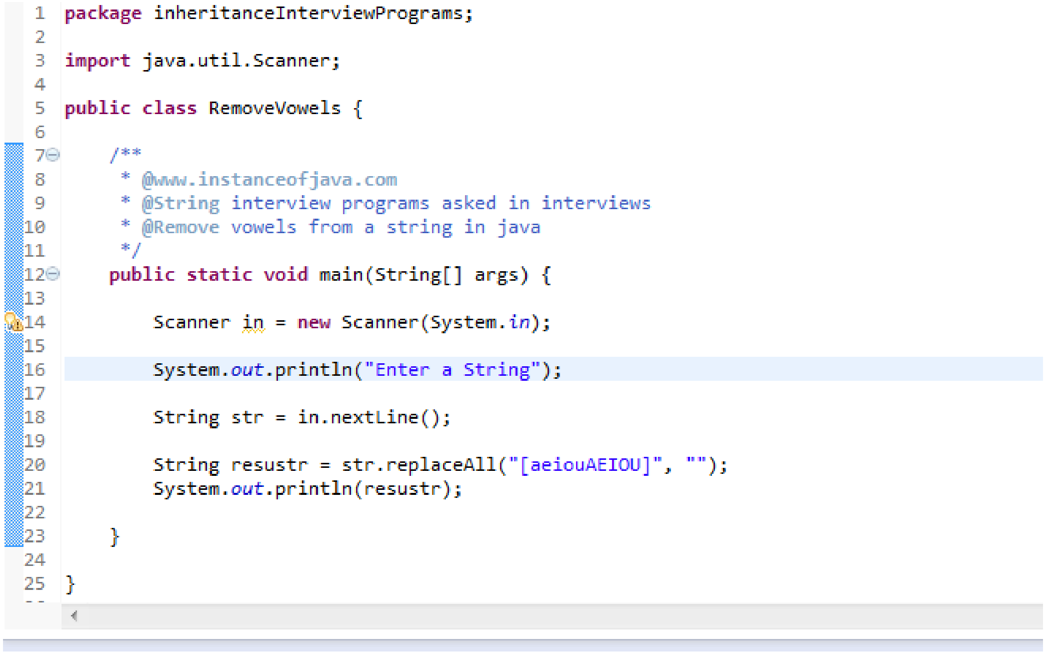
## Java example program to remove all vowels from a String

```
package inheritanceInterviewPrograms;
public class RemoveVowels {

    /**
     * @www.instanceofjava.com
     * @String interview programs asked in interviews
     * @Remove vowels from a string in java
     */

 public static void main(String[] args) {

        String str = "RemoveVowels";
        String resustr = str.replaceAll("[aeiouAEIOU]", "");
        System.out.println(resustr);

    }

}
```

**o/p**

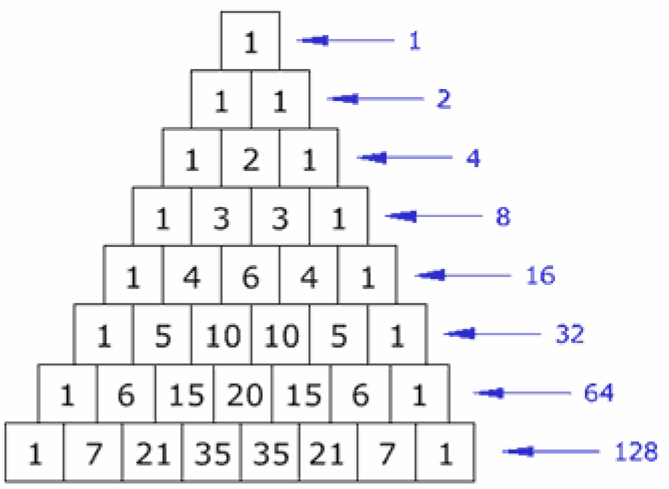
```
RmvVwls
```

### Program #2: Java example program to remove all vowels from a String by taking input from user



```
package inheritanceInterviewPrograms;

import java.util.Scanner;

public class RemoveVowels {

    /**
     * @www.instanceofjava.com
     * @String interview programs asked in interviews
     * @Remove vowels from a string in java
     */
    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter a String");

        String str = in.nextLine();

        String resustr = str.replaceAll("[aeiouAEIOU]", "");
        System.out.println(resustr);

    }

}
```

- Pascals triangle means arranging numbers in pascal triangle format.
- First row starts with number 1.
- Here is the pascal triangle with 8 rows.



1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1

```
package interviewprograms.instanceofjava;

import java.util.Scanner;
/*
 * www.instanceofjava.com
 */
public class PasclasTriangleProgram {

public static void main(String args[]){

Scanner in = new Scanner(System.in);
System.out.println("Enter number of rows ");

int rows= in.nextInt();

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

       int number = 1;

  System.out.format("%"+(rows-i)*2+"s","");

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

    System.out.format("%4d",number);

      number = number * (i - j) / (j + 1);

}

  System.out.println();

}

}
}
```

**Output:**

```
Enter number of rows
8
                 1
               1   1
             1   2   1
           1   3   3   1
         1   4   6   4   1
       1   5  10  10   5   1
     1   6  15  20  15   6   1
   1   7  21  35  35  21   7   1

```

**Print 1 to 10 without using recursion in java?**

```
package com.instanceofjavaTutorial;
class PrintDemo{

public static void recursivefun(int n)
{

  if(n <= 10) {

       System.out.println(n);
         recursivefun(n+1);   }
}

public static void main(String args[])
{

recursivefun(1);

 }

}
```

