1. WAP to add two no.

import java.util.Scanner;

public class Addition {

public static void main(String[]args)

{

int a,b,sum;

Scanner scan=new Scanner(System.in);

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

a=scan.nextInt();

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

b=scan.nextInt();

scan.close();

sum=a+b;

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

}

}

2. WAP to print table of any no.

import java.util.Scanner;

public class table{

public static void main(String[] args){

 Scanner scn=new Scanner(System.in);

int n;

int i=0;

System.out.println("Enter the number you want a table of:");

n=scn.nextInt();

scn.close();

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

{

int m=n\*i;

System.out.println(n+"X"+i+"="+m);

}

}

}

3. WAP to change int value to byte.

import java.util.Scanner;

public class Integer\_Conversion

{

   public static void main(String[] args)

    {

        int a;

        byte b;

        Scanner s = new Scanner(System.in);

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

        a = s.nextInt();

        b = (byte) a;

        System.out.println("Conversion into byte:"+b);

    }

}

4. WAP to perform any two arithmetic operations using typecasting.

class addn\_typecast

{

public static void main(String[] args)

{

int i=51;

char c ='C';

int r=(int)c;

System.out.println("Addition :"+(i+r));

System.out.println("product :"+(i\*r));

}

}

5. WAP to find, the no is odd or even.

import java.util.Scanner;

public class Sort\_no

{

public static void main(String[] args)

{

int num;

Scanner sc=new Scanner(System.in);

System.out.print("ENTER ANY NUMBER=====> ");

num=sc.nextInt();

sc.close();

if (num%2==0)

{System.out.println("THE NUMBER YOU ENTERED IS EVEN");

}

else

{System.out.println("THE NUMBER YOU ENTERED IS ODD");

}

}

}

6. WAP to find MAX no using ternary operator.

class GFG

{

public static void main(String args[])

{

    // variable declaration

    int n1 = 5, n2 = 10, max;

    // Largest among n1 and n2

    max = (n1 > n2) ? n1 : n2;

    // Print the largest number

    System.out.println("Largest number between " + n1 +

                  " and " + n2 + " is " + max + ". " );

}

}

7. WAP of the following Operators:-

i).Arithmetic Operator (+,-,\*,/,%)

import java.util.Scanner;

public class Addition {

public static void main(String[]args)

{

int a,b,sum;

Scanner scan=new Scanner(System.in);

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

a=scan.nextInt();

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

b=scan.nextInt();

scan.close();

sum=a+b;

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

b)

import java.util.Scanner;

public class Subtraction{

public static void main(String[]args)

{

int a,b,sub;

Scanner scan=new Scanner(System.in);

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

a=scan.nextInt();

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

b=scan.nextInt();

scan.close();

sub=a-b;

System.out.println("the minus of the two numbers is: "+sub);

c)

import java.util.Scanner;

public class product{

public static void main(String[]args)

{

int a,b,prod;

Scanner scan=new Scanner(System.in);

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

a=scan.nextInt();

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

b=scan.nextInt();

scan.close();

prod=a\*b;

System.out.println("the product of the two numbers is: "+prod);

ii).Logical Operator (&&,||,!)

iii).Relational Operator (==,!=,>,<,>=,<=)

a)

class Relational\_1 {

    public static void main(String[] args)

    {

       int a= 5, b = 10, c = 5;

        System.out.println("no.1 = " +a);

        System.out.println("no.2 = " + b);

        System.out.println("no.3 = " + c);

        System.out.println("var1 == var2: "

                           + (var1 == var2));

        System.out.println("var1 == var3: "

                           + (var1 == var3));

    }

}

b)

class Relational\_2 {

    public static void main(String[] args)

    {

        int var1 = 15, var2 = 10, var3 = 5;

        System.out.println("Var1 = " + var1);

        System.out.println("Var2 = " + var2);

        System.out.println("Var3 = " + var3);

        System.out.println("var1 != var2: "

                           + (var1 != var2));

        System.out.println("var1 != var3: "

                           + (var1 != var3));

    }

}

c)

class Relational\_3 {

    public static void main(String[] args)

    {

               int var1 = 10, var2 = 20, var3 = 5;

        System.out.println("Var1 = " + var1);

        System.out.println("Var2 = " + var2);

        System.out.println("Var3 = " + var3);

        System.out.println("var1 < var2: "

                           + (var1 < var2));

               System.out.println("var2 < var3: "

                           + (var2 < var3));

    }

}

d)

class Relational\_4 {

    public static void main(String[] args)

    {

        int var1 = 20, var2 = 20, var3 = 10;

        System.out.println("Var1 = " + var1);

        System.out.println("Var2 = " + var2);

        System.out.println("Var3 = " + var3);

        System.out.println("var1 >= var2: "

                           + (var1 >= var2));

        System.out.println("var2 >= var3: "

                           + (var3 >= var1));

    }

}

e)

class Relational\_5 {

    public static void main(String[] args)

    {

        int var1 = 10, var2 = 10, var3 = 9;

        System.out.println("Var1 = " + var1);

        System.out.println("Var2 = " + var2);

        System.out.println("Var3 = " + var3);

        System.out.println("var1 <= var2: "

                           + (var1 <= var2));

        System.out.println("var2 <= var3: "

                           + (var2 <= var3));

    }

}

f)

class Relational\_6 {

    public static void main(String[] args)

    {

        int var1 = 10, var2 = 10, var3 = 9;

        System.out.println("Var1 = " + var1);

        System.out.println("Var2 = " + var2);

        System.out.println("Var3 = " + var3);

        System.out.println("var1 < var2: "

                           + (var1 < var2));

        System.out.println("var2 < var3: "

                           + (var2 < var3));

    }

}

iv).Increment/Decrement Operator (++,--)

a)

public class IncrementDecrement

{

   public static void main(String[] args)

   {

      int number = 50;

      System.out.println("Number is " + number);

      number++;

      System.out.println("Now, number is " + number);

      number--;

      System.out.println("Now, number is " + number);

   }

}

v).Bitwise Operator (~,&,|,^,>>,<<,>>>)

a)class Main {

  public static void main(String[] args) {

    int number1 = 12, number2 = 25, result;

    result = number1 | number2;

    System.out.println(result);

 result = number1 & number2;

    System.out.println(result);

 result = number1 ^ number2;

    System.out.println(result);

  }

}

b)

class Main {

  public static void main(String[] args) {

    int number = 35, result;

    result = ~number;

    System.out.println(result);

  }

}

c)

class Main {

  public static void main(String[] args) {

    int number = 2;

     int result = number << 2;

    System.out.println(result);

  }

}

d)

class Main {

  public static void main(String[] args) {

    int number1 = 8;

    int number2 = -8;

    System.out.println(number1 >> 2);

    System.out.println(number2 >> 2);

  }

}

e)

class Main {

  public static void main(String[] args) {

    int number1 = 8;

    int number2 = -8;

    System.out.println(number1 >>> 2);

    System.out.println(number2 >>> 2);

  }

}

vi).Assignment Operator (=,+=,-=,\*=,/=,%=,&=,|=,^=,>>=,<<=,>>>=)

a)

class Assgn{

public static void main(String args[]){

int a=100;

int b=200;

a+=4;

b-=4;

System.out.println(a);

System.out.println(b);

a\*=2;

b/=2;

System.out.println(a);

System.out.println(b);

a%=2;

b|=2;

System.out.println(a);

System.out.println(b);

a^=2;

b>>=2;

System.out.println(a);

System.out.println(b);

a<<=2;

b>>>=2;

System.out.println(a);

System.out.println(b);

}

}

vii).Ternary Operator (?:)

Class Tern{

public static void main(String args[]){

int a=10;

int b=5;

int max=(a>b)?a:b;

System.out.println(max);

}

}

8. WAP to perform arithmetic calculation using switch.

a)

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner reader = new Scanner(System.in);

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

    int first = reader.nextInt();

    int second = reader.nextInt();

    reader.close();

    int result;

    switch (addition) {

      case 1:

        result = first + second;

        break;

default:

        System.out.println("Error! operator is not correct");

        return;

    }

    System.out.println(first + " " + operator + " " + second + " = " + result);

  }

}

9. WAP to demonstrate all three iterators(for,while and do while).

a)

import java.util.Scanner;

public class table{

public static void main(String[] args){

 Scanner scn=new Scanner(System.in);

int n;

int i=0;

System.out.println("Enter the number you want a table of:");

n=scn.nextInt();

scn.close();

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

{

int m=n\*i;

System.out.println(n+"X"+i+"="+m);

}

}

}

b)

class Main {

  public static void main(String[] args) {

        int i = 1, n = 5;

       while(i <= n) {

      System.out.println(i);

      i++;

    }

  }

}

10. WAP of Linear search.

a)

import java.util.Scanner;

class LinearSearch

{

   public static void main(String args[])

   {

      int counter, num, item, array[];

      Scanner input = new Scanner(System.in);

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

      num = input.nextInt();

      array = new int[num];

      System.out.println("Enter " + num + " integers");

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

        array[counter] = input.nextInt();

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

      item = input.nextInt();

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

      {

         if (array[counter] == item)

         {

           System.out.println(item+" is present at location "+(counter+1));

           break;

         }

      }

      if (counter == num)

        System.out.println(item + " doesn't exist in array.");

   }

}

11. WAP of Binary search.

a)

class BinarySearch{

 public static void binarySearch(int arr[], int first, int last, int key){

   int mid = (first + last)/2;

   while( first <= last ){

      if ( arr[mid] < key ){

        first = mid + 1;

      }else if ( arr[mid] == key ){

        System.out.println("Number found at location: " + mid);

        break;

      }else{

         last = mid - 1;

      }

      mid = (first + last)/2;

   }

   if ( first > last ){

      System.out.println("Element is not found!");

   }

 }

 public static void main(String args[]){

        int arr[] = {10,20,30,40,50};

        int key = 30;

        int last=arr.length-1;

        binarySearch(arr,0,last,key);

 }

}

12. Wap to find independent vectors in the given matrix.

a)

class Matrix\_rank {

    static final int R = 3;

    static final int C = 3;

    static void swap(int mat[][],

          int row1, int row2, int col)

    {

        for (int i = 0; i < col; i++)

        {

            int temp = mat[row1][i];

            mat[row1][i] = mat[row2][i];

            mat[row2][i] = temp;

        }

    }

    static void display(int mat[][],

                     int row, int col)

    {

        for (int i = 0; i < row; i++)

        {

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

                System.out.print(" "

                          + mat[i][j]);

            System.out.print("\n");

        }

    }

    static int rankOfMatrix(int mat[][])

    {

        int rank = C;

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

        {

            if (mat[row][row] != 0)

            {

                for (int col = 0; col < R; col++)

                {

                    if (col != row)

                    {

                        double mult =

                           (double)mat[col][row] /

                                    mat[row][row];

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

                            mat[col][i] -= mult

                                       \* mat[row][i];

                    }

                }

            }

            else

            {

                boolean reduce = true;

                for (int i = row + 1; i < R; i++)

                {

                    if (mat[i][row] != 0)

                    {

                        swap(mat, row, i, rank);

                        reduce = false;

                        break ;

                    }

                }

                if (reduce)

                {

                    rank--;

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

                        mat[i][row] = mat[i][rank];

                }

                row--;

            }

        }

        return rank;

    }

    public static void main (String[] args)

    {

        int mat[][] = {{10, 20, 10},

                       {-20, -30, 10},

                       {30, 50, 0}};

        System.out.print("Rank of the matrix is : "

                               + rankOfMatrix(mat));

    }

}

13. Wap to find the rank of a matrix.

a)

class Matrix\_rank {

    static final int R = 3;

    static final int C = 3;

    static void swap(int mat[][],

          int row1, int row2, int col)

    {

        for (int i = 0; i < col; i++)

        {

            int temp = mat[row1][i];

            mat[row1][i] = mat[row2][i];

            mat[row2][i] = temp;

        }

    }

    static void display(int mat[][],

                     int row, int col)

    {

        for (int i = 0; i < row; i++)

        {

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

                System.out.print(" "

                          + mat[i][j]);

            System.out.print("\n");

        }

    }

    static int rankOfMatrix(int mat[][])

    {

        int rank = C;

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

        {

            if (mat[row][row] != 0)

            {

                for (int col = 0; col < R; col++)

                {

                    if (col != row)

                    {

                        double mult =

                           (double)mat[col][row] /

                                    mat[row][row];

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

                            mat[col][i] -= mult

                                       \* mat[row][i];

                    }

                }

            }

            else

            {

                boolean reduce = true;

                for (int i = row + 1; i < R; i++)

                {

                    if (mat[i][row] != 0)

                    {

                        swap(mat, row, i, rank);

                        reduce = false;

                        break ;

                    }

                }

                if (reduce)

                {

                    rank--;

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

                        mat[i][row] = mat[i][rank];

                }

                row--;

            }

        }

        return rank;

    }

    public static void main (String[] args)

    {

        int mat[][] = {{10, 20, 10},

                       {-20, -30, 10},

                       {30, 50, 0}};

        System.out.print("Rank of the matrix is : "

                               + rankOfMatrix(mat));

    }

}

14. WAP Generate a matrix having each element equal to the sum of specified submatrices of a given matrix

a)

import java.io.\*;

import java.util.\*;

class Matrix\_2{

  static void matrixBlockSum(int mat[][], int K)

  {

    int n = mat.length;

    int m = mat[0].length;

    int mat1[][] = new int[n][m];

    int cnt = 1;

    for (int i = 0; i < n; i++)

    {

      int k = 0;

      for (int j = 0; j < m; j++)

      {

        k += mat[i][j];

        mat1[i][j] = k;

        mat[i][j] = cnt;

        cnt += 1;

      }

    }

    int ans[][] = new int[n][m];

    for (int i = 0; i < n; i++)

    {

      for (int j = 0; j < m; j++)

      {

        int mnr = Math.max(0, i - K);

        int mnc = Math.max(0, j - K);

        int mxr = Math.min(n - 1, i + K);

        int mxc = Math.min(m - 1, j + K);

        int ans1 = 0;

        for (int k = mnr; k <= mxr; k++)

          ans1 += (mat1[k][mxc] - mat1[k][mnc])

          + mat[k][mnc];

        ans[i][j] = (ans1);

      }

    }

    int xx = ans.length, y = 0;

    System.out.print("[");

    for (int i = 0; i < n; i++)

    {

      System.out.print("[");

      y++;

      for (int j = 0; j < m - 1; j++)

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

      System.out.print(ans[i][m - 1] + "]");

      if (y < xx)

        System.out.print(", ");

    }

    System.out.print("] ");

  }

  public static void main(String[] args)

  {

    int mat[][]

      = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

    int K = 1;

    matrixBlockSum(mat, K);

  }

}

15. WAP Breadth First Traversal ( BFS ) on a 2D array.

a)

import java.util.LinkedList;

import java.util.Queue;

public class BFSInMatrix {

    public void BFS(int[][] grid) {

        int h = grid.length;

        if (h == 0)

            return;

        int l = grid[0].length;

        boolean[][] visited = new boolean[h][l];

        Queue<String> queue = new LinkedList<>();

        queue.add(0 + "," + 0);

        System.out.println("Breadth-First Traversal: ");

        while (queue.isEmpty() == false) {

            String x = queue.remove();

            int row = Integer.parseInt(x.split(",")[0]);

            int col = Integer.parseInt(x.split(",")[1]);

            if (row < 0 || col < 0 || row >= h || col >= l || visited[row][col])

                continue;

            visited[row][col] = true;

            System.out.print(grid[row][col] + " ");

            queue.add(row + "," + (col - 1));

            queue.add(row + "," + (col + 1));

            queue.add((row - 1) + "," + col);

            queue.add((row + 1) + "," + col);

        }

    }

    public static void main(String[] args) {

        int [][] grid = new int[][] {

                {1, 2, 3, 4},

                {5, 6, 7, 8},

                {9, 10, 11, 12},

                {13, 14, 15, 16}

        };

        BFSInMatrix d = new BFSInMatrix();

        d.BFS(grid);

    }

}

16. WAP Print boundary elements of a given matrix in clockwise manner.

a)

import java.util.\*;

class GFG {

    public static void boundaryTraversal(

        int arr[][], int N, int M)

    {

        for (int i = 0; i < M; i++) {

            System.out.print(arr[0][i] + " ");

        }

        for (int i = 1; i < N; i++) {

            System.out.print(arr[i][M - 1] + " ");

        }

        if (N > 1) {

            for (int i = M - 2; i >= 0; i--) {

                System.out.print(arr[N - 1][i] + " ");

            }

        }

        if (M > 1) {

            for (int i = N - 2; i > 0; i--) {

                System.out.print(arr[i][0] + " ");

            }

        }

    }

    public static void main(String[] args)

    {

        int arr[][]

            = { { 1, 2, 3 },

                { 4, 5, 6 },

                { 7, 8, 9 } };

        int N = arr.length;

        int M = arr[0].length;

        boundaryTraversal(arr, N, M);

    }

}

17. WAP Generate a Matrix such that given Matrix elements are equal to Bitwise OR of all corresponding row and column elements of generated Matrix.

a)

Class Gen\_mat{

static void findOriginalMatrix(int[][] B, int N,

                               int M)

{

    int[][] A = new int[N][M];

    for(int i = 0; i < N; ++i)

    {

        for(int j = 0; j < M; ++j)

        {

            A[i][j] = 1;

        }

    }

    for(int i = 0; i < N; ++i)

    {

        for(int j = 0; j < M; ++j)

        {

            // If B[i][j] is equal to 0

            if (B[i][j] == 0)

            {

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

                {

                    A[i][k] = 0;

                }

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

                {

                    A[k][j] = 0;

                }

            }

        }

    }

    for(int i = 0; i < N; ++i)

    {

        for(int j = 0; j < M; ++j)

        {

            int c = 0;

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

            {

                if (c == 1)

                    break;

                c += A[i][k];

            }

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

            {

                if (c == 1)

                    break;

                c += A[k][j];

            }

            if (c != B[i][j])

            {

                System.out.println("Not Possible");

                return;

            }

        }

    }

    for(int i = 0; i < N; ++i)

    {

        for(int j = 0; j < M; ++j)

        {

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

        }

        System.out.println();

    }

}

public static void main(String[] args)

{

    int[][] B = new int[][]{ { 1, 1, 1 },

                             { 1, 1, 1 } };

    int N = B.length;

    int M = B[0].length;

    findOriginalMatrix(B, N, M);

}

}

18. WAP to implement inheritance in java.

a)

class Calculation {

   int z;

   public void addition(int x, int y) {

      z = x + y;

      System.out.println("The sum of the given numbers:"+z);

   }

   public void Subtraction(int x, int y) {

      z = x - y;

      System.out.println("The difference between the given numbers:"+z);

   }

}

public class My\_Calculation extends Calculation {

   public void multiplication(int x, int y) {

      z = x \* y;

      System.out.println("The product of the given numbers:"+z);

   }

   public static void main(String args[]) {

      int a = 20, b = 10;

      My\_Calculation demo = new My\_Calculation();

      demo.addition(a, b);

      demo.Subtraction(a, b);

      demo.multiplication(a, b);

   }

}

19. WAP to implement interface in java.

a)

public class MammalInt implements Animal {

   public void eat() {

      System.out.println("Mammal eats");

   }

   public void travel() {

      System.out.println("Mammal travels");

   }

   public int noOfLegs() {

      return 0;

   }

   public static void main(String args[]) {

      MammalInt m = new MammalInt();

      m.eat();

      m.travel();

   }

}

20. WAP to demonstrate final keyword.

a)

class Bike9{

 final int speedlimit=90;

 void run(){

  speedlimit=400;

 }

 public static void main(String args[]){

 Bike9 obj=new  Bike9();

 obj.run();

 }

}