

# Java Basic Programs

1)Swapping is the process of exchange the values of two variables with each other. For example, variable num1 contains 1 and num2 contains 2 after swap their values are num1 contains 2 and num2 contains 1.

SOURCE CODE::

```
import java.util.*;

public class Swapnumber
{
    public static void main(String[] args)
    {
        int a, b, c;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter Value in a :");
        a=s.nextInt();
        System.out.println("Enter Value in b :");
        b=s.nextInt();

        c=a;
        a=b;
        b=c;

        System.out.println("Values in a:" +a);
        System.out.println("Values in b:" +b);
    }
}
```

OUTPUT ::

# Java Basic Programs

Enter Value in a : 1

Enter Value in b : 2

Value in a : 2

Value in b : 1

2)Java Program to Swap two numbers without using third variable

SOURCE CODE : :

```
import java.util.Scanner;

class Swapnumber
{
    public static void main(String[] args)
    {
        int a, b;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter Value in a :");
        a=s.nextInt();
        System.out.println("Enter Value in b :");
        b=s.nextInt();
        a=a+b;
        b=a-b;
        a=a-b;
        System.out.println("Values in a:" +a);
        System.out.println("Values in b:" +b);
    }
}
```

# Java Basic Programs

OUTPUT ::

Enter Value in a :

5

Enter Value in b :

2

Value in a : 2

Value in b : 5

3)Write a Java Program Swap Two Numbers using function

SOURCE CODE: :

```
import java.util.*;
```

```
class SwapTwoNumbersFunc
```

```
{
```

```
    int a,b;
```

```
    public void swap(SwapTwoNumbersFunc swp)
```

```
    {
```

```
        int temp;
```

```
        temp=swp.a;
```

```
        swp.a=swp.b;
```

# Java Basic Programs

```
        swp.b=temp;
    }
    public static void main(String s[])
    {
        SwapTwoNumbersFunc objSwp= new SwapTwoNumbersFunc();
        try
        {
            Scanner sc=new Scanner(System.in);

            System.out.print("Enter first number: ");
            objSwp.a=sc.nextInt();

            System.out.print("Enter second number: ");
            objSwp.b=sc.nextInt();

            objSwp.swap(objSwp);
            System.out.println("After swapping -a: " + objSwp.a + ", b: " + objSwp.b);
        }
        catch(Exception E)
        {
            System.out.println("Exception: " + E.toString());
        }
    }
}
```

OUTPUT ::

# Java Basic Programs

Enter first number: 1

Enter second number: 2

After swapping a: 2, b: 1

## 4)Java Program to Check entered input is Prime Number or Not

A Prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. It means it is only divisible by 1 and itself, and it start from 2. The smallest prime number is 2.

Here i will show you how to write this program. You can also download below code nad extract file, after extract file you can easily run the program.

SOURCE CODE::

```
import java.util.Scanner;

public class JavaProgram {

    public static void main(String[] args) {

        int n,i=2;

        Scanner sc = new Scanner(System.in);

        System.out.println("Please Enter any No. : ");

        n=sc.nextInt();

        while(true)

        {

            if(n==1)
```

# Java Basic Programs

```
{
    System.out.println("Smallest Prime number is 2");
    break;
}
if(n%i==0)
{
    break;
}
else
{
    i++;
}
}
if(n==i)
{
    System.out.println("Prime No.");
}
else
{
    System.out.println("Not Prime No.");
}
}
```

  

```
}
```

OUTPUT ::

# Java Basic Programs

Please Enter any No. :

97

Prime No.

BUILD SUCCESSFUL (total time: 1 second)

5)Write a Java Program to Check input number is Even or Odd

Even numbers are those which are divisible by 2, and which numbers are not divisible 2 is called odd numbers.

In this program we receive input value from keyboard and find modulo 2 ( $n\%2$ ) of number if reminder of number is zero then that number is even other wise odd.

SOURCE CODE ::

```
import java.util.Scanner;

public class Even_odd {

    public static void main(String[] args) {

        int n,i;

        System.out.println("Please Enter any No. u Want to Check :");
```

# Java Basic Programs

```
Scanner sc = new Scanner(System.in);
```

```
n = sc.nextInt();
```

```
if(n%2==0)
```

```
{
```

```
    System.out.println("Entered number is Even Number");
```

```
}
```

```
else
```

```
{
```

```
    System.out.println("Entered number is Odd Number");
```

```
}
```

```
}
```

```
}
```

OUTPUT ::

Please Enter any No. u Want to Check :

87

Entered number is Odd Number

6)Write a Java Program to find Factorial using While loop

Factorial of any number is the product of an integer and all the integers below it



# Java Basic Programs

For example :: factorial of 5 is  $5! = 5 * 4 * 3 * 2 * 1 = 120$ .

In below source code i will show you how to Write a Java program to find factorial using while loop.

SOURCE CODE ::

```
import java.util.Scanner;

public class Factorial{

    public static void main(String[] args)
    {

        int n, fact=1;

        Scanner sc = new Scanner(System.in);

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

        n=sc.nextInt();

        int i=1;

        while(n>=i)
        {

            fact=fact*i;

            i++;

        }

    }

}
```

# Java Basic Programs

```
        System.out.println("Factorial is :"+fact);  
    }  
  
}
```

OUTPUT ::

Enter any number :

9

Factorial is :362880

7)Write a Java Program to find Reverse Number using While Loop

Reverse number means reverse the position of all digits of any number.

For example : Reverse of 435 is 534.

For this program you need modulus operator concept and while loop, while loop is used for check condition and modulus used for find the remainder.

SOURCE CODE ::

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

```
public class ReverseNumber {  
Prepared By :Chandra Sekhar Bijibilla  
10
```

# Java Basic Programs

```
public static void main(String[] args) {  
  
    int n,i,reverse=0,k;  
  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter the no. which u want to reverse : ");  
    n=sc.nextInt();  
    k=n;  
    while(n>0)  
    {  
        i=n%10;  
        reverse=reverse*10+i;  
        n=n/10;  
    }  
    System.out.println("Reverse of " + k + " is : "+ reverse);  
  
    }  
  
}
```

OUTPUT ::

Enter the no. which u want to reverse : 123456

Reverse of 123456 is : 654321

8)Java Program to display Fibonacci Series using While Loop

Fibonacci series is in the form of 1, 1, 2, 3, 5, 8, 13, 21,.....

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

To find this series we add two previous terms/digits and get next term/number.

SOURCE CODE ::

```
import java.util.Scanner;

public class Fibonacci {

    public static void main(String[] args) {

        int a=1,b=1,n,fibon=1;

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter how many terms u want : ");
        n=sc.nextInt();
        System.out.print(fibon);
        while(n>1)
        {
            System.out.print(" "+fibon);
            fibon=a+b;
            a=b;
            b=fibon;
            n--;
        }
    }
}
```

# Java Basic Programs

```
}  
    System.out.println("");  
  
}  
  
}
```

OUTPUT ::

Enter how many terms u want :

11

1 1 2 3 5 8 13 21 34 55 89

9)Write a Java Program to find GCD (HCF) of Two Numbers

The greatest common divisor (gcd) of two or more integers, when at least one of them is not zero, is the largest positive integer that is a divisor of both numbers.

For example, the GCD of 8 and 12 is 4.

This is java program to find the gcd of given two numbers. GCD is calculated using Euclidean Algorithm.

Here is the source code of the Java Program to Find the GCD of 2 Numbers. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

SOURCE CODE ::

# Java Basic Programs

```
import java.util.Scanner;

public class GCD {

    public static void main(String[] args) {

        int a,b,gcd,temp1,temp2;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter 1st No.");
        a=sc.nextInt();
        System.out.println("Enter 2nd No.");
        b=sc.nextInt();

        temp1=a;
        temp2=b;

        while(true)
        {
            if(a>b)
            {
                if(a%b==0)
                {
                    gcd=b;
                    break;
                }
            }
        }
    }
}
```

# Java Basic Programs

```
        else
        {
            a=a%b;
        }
    }
    else
    {
        if(b%a==0)
        {
            gcd=a;
            break;
        }
        else
        {
            b=b%a;
        }
    }
}
```

```
        System.out.println("GCD of "+ temp1 +" and "+ temp2 +" = "+ gcd);
    }

}
```

OUTPUT ::

# Java Basic Programs

Enter 1st No.

24

Enter 2nd No.

60

GCD of 24 and 60 = 12

10)Java Program to Check whether Given Number is Armstrong or Not

Armstrong number is a number that is the sum of its own digits each raised to the power of the number of digits is equal to the number itself.

For example:

Three Digits Armstrong number is 153,  $1^3 + 5^3 + 3^3 = 153$

Four Digits Armstrong number is 1634,  $1^4 + 6^4 + 3^4 + 4^4 = 1634$

and So on.....

SOURCE CODE ::

```
import java.util.Scanner;
import java.lang.Math;
public class Armstrong {

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

```
        int a,b,n,arm=0,temp;
```

Prepared By :Chandra Sekhar Bijibilla



# Java Basic Programs

```
System.out.print("Enter No. which u want to Check :");

Scanner sc = new Scanner(System.in);

n = sc.nextInt();

temp=n;

a=countdigit(n);

while(n>0)

{

    b=n%10;

    n=n/10;

    arm = (int) (arm + Math.pow(b, a)) ;

}

if(arm==temp)

{

    System.out.println("Armstrong No.");

}

else

{

    System.out.println("Not Armstrong No.");

}

}
```

```
static int countdigit(int n)

{

    int c=0;

    while(n>0)
```

# Java Basic Programs

```
{  
    n=n/10;  
  
    c++;  
}  
return c;  
  
}  
  
}
```

OUTPUT ::

Enter No. which u want to Check :153

Armstrong No.

-----

Enter No. which u want to Check :222

Not Armstrong No.

-----

Enter No. which u want to Check :1741725

Armstrong No.

11)Write a Java Program to find Largest among three Numbers

Prepared By :Chandra Sekhar Bijibilla

18

# Java Basic Programs

Input three number from user and compare these number with each others and find largest number among these three numbers.

SOURCE CODE ::

```
import java.util.Scanner;

public class Largest
{

    public static void main(String[] args)
    {

        int a,b,c,largest;

        Scanner sc=new Scanner(System.in);

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

        a=sc.nextInt();

        b=sc.nextInt();

        c=sc.nextInt();


        if(a>=b && a>=c)
        {
            System.out.println("Largest number = "+a);
        }

        if(b>=a && b>=c)
        {
            System.out.println("Largest number = "+b);
        }

        if(c>=a && c>=b)
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
{  
    System.out.println("Largest number = "+c);  
}  
}  
}
```

OUTPUT ::

Enter three numbers:

23

33

14

Largest number = 33

12)Write a Java Program to find given number is Palindrome or Not

A Palindrome number is a number that remains the same when its digits are reversed.

Like 12321, for example: we take 121 and reverse it, after revers it is same as original number.

SOURCE CODE ::

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

Prepared By :Chandra Sekhar Bijibilla

20

# Java Basic Programs

```
class Palindrome
{
public static void main(String[] args)
{
    int a,no,b,temp=0;
    Scanner s=new Scanner(System.in);
    System.out.println("Enter any num: ");
    no=s.nextInt();
    b=no;
    while(no>0)
    {
        a=no%10;
        no=no/10;
        temp=temp*10+a;
    }
    if(temp==b)
    {
        System.out.println("Palindrome");
    }
    else
    {
        System.out.println("not Palindrome");
    }
}
}
```

OUTPUT ::

Prepared By :Chandra Sekhar Bijibilla  
21

# Java Basic Programs

Enter any num:

1221

Palindrome

13)Write a Java Program to Print Floyd's Triangle using For Loop

To print the Floyd's triangle in Java programming, you have to use two for loops, the outer loop is responsible for rows and the inner loop is responsible for columns and start printing the Floyd's triangle as shown in the following program.

As we know that, Floyd's triangle is a right angled-triangle using the natural numbers, so the following Java Program ask to the user to enter the range ( how many line he/she want) to print the Floyd's Triangle.

Following is the sample of Floyd's Triangle :

SOURCE CODE ::

```
import java.util.Scanner;

public class FloydTriangle {

    public static void main(String[] args) {
Prepared By :Chandra Sekhar Bijibilla
22
```

# Java Basic Programs

```
int i,j,k=1,n;
System.out.print("Enter how many lines u want : ");
Scanner sc = new Scanner(System.in);
n=sc.nextInt();
for(i=1;i<=n;i++)
{

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

        System.out.print(" "+k);
        k++;
    }
    System.out.println("");
}
}
```

OUTPUT ::

Enter how many lines u want : 6

1

2 3

Prepared By :Chandra Sekhar Bijibilla  
23

# Java Basic Programs

4 5 6

7 8 9 10

11 12 13 14 15

16 17 18 19 20 21

14) Write a Java Program to Print Pascal Triangle using Recursion

Following Java Program ask to the user to enter the number of line/row upto which the Pascal's triangle will be printed to print the Pascal's triangle on the screen.

java-pascal-triangle

Given below is the program which uses the recursion to print Pascal's triangle.

SOURCE CODE ::

```
import java.util.Scanner;

public class PascalTriangle {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        int rows = scanner.nextInt();

        System.out.println("Pascal Triangle:");
```

Prepared By :Chandra Sekhar Bijibilla



# Java Basic Programs

```
print(rows);
scanner.close();

}

public static void print(int n) {

    for (int i = 0; i < n; i++) {
        for (int k = 0; k < n - i; k++) {
            System.out.print(" "); // print space for triangle like structure
        }
        for (int j = 0; j <= i; j++) {
            System.out.print(pascal(i, j) + " ");
        }
        System.out.println();
    }
}

public static int pascal(int i, int j) {
    if (j == 0 || j == i) {
        return 1;
    } else {
        return pascal(i - 1, j - 1) + pascal(i - 1, j);
    }
}
}
```

# Java Basic Programs

OUTPUT ::

Enter the number of rows to print: 5

Pascal Triangle:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

15)Java Program to Make a Simple Calculator using switch case

Java Program to Make a Simple Calculator using switch case in Java Programming which performs the basic four mathematical operations i.e., addition, subtraction, multiplication, and division.

Following is a simple Java Program which is a menu-driven program based on simple calculation like addition, subtraction, multiplication and division according to user's choice:

SOURCE CODE ::

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

```
public class Calculator
```

```
{
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
public static void main(String args[])
{
    float a, b, res;
    char choice, ch;
    Scanner scan = new Scanner(System.in);

    do
    {
        System.out.print("1. Addition\n");
        System.out.print("2. Subtraction\n");
        System.out.print("3. Multiplication\n");
        System.out.print("4. Division\n");
        System.out.print("5. Exit\n\n");
        System.out.print("Enter Your Choice : ");
        choice = scan.next().charAt(0);
        switch(choice)
        {
            case '1' : System.out.print("Enter Two Number : ");
                a = scan.nextFloat();
                b = scan.nextFloat();
                res = a + b;
                System.out.print("Result = " + res);
                break;
            case '2' : System.out.print("Enter Two Number : ");
                a = scan.nextFloat();
                b = scan.nextFloat();
                res = a - b;
                System.out.print("Result = " + res);
```

# Java Basic Programs

```
        break;
    case '3' : System.out.print("Enter Two Number : ");
        a = scan.nextFloat();
        b = scan.nextFloat();
        res = a * b;
        System.out.print("Result = " + res);
        break;
    case '4' : System.out.print("Enter Two Number : ");
        a = scan.nextFloat();
        b = scan.nextFloat();
        res = a / b;
        System.out.print("Result = " + res);
        break;
    case '5' : System.exit(0);
        break;
    default : System.out.print("Wrong Choice!!!");
        break;
}
System.out.print("\n-----\n");
}while(choice != 5);
}
}
```

OUTPUT ::

# Java Basic Programs

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter Your Choice : 1

Enter Two Number : 43

23

Result = 66.0

-----

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter Your Choice : 4

Enter Two Number : 88

11

Result = 8.0

-----

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

# Java Basic Programs

Enter Your Choice : 5

16)Java program to convert Fahrenheit to Celsius

Java Program to convert Fahrenheit to celsius in Java programming, you have to ask to the user to enter the temperature in Fahrenheit temperature to convert it into celsius to display the equivalent temperature value in centigrade as shown in the following program.

SOURCE CODE ::

```
import java.util.Scanner;

public class FahrenheitToCelsius {
    public static void main(String[] args) {
        float temperatue;
        Scanner in = new Scanner(System.in);

        System.out.println("Enter temperatue in Fahrenheit");
        temperatue = in.nextInt();

        temperatue = ((temperatue - 32)*5)/9;

        System.out.println("Temperatue in Celsius = " + temperatue);
    }
}
```

# Java Basic Programs

OUTPUT ::

Enter temperatue in Fahrenheit

87

Temperatue in Celsius = 30.555555

17)Java program to generate random numbers

To generate random numbers in Java programming, you have to create the object of Random class available in the java.util.Random package as shown in the following program.

Following Java program to generate random numbers: This code generates random numbers in range 0 to 10000

SOURCE CODE ::

```
import java.util.Scanner;  
import java.util.Random;  
  
public class Random_numbers{  
    public static void main(String[] args) {  
        int c;  
        Random t = new Random();  
  
        for (c = 1; c <= 15; c++) {  
            System.out.println(t.nextInt(10000));  
        }  
    }  
}
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
}  
}  
}
```

OUTPUT ::

7307

3364

9601

6944

1573

7080

199

1261

3046

5751

1963

9607

6334

8294

4390

18)Java program to find roots of quadratic equation

Write a Java program to find roots of quadratic equation in all cases. This Java Program To Compute Roots of Quadratic Equation makes use of If – Else Block.

What is a Quadratic Equation?

Prepared By :Chandra Sekhar Bijibilla

32



# Java Basic Programs

It is a term used in Elementary Algebra. The Standard Form of a Quadratic Equation is  $ax^2 + bx + c = 0$ , where a, b, c are constant values which cannot be changed and x is a variable entity.

## Quadratic Equation Formula

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

## Conditions For Discriminants

If  $b^2 - 4ac = 0$ , One Real Solution is possible

If  $b^2 - 4ac$  is Positive, Two Real Solutions are possible

If  $b^2 - 4ac$  is Negative, Two Complex Solutions are possible

Enter the Quadratic Equation as an input. Now we use the given formulas to calculate the roots of the equation. The `java.lang.*` package consists of `Math.sqrt()` method which helps to calculate the Square Root of the Discriminant.

Here is the source code of the Java Program to Find the Roots of Quadratic Equation. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

SOURCE CODE : :

```
import java.util.Scanner;

public class quadratic_equation
{
    public static void main(String args[])
    {
        double a, b, c;

        double root1, root2, D, sqrroot;

        System.out.println("\nEnter The Values");

        Scanner sc = new Scanner(System.in);

        System.out.print("\nA: ");
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
a = sc.nextDouble();
System.out.print("\nB: ");
b = sc.nextDouble();
System.out.print("\nC: ");
c = sc.nextDouble();
System.out.println("\nGiven quadratic equation : "+a+"x^2 + "+b+"x + "+c);

if((a==0)&&(b==0)&&(c==0))
{
    System.out.println("\nEnter atleast two non-zero co-efficients");
    System.exit(0);
}
else if(a==0 && b!=0)
{
    root1= -c/b;
    System.out.println("\nThe roots are:"+root1);
    System.exit(0);
}
else if(a==0&&b==0&&c!=0)
{
    System.out.println("\nThe equation has no solution");
    System.exit(0);
}
else
{
    D = b*b - 4*a*c;
    sqrroot = Math.sqrt(D);
```

# Java Basic Programs

```
if(D<0)
{
    System.out.println("\nRoots Are Imaginary\n");
}
else if(D == 0)
{
    System.out.println("\nRoots are real and equal");
    root1 = (-b + sqrtroot) / (2*a);
    System.out.println("\nRoots are : "+root1);

}
else
{
    System.out.println("\nRoots are real and unequal");
    root1 = (-b + sqrtroot) / (2*a);
    root2 = (-b - sqrtroot) / (2*a);
    System.out.println("\nRoot 1 = " + root1 + "\n");
    System.out.println("Root 2 = " + root2 + "\n");
}
}
```

```
}
```

```
}
```

OUTPUT ::

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

Enter The Values

A: 2

B: 8

C: 4

Given quadratic equation :  $2.0x^2 + 8.0x + 4.0$

Roots are real and unequal

Root 1 = -0.5857864376269049

Root 2 = -3.414213562373095

Description-

This very simple program inputs the quadratic equation by inputting the coefficients of  $x^2$ ,  $x$  and constant term.

After that the value of discriminant is calculated by its formula  $\text{Discriminant} = b^2 - 4ac$ .

As we know that the roots of a Quadratic equation depends largely on the value of discriminant ,three cases occur :

If discriminant is greater than zero then two distinct and real roots are obtained , which are calculated and given by Dharacharya formula.

If discriminant is zero then it means that the equation is a Perfect Square and two equal roots are obtained.

If discriminant is less than zero then no real roots are obtained and thus it has no real solution and since no datatype of JAVA can hold imaginary value, no solution can be represented here.

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

19)Java program to multiply N numbers without using \* operator

Write a Java program to multiply N numbers without using \* multiplication operator.As we all know that there are lot of methods or techniques to solve problem to produce desired output.

Basic idea of this java program to show varies methods of problem solving techniques, although we have standard operator " \* " to perform the basic operation of multiplication.

Our aim is to do that without using " \* " operator we will show you multiplication operation on integer but you can extend it to your aspect of need.

In this program, first we input N integers from the user and store them into an Array.Then we make iterations inside loop and add the previous element k times to the Current element to produce desired output.

Here, below is the source code of Java Program to multiply N numbers without using \* multiplication operator which is successfully compiled and run(Netbeans) on the Windows System to produce particular output.Let's look at the program below.

SOURCE CODE : :

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

```
public class Multiplication
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        int i,n;
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("How many elements u want to multiply : ");
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
n=sc.nextInt();

int a[]=new int[n];

System.out.println("Enter your elements below :---- \n");
for(i=0;i<n;i++)
{
    System.out.print("Enter "+(i+1)+" Element : ");
    a[i]=sc.nextInt();
}

for(i=1;i<n;i++)
{
    int j=0,sum=0;
    while(j<a[i])
    {
        sum+=a[i-1];
        j++;
    }
    a[i]=sum;
}

System.out.println("\nMultiplication of "+n+" Numbers :"+ a[n-1] +" \n");
}
}
```

OUTPUT ::

# Java Basic Programs

How many elements u want to multiply : 6

Enter your elements below :----

Enter 1 Element : 1

Enter 2 Element : 2

Enter 3 Element : 3

Enter 4 Element : 4

Enter 5 Element : 5

Enter 6 Element : 6

Multiplication of 6 Numbers :720

20)Write a Java Program for Binary Search on Unsorted Array

Binary Search

Java Program first ask to the user to enter “how many element he/she want to store in a array”, then ask to enter the array element one by one in any order in which they want”.

After storing the element in the array, following program again ask to the user to enter the element which he/she want to search in the array whether the entered element/number is present in the list or not.

The searching technique used here is binary search which is fast technique :

SOURCE CODE ::

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

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
public class BinarySearch {

    //-----Function to Sort Array-----

    static int[] sorted(int arr[])
    {
        int l,max,temp;
        l=arr.length;
        for(int i=0;i<l;i++)
        {
            for(int j=i+1;j<l;j++)
            {
                if(arr[i]>arr[j])
                {
                    temp=arr[j];
                    arr[j]=arr[i];
                    arr[i]=temp;
                }
            }
        }
        return arr;
    }

    //-----function to search key in array-----

    static int binarysearch(int a[],int key)
    {
```



# Java Basic Programs

```
int beg,l,end,mid;

l=a.length;

beg=0;

end=l-1;

mid=(beg+end)/2;

while(beg<=end)

{

    if(a[mid]<key)

    {

        beg=mid+1;

        mid=(beg+end)/2;

    }

    else if(a[mid]==key)

    {

        break;

    }

    else

    {

        end=mid-1;

        mid=(beg+end)/2;

    }

    if(beg>end)

    {

        mid=0;

    }

}
```

# Java Basic Programs

```
}

return mid;
}

public static void main(String[] args) {

    int i,n,b[],key,c;
    System.out.print("Enter size of array: ");
    Scanner sc = new Scanner(System.in);
    n=sc.nextInt();
    int a[]=new int[n];

    //-----Enter values to array-----

    for(i=0;i<n;i++)
    {
        System.out.print(i+1 +" element : ");
        a[i]=sc.nextInt();
    }
    System.out.print("");

    //-----Sort the array-----

    b=sorted(a);
    for(i=0;i<n;i++)
    {
```

# Java Basic Programs

```
System.out.println(i+1 +" element : "+b[i]);

}

//-----Enter search key u want to find-----

System.out.println("Enter the number u want to search: ");

key=sc.nextInt();

c=binarysearch(b,key);

if(c==0)
{
    System.out.println("Sorry,Key doesn't exist in the array");
}
else
{
    System.out.println("Position of element is : "+(c+1));
}

}

}
```

OUTPUT ::

# Java Basic Programs

Enter size of array: 6

1 element : 3

2 element : 6

3 element : 1

4 element : 8

5 element : 4

6 element : 0

Sorted Array

1 element : 0

2 element : 1

3 element : 3

4 element : 4

5 element : 6

6 element : 8

Enter the number u want to search:

4

Position of element is : 4

21)Write a Java Program to Add Two Matrices

To add two matrices in Java Programming, you have to ask to the user to enter the elements of both the  $n \times n$  matrix, now start adding the two matrix to form a new/third matrix which is the addition result of the two given matrix.

Prepared By :Chandra Sekhar Bijibilla

44

# Java Basic Programs

After adding the two matrices, display the third matrix which is the result of the addition of the two matrices.

Following Java Program add two  $n \times n$  matrices to form the third matrix :

SOURCE CODE ::

```
import java.util.Scanner;

public class Add_matrices {
    public static void main(String args[])
    {
        int m, n, c,d;
        Scanner in = new Scanner(System.in);

        System.out.println("Enter the number of rows and columns of matrix");
        m = in.nextInt();
        n = in.nextInt();

        int first[][] = new int[m][n];
        int second[][] = new int[m][n];
        int sum[][] = new int[m][n];

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

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
            {
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
System.out.print(c+""+d+" Element : ");  
first[c][d] = in.nextInt();  
}
```

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

```
for ( c = 0 ; c < m ; c++ )  
    for ( d = 0 ; d < n ; d++ )  
    {  
        System.out.print(c+""+d+" Element : ");  
        second[c][d] = in.nextInt();  
    }
```

```
//-----Add two matrix-----
```

```
for ( c = 0 ; c < m ; c++ )  
    for ( d = 0 ; d < n ; d++ )  
        sum[c][d] = first[c][d] + second[c][d];
```

```
System.out.println("Sum of entered matrices:-");
```

```
for ( c = 0 ; c < m ; c++ )  
{  
    for ( d = 0 ; d < n ; d++ )  
        System.out.print(sum[c][d]+"\\t");  
  
    System.out.println();  
}
```

# Java Basic Programs

```
}  
}
```

OUTPUT ::

Enter the number of rows and columns of matrix

2

2

Enter the elements of first matrix

00 Element : 1

01 Element : 2

10 Element : 3

11 Element : 4

Enter the elements of second matrix

00 Element : 1

01 Element : 2

10 Element : 3

11 Element : 4

Sum of entered matrices:-

2    4

6    8

22)Write a Java Program for Linear Search on unsorted array

Prepared By :Chandra Sekhar Bijibilla

47

# Java Basic Programs

To search any element present inside the array in Java Programming using linear search technique, you have to use only one for loop to check whether the entered number is found in the list or not as shown in the following program.

Following Java program first ask to the user to enter the array size then it will ask to enter the array elements, then it will finally ask to enter a number to be search in the given array to check whether it is present in the array or not, if it is present then the program will show the position of that number present in the array:

SOURCE CODE ::

```
import java.util.Scanner;

public class LinearSearch {

    //-----Function to Sort Array-----

    static int[] sorted(int arr[])
    {
        int l,max,temp;
        l=arr.length;
        for(int i=0;i<l;i++)
        {
            for(int j=i+1;j<l;j++)
            {
                if(arr[i]>arr[j])
```



# Java Basic Programs

```
{
    temp=arr[j];
    arr[j]=arr[i];
    arr[i]=temp;
}
}
}
return arr;
}

//-----function to search key in array-----

static void search(int arr[],int key)
{
    int i,l,flag=0,position=0;
    l=arr.length;
    for(i=0;i<l;i++)
    {
        if(arr[i]==key)
        {
            position=i+1;
            flag=1;
            break;
        }
    }
    if(flag==0)
    {
        System.out.println("Sorry,Key doesn't exist in the array");
    }
}
```

# Java Basic Programs

```
}  
else  
{  
    System.out.println(key+ " found at position " + position);  
}  
  
}
```

```
//-----Main Menu-----
```

```
public static void main(String[] args) {  
  
    int i,n,b[],key,c;  
    System.out.print("Enter size of array: ");  
    Scanner sc = new Scanner(System.in);  
    n=sc.nextInt();  
    int a[]=new int[n];
```

```
//-----Enter values to array-----
```

```
for(i=0;i<n;i++)  
{  
    System.out.print(i+1 +" Element : ");  
    a[i]=sc.nextInt();  
}
```

```
//-----Sort the array-----
```

```
b=sorted(a);
```

# Java Basic Programs

```
System.out.println("\nSorted Array :-->");
```

```
//-----Display the Sorted Array-----
```

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

```
{
```

```
    System.out.println(i+1 +" Element : "+a[i]);
```

```
}
```

```
//-----Enter search key u want to find-----
```

```
System.out.print("\nEnter the number u want to search: ");
```

```
key=sc.nextInt();
```

```
search(b,key);
```

```
}
```

```
}
```

OUTPUT ::

# Java Basic Programs

Enter size of array: 5

1 Element : 4

2 Element : 1

3 Element : 5

4 Element : 6

5 Element : 3

Sorted Array :-->

1 Element : 1

2 Element : 3

3 Element : 4

4 Element : 5

5 Element : 6

Enter the number u want to search: 3

3 found at position 2

23)Write a Java Program to Perform Bubble Sort

To perform bubble sort in Java Programming, you have to ask to the user to enter the array size then ask to enter the array elements, now start sorting the array elements using the bubble sort technique.

Following Java Program sort the array using the Bubble Sort technique :

SOURCE CODE ::

Prepared By :Chandra Sekhar Bijibilla

52

# Java Basic Programs

```
import java.util.Scanner;

public class BubbleSort {

//-----Enter data Function-----

static int [] enter_data()
{
    int i,n;
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter no. of elements u want to sort :");
    n=sc.nextInt();
    int a[]=new int[n];
    System.out.println("Enter elements--->");

    for(i=0;i<n;i++)
    {
        System.out.print((i+1)+" Element : ");
        a[i]=sc.nextInt();
    }
    return a;
}

//-----Bubble Sort Function-----
```

# Java Basic Programs

```
static int [] bubblesort(int a[])
{
    int i,j,temp,len;
    len=a.length;

    for(i=0;i<len;i++)
    {
        for(j=0;j<len-i-1;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }

    return a;
}

//-----Print array-----

static void printarray(int arr[])
{
    int i,len;
    len=arr.length;

    System.out.println("\nAfter Sorting Elements are : ");
```

# Java Basic Programs

```
for(i=0;i<len;i++)  
{  
    System.out.println((i+1)+" Element : "+arr[i]);  
}  
}
```

```
//-----Main Function-----
```

```
public static void main(String[] args) {  
  
    int a[],b[];  
  
    a=enter_data();  
  
    b=bubblesort(a);  
  
    printarray(b);  
  
}  
  
}
```

OUTPUT ::

# Java Basic Programs

Enter no. of elements u want to sort :5

Enter elements--->

1 Element : 3

2 Element : 1

3 Element : 6

4 Element : 9

5 Element : 3

After Sorting Elements are :

1 Element : 1

2 Element : 3

3 Element : 3

4 Element : 6

5 Element : 9

## 24)Java Program to perform Selection Sort

Selection sort is a sorting algorithm which sorts the given array elements either in descending or ascending order by finding the misplaced element first and putting it in its final place.

Selection sort algorithm first selects the lowest (or highest) element and places it in the ordered position. Selection sort is recommended only for small size arrays (less than 1000 elements).

It doesn't give much performance as compared to other sorting algorithms when used on large size arrays.

It has  $O(n^2)$  time complexity, making it inefficient on large lists, and generally performs worse than the similar insertion sort.

Selection sort is noted for its simplicity, and it has performance advantages over more complicated algorithms in certain situations, particularly where auxiliary memory is limited.

Prepared By :Chandra Sekhar Bijibilla



# Java Basic Programs

Worst case performance :  $O(n^2)$

Best case performance :  $O(n^2)$

Average case performance :  $O(n^2)$

Here is source code of the Java Program to implement Selection Sort. The Java program is successfully compiled and run on a Windows system(NetBeans). The program output is also shown below.

SOURCE CODE : :

```
import java.util.Scanner;

public class SelectionSort {

    //-----Enter data Function-----

    static int [] enter_data()
    {
        int i,n;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of elements u want to sort :");
        n=sc.nextInt();
        int a[]=new int[n];
        System.out.println("Enter elements--->");
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
for(i=0;i<n;i++)
{
    System.out.print((i+1)+" Element : ");
    a[i]=sc.nextInt();
}
return a;
}

//-----Selection Sort Function-----

static int [] selectionsort(int a[])
{
    int i,j,temp,len,min;
    len=a.length;

    for(i=0;i<len-1;i++)
    {
        min=a[i];
        for(j=i+1;j<len;j++)
        {
            if(min>a[j])
            {
                temp=a[j];
                a[j]=min;
                min=temp;
            }
            a[i]=min;
        }
    }
```

# Java Basic Programs

```
}

return a;
}

//-----Print array-----

static void printarray(int arr[])
{
    int i,len;
    len=arr.length;
    System.out.println("\nAfter Sorting Elements are : ");
    for(i=0;i<len;i++)
    {
        System.out.println((i+1)+" Element : "+arr[i]);
    }
}

//-----Main Function-----

public static void main(String[] args) {

    int a[],b[];

    a=enter_data();

    b=selectionsort(a);
```

# Java Basic Programs

```
    printarray(a);  
  
    }  
  
}
```

OUTPUT ::

Enter no. of elements u want to sort :5

Enter elements--->

1 Element : 3

2 Element : 1

3 Element : 8

4 Element : 5

5 Element : 7

After Sorting Elements are :

1 Element : 1

2 Element : 3

3 Element : 5

4 Element : 7

5 Element : 8

25)Java Program to perform Insertion Sort

The Insertion sort is another simple sorting algorithm, which can be used to sort any linear data structure like an array or linked list.

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

It is a simple sorting algorithm that builds the final sorted array one item at a time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort.

Every repetition of insertion sort removes an element from the input data, inserting it into the correct position in the already-sorted list, until no input elements remain.

The choice of which element to remove from the input is arbitrary, and can be made using almost any choice algorithm.