Output:

```
   1
   2
   3
   4
   5
   6
   7
   8
   9
   10
```

**Write a Basic java example  program to find area of circle**

```
package com.BasicJavaProgramsExamples;
import java.util.Scanner;
public Class AreaOfCirle{


public static void main(String args[]) {

int radius = 0;
Scanner in= new Scanner(System.in);
System.out.println("Please enter radius of a circle");

  radius=in.nextInt();

/*
 * where r is a radius of a circle then Area of a circle is
 *Area= pi * r * r
 *
 */

  double area=Math.PI* radius * radius;

  System.out.println("Area of the circle ="+area);

}
}
```

**Output:**

```
Please enter radius of a circle
23
Area of the circle =1661.9025137490005
```

**Program to print prime numbers in java**

```
   public class primenumbers {
```

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

int num=50;
int count=0;

for(int i=2;i<=num;i++){

count=0;

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

if(i%j==0){
count++;
break;
}

}
 if(count==0){
            System.out.println(i);
                    }

}

}

}
```

**Output:**

```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

**Add two matrices**

```
package com.instanceofjava;

import java.util.Scanner;

class Add2Matrix
{

  public static void main(String args[])
  {

    int rows, cols, c, d;

    Scanner in = new Scanner(System.in);

    System.out.println("Please Enter number of rows and columns");

    rows = in.nextInt();
    cols  = in.nextInt();

    int first[][] = new int[rows][cols];
    int second[][] = new int[rows][cols];
    int sum[][] = new int[rows][cols];

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

    for (  c = 0 ; c < rows ; c++ )
      for ( d = 0 ; d < cols ; d++ )
        first[c][d] = in.nextInt();

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

    for ( c = 0 ; c < rows ; c++ )
      for ( d = 0 ; d < cols ; d++ )
        second[c][d] = in.nextInt();

    for ( c = 0 ; c < rows ; c++ )
      for ( d = 0 ; d < cols ; d++ )
         sum[c][d] = first[c][d] + second[c][d];  //replace '+' with '-' to subtract matrices

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

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

  }
```

**Output:**

```
Please Enter number of rows and columns

3
3

Please Enter elements of first matrix

1 1 1
1 1 1
1 1 1

Please Enter elements of second matrix

2 2 2
2 2 2
2 2 2

Sum of entered matrices:-

3    3    3
3    3    3
3    3    3
```

**Program to check number is even or odd by using division "/" operator**

```
package instanceofjava;
import java.util.Scanner;

public class EvenorOdd {

public static void main(String []args )    {

   int number;
   Scanner in= new Scanner(System.in);

   System.out.println("Enter a number to check even or odd");
   number=in.nextInt();

   if((number / 2)*2==number){
      System.out.println(+number+" is Even number");
   }else{
      System.out.println(+number+" is Odd Number");
   }

}
}
```

### Program to check number is even or odd by using "&" operator

```
package instanceofjava;
import java.util.Scanner;

public class EvenorOdd {

public static void main(String []args )    {

    int number;
    Scanner in= new Scanner(System.in);

    System.out.println("Enter a number to check even or odd");
    number=in.nextInt();
 // using the concept any number with & gives 0 then it is even number
    if((number & 1)==0){
        System.out.println(+number+" is Even number");
    }else{
        System.out.println(+number+" is Odd Number");
    }

}
}
```

### Program: Write a program to find perfect number or not.

Description:
A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors. The first perfect number is 6, because 1, 2 and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors:

$$( 1 + 2 + 3 + 6 ) / 2 = 6.$$

```
package com.java2novice.algos;

public class IsPerfectNumber {

    public boolean isPerfectNumber(int number){

        int temp = 0;
        for(int i=1;i<=number/2;i++){
            if(number%i == 0){
                temp += i;
            }
        }
        if(temp == number){
            System.out.println("It is a perfect number");
            return true;
        } else {
            System.out.println("It is not a perfect number");
            return false;
        }
    }

    public static void main(String a[]){
        IsPerfectNumber ipn = new IsPerfectNumber();
        System.out.println("Is perfect number: "+ipn.isPerfectNumber(28));
    }
}
```

Output:

```
It is a perfect number 28 Is perfect number: true
```

**Program: Write a program to reverse a string using recursive algorithm. - See more at:**

Description:
Write a program to reverse a string using recursive methods.
You should not use any string reverse methods to do this.

```
package com.java2novice.algos;

public class StringRecursiveReversal {

    String reverse = "";

    public String reverseString(String str){

        if(str.length() == 1){
            return str;
        } else {
            reverse += str.charAt(str.length()-1)
                    +reverseString(str.substring(0,str.length()-1));
            return reverse;
        }
    }

    public static void main(String a[]){
        StringRecursiveReversal srr = new StringRecursiveReversal();
        System.out.println("Result: "+srr.reverseString("Java"));
    }
}
 Output:
Result: avaJ
```

**Write a program to reverse a number using numeric operations. Below example shows how to reverse a number using numeric operations.**

```
public class NumberReverse {

    public int reverseNumber(int number){

        int reverse = 0;
        while(number != 0){
            reverse = (reverse*10)+(number%10);
            number = number/10;
        }
        return reverse;
    }

    public static void main(String a[]){
        NumberReverse nr = new NumberReverse();
        System.out.println("Result: "+nr.reverseNