60;

    void travelTime( double distance ){

```

```

        this.distance = distance;

        double time = distance/speed;

        System.out.println("The time taken in hours is:"+time);
    }
}

```

```

class DistanceMKS extends Distance{

    double distance ;

    int speed = 100;

    void travelTime(double distance){

        this.distance = distance;

        double distanceKM = distance*1.609;

        double time = distanceKM/speed;

        System.out.println("The time in seconds is :"+time);

    }

}

```

```

public class Ass212 {

    @SuppressWarnings("resource")

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        double n = sc.nextDouble();

        DistanceMKS d = new DistanceMKS();

        d.travelTime(n);

    }

}

```

24-11-2024

13. Write a Java program to explain “multilevel inheritance.”

```

class Animal{
    void eat(){
        System.out.println("eating...");
    }
}

class Dog extends Animal{
    void bark(){
        System.out.println("barking...");
    }
}

public class singleInheritance {
    public static void main(String[] args) {
        Dog d = new Dog();
        d.bark();
        d.eat();
    }
}

```

24-11-2024

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.

```

import java.util.Scanner;

public class Employee {
    String emp_id, emp_name ;
    double gross_salary, basic_salary;
    double hra ;
}

```

```
void employeeDetails(){
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the employee id :");
    String emp_id = sc.nextLine();
    System.out.println("Enter the employee name:");
    String emp_name = sc.nextLine();
    System.out.println("Enter the basic salary:");
    double basic_salary = sc.nextDouble();

}
```

```
void displayGross_Salary(){
    hra = 0.12*basic_salary;
    gross_salary = basic_salary + hra;
    System.out.println("Gross_Salary :"+gross_salary);
}
```

```
public static void main(String[] args) {
    Employee e = new Employee();
    e.employeeDetails();
    e.displayGross_Salary();
}
}
```

24-11-2024

15. Write a program to demonstrate use of 'this' keyword.

```
class Made{
```



```

String cookingTechnique;

String Item;

String mainIngredient;

public Made(String cookingTechnique,String Item,String mainIngredient){

    this.cookingTechnique = cookingTechnique;

    this.Item = Item;

    this.mainIngredient = mainIngredient;

}

void Recipe(){

    System.out.println("THE RECIPE OF TODAY IS:");

    System.out.println("The cooking technique is :"+ cookingTechnique);

    System.out.println("The dish is:"+Item);

    System.out.println("The main ingredient is:"+mainIngredient);

}

}

public class This {

    public static void main(String[] args) {

        Made M = new Made("Baking", "Cake", "Flour");

        M.Recipe();

    }

}

```

24-11-2024

16. Write a program to demonstrate use of 'static' keyword.

```

public class Static {

```

```

static int var = 10;

static void weep(){
    System.out.println("Weeping ");
}

public static void main(String[] args) {

    weep();

    System.out.println("The value of var is :"+var);
}
}

```

24-11-2024

17. Write program, which finds the sum of numbers formed by consecutive digits.

Input : 2415

output : 24+41+15=80.

```

import java.util.Scanner;

public class Ass217{
    @SuppressWarnings("resource")
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number :");
        int n = sc.nextInt();
        int sum =0 ;
        int rem;
        while(n>=10){
            rem = n%100;
            sum = sum+rem;

```

```

        n = n/10;
    }
    System.out.println("The sum of consecutive digits is :"+sum);
}
}

```

24-11-2024

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.

```

interface A{
    void smile();
    void run();
}

```

```

interface B{
    void cry();
    void eat();
}

```

```

interface C{
    void sleep();
    void walk();
}

```

```

interface D extends A,B,C{
    void fun();
}

```

```
}
```

```
class Base implements A,B,C{
```

```
    public void smile(){
```

```
        System.out.println("this makes me smile!");
```

```
    }
```

```
    public void run(){
```

```
        System.out.println("She is crying badly");
```

```
    }
```

```
    public void eat(){
```

```
        System.out.println("this is super delicious");
```

```
    }
```

```
    public void sleep(){
```

```
        System.out.println("she is sleeping");
```

```
    }
```

```
    public void walk(){
```

```
        System.out.println("she is walking in garden");
```

```
    }
```

```
    public void cry(){
```

```
        System.out.println("This feeling makes her cry");
```

```
    }
```

```
}
```

```
class Heirs extends Base implements D{
```

```
    public void fun(){
```

```
        System.out.println("she is fun activity");
```

```

    }

    public void A1(A a1){
        a1.run();
        a1.smile();

    }

    public void A2( B b1){
        b1.cry();
        b1.eat();
    }

    public void A3( C c1){
        c1.sleep();
        c1.walk();
    }

    public void A4( D d1){
        d1.cry();
        d1.eat();
        d1.fun();
        d1.run();
        d1.sleep();
        d1.smile();
        d1.walk();
    }

}

public class Ass218 {
    public static void main(String[] args) {
        Heirs H = new Heirs();
        H.A1(H);
    }
}

```

```

        H.A2(H);
        H.A3(H);
        H.A4(H);

    }
}

```

24-11-2024

19. Write a Java program to show the use of all keywords for exception handling.

```

import java.lang.Exception;

class NegativeNumberException extends Exception {
    public NegativeNumberException(String message) {
        super(message);
    }
}

public class exceptionHandling {

    public static void checkNumber(int number) throws NegativeNumberException{
        if (number < 0) {
            throw new NegativeNumberException("The number " + number + " is negative.");
        }
    }

    public static void main(String[] args) {
        int[] numbers = {10, -5, 3, -1, 7, 8};

        for (int number : numbers) {
            try {

```

```

        checkNumber(number);
    } catch (NegativeNumberException e) {
        System.err.println("Caught NegativeNumberException: " + e.getMessage());
    } catch (Exception e) {
        System.err.println("Caught Exception: " + e.getMessage());
    } finally{
        System.out.println("It will always work");
    }
}
}
}

```

24-11-2024

20. Write a Java program using try and catch to generate NegativeArrayIndex Exception and Arithmetic Exception.