Time Complexity :

Worst case performance :  $O(n^2)$  comparisons, swaps

Best case performance :  $O(n)$  comparisons,  $O(1)$  swaps

Average case performance :  $O(n^2)$  comparisons, swaps

Here is the source code of the Java Program to perform Insertion Sort. The Java program is successfully compiled and run on a Windows system. The program output is also shown below :

SOURCE CODE : :

```
import java.util.Scanner;

public class InsertionSort {

    //-----Enter data Function-----

    static int [] enter_data()
    {
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
int i,n;

Scanner sc = new Scanner(System.in);

System.out.print("Enter no. of elements u want to sort :");

n=sc.nextInt();

int a[]=new int[n];

System.out.println("Enter elements--->");


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

{

    System.out.print((i+1)+" Element : ");

    a[i]=sc.nextInt();

}

return a;

}


//-----Insertion Sort Function-----


static int [] insertionsort(int a[])

{

    int i,j,temp,len,min;

    len=a.length;

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

    {

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

        {

            if(a[j-1]>a[j])
```

# Java Basic Programs

```
{
    temp=a[j];
    a[j]=a[j-1];
    a[j-1]=temp;
}
}
}

return a;
}

//-----Print array-----

static void printarray(int arr[])
{
    int i,len;
    len=arr.length;
    System.out.println("\nAfter Sorting Elements are : ");
    for(i=0;i<len;i++)
    {
        System.out.println((i+1)+" Element : "+arr[i]);
    }
}

//-----Main Function-----

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

# Java Basic Programs

```
int a[],b[];

a=enter_data();

b=insertionsort(a);

printarray(a);

}

}
```

OUTPUT : :

Enter no. of elements u want to sort :10

Enter elements--->

1 Element : 4

2 Element : 1

3 Element : 5

4 Element : 8

5 Element : 2

6 Element : 9

7 Element : 3

8 Element : 4

9 Element : 1

Prepared By :Chandra Sekhar Bijibilla



# Java Basic Programs

10 Element : 0

After Sorting Elements are :

1 Element : 0

2 Element : 1

3 Element : 1

4 Element : 2

5 Element : 3

6 Element : 4

7 Element : 4

8 Element : 5

9 Element : 8

10 Element : 9

26)Java Program to perform Quick Sort

Quicksort is a divide and conquer algorithm. In a divide and conquer sorting algorithm the original data is separated into two parts “divide” which are individually sorted and “conquered” and then combined.

If the array contains only one element or zero elements than the array is sorted.

If the array contains more than one element than:

Select an element from the array. This element is called the “pivot element”. For example select the element in the middle of the array.

All elements which are smaller then the pivot element are placed in one array and all elements which are larger are placed in another array.

Sort both arrays by recursively applying Quicksort to them.

Combine the arrays.

Quicksort can be implemented to sort “in-place”. This means that the sorting takes place in the array and that no additional array needs to be created.

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

This is a Java Program to perform Quick Sort Algorithm. Here is the source code of the Java program to implement Quick Sort Algorithm. The program output is also shown below.

SOURCE CODE : :

```
import java.util.Scanner;

public class QuickSort {

    //-----Enter data Function-----

    static int [] enter_data()
    {
        int i,n;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of elements u want to sort :");
        n=sc.nextInt();
        int a[]=new int[n];
        System.out.println("Enter elements--->");

        for(i=0;i<n;i++)
        {
            System.out.print((i+1)+" Element : ");
            a[i]=sc.nextInt();
        }
        return a;
    }
}
```

Prepared By :Chandra Sekhar Bijibilla

# Java Basic Programs

```
//-----QuickSort-----
```

```
static void quicksort(int a[],int p,int r)
```

```
{
```

```
    if(p<r)
```

```
    {
```

```
        int q;
```

```
        q=partition(a,p,r);
```

```
        quicksort(a,p,q);
```

```
        quicksort(a,q+1,r);
```

```
    }
```

```
}
```

```
//-----Partition-----
```

```
static int partition(int a[],int p,int r)
```

```
{
```

```
    int i, j, pivot, temp;
```

```
    pivot = a[p];
```

```
    i = p;
```

```
    j = r;
```

```
    while(true)
```

```
    {
```

```
        while(a[i] < pivot && a[i] != pivot)
```

```
        {
```

# Java Basic Programs

```
        i++;
    }

    while((a[j] > pivot) && (a[j] != pivot))
    {

        j--;
    }

    if(i < j)
    {
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }
    else
    {
        return j;
    }
}

//-----Print array-----

static void printarray(int arr[])
{
    int i,len;
```

# Java Basic Programs

```
len=arr.length;

System.out.println("\nAfter Sorting Elements are : ");

for(i=0;i<len;i++)
{
    System.out.println((i+1)+" Element : "+arr[i]);
}

}

//-----Main Function-----

public static void main(String[] args) {

    int a[],b[],p=0,r;

    a=enter_data();
    r=a.length;
    quicksort(a,p,r-1);
    printarray(a);

}

}
```

OUTPUT : :

# Java Basic Programs

Enter no. of elements u want to sort :7

Enter elements--->

1 Element : 5

2 Element : 1

3 Element : 9

4 Element : 3

5 Element : 4

6 Element : 0

7 Element : 8

After Sorting Elements are :

1 Element : 0

2 Element : 1

3 Element : 3

4 Element : 4

5 Element : 5

6 Element : 8

7 Element : 9