```

public class twoExceptions {
    public static void main(String[] args) {
        try {
            int[] array = {-1, 2, 3};
            int result = array[-2] / 0; // Both exceptions will be thrown
        } catch (NegativeArraySizeException e) {
            System.out.println("NegativeArraySizeException caught: " + e.getMessage());
        } catch (ArithmeticException e) {
            System.out.println("ArithmeticException caught: " + e.getMessage());
        }
    }
}
}

```

24-11-2024

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.

```
import java.util.HashMap;

import java.util.Map;

public class 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];

        Map<String, String> capitals = new HashMap<>();

        capitals.put("India", "New Delhi");

        capitals.put("USA", "Washington D.C.");

        capitals.put("France", "Paris");

        try {

            String capital = capitals.get(country);

            if (capital == null) {

                throw new NoMatchFoundException("No capital found for the country: " +

country);

            }

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

        } catch (NoMatchFoundException e) {

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

        }

    }

}
```



```

    }

    static class NoMatchFoundException extends Exception {
        public NoMatchFoundException(String message) {
            super(message);
        }
    }
}

```

24-11-2024

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

```

import java.lang.Exception;

class MyMathException extends Exception{
    public MyMathException(String message){
        super(message);
    }
}

```

```

public class MathException {
    public static void main(String[] args) {
        int num1 = 10, num2 = 0;

        try {
            // Division by zero
            int result = num1 / num2;
            System.out.println("Result: " + result);
        }
    }
}

```

```

        // Square root of negative number
        double sqrt = Math.sqrt(-1);
        System.out.println("Square root: " + sqrt);
    } catch (ArithmeticException e) {
        if (e.getMessage().equals("/ by zero")) {
            System.out.println("Cannot divide by zero!");
        } else {
            System.out.println("Invalid input: Cannot calculate square root of negative
number.");
        }
    } catch (Exception e) {
        System.out.println("An unexpected error occurred: " + e.getMessage());
    }
}
}

```

24-11-2024

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.

```

public class TemperatureCheck {
    public static void main(String[] args) {
        if (args.length == 0) {

```

```
        System.out.println("Please provide a temperature in Celsius as a command-line argument.");
```

```
        return;
```

```
    }
```

```
    try {
```

```
        double temperature = Double.parseDouble(args[0]);
```

```
        if (temperature > 35) {
```

```
            throw new TooHotException("Temperature is too hot!");
```

```
        } else if (temperature < 5) {
```

```
            throw new TooColdException("Temperature is too cold!");
```

```
        } else {
```

```
            double fahrenheit = (temperature * 9/5) + 32;
```

```
            System.out.println("Normal temperature. Fahrenheit: " + fahrenheit);
```

```
        }
```

```
    } catch (NumberFormatException e) {
```

```
        System.out.println("Invalid input. Please enter a valid number.");
```

```
    } catch (TooHotException | TooColdException e) {
```

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

```
    }
```

```
}
```

```
static class TooHotException extends Exception {
```

```
    public TooHotException(String message) {
```

```
        super(message);
```

```
    }
```

```
}
```

```
static class TooColdException extends Exception {
```

```

    public TooColdException(String message) {
        super(message);
    }
}

```

24-11-2024

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.

```

import java.util.Scanner;

class TooOlderException extends Exception {
    public TooOlderException(String message) {
        super(message);
    }
}

class TooYoungerException extends Exception {
    public TooYoungerException(String message) {
        super(message);
    }
}

```

```

public class ageException {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter candidate's name: ");
        String name = scanner.nextLine();

        System.out.print("Enter candidate's age: ");
        int age = scanner.nextInt();

        try {
            if (age > 45) {
                throw new TooOlderException("Candidate is too older.");
            } else if (age < 20) {
                throw new TooYoungerException("Candidate is too younger.");
            } else {
                System.out.println("Eligible: " + name);
            }
        } catch (TooOlderException | TooYoungerException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

24-11-2024

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

```

import java.util.Scanner;

```

```
class InvalidUsernameException extends Exception {  
    public InvalidUsernameException(String message) {  
        super(message);  
    }  
}
```

```
class InvalidPasswordException extends Exception {  
    public InvalidPasswordException(String message) {  
        super(message);  
    }  
}
```

```
public class UserValidation  
{  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter username: ");  
        String username = scanner.nextLine();  
  
        System.out.print("Enterpassword: ");  
        String password = scanner.nextLine();  
  
        try {  
            if (username.length() < 6) {  
                throw new InvalidUsernameException("Username must be at least 6 characters  
long.");  
            }  
        }  
    }  
}
```

```

        if (!password.equals("password123")) {
            throw new InvalidPasswordException("Invalid password.");
        }
    } catch (InvalidUsernameException | InvalidPasswordException e) {
        System.out.println(e.getMessage());
    }
}
}
}

```

24-11-2024

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

```

import java.util.Scanner;

class NegativeAgeException extends Exception {
    public NegativeAgeException(String message) {
        super(message);
    }
}

public class AgeValidation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter your name: ");
        String name = scanner.nextLine();

        System.out.print("Enter your age: ");
        int age = scanner.nextInt();
    }
}

```

```
try {  
    if (age < 0) {  
        throw new NegativeAgeException("Age cannot be negative.");  
    } else {  
        System.out.println("Name: " + name);  
        System.out.println("Age: " + age);  
    }  
} catch (NegativeAgeException e) {  
    System.out.println(e.getMessage());  
  
}  
  
}  
  
}
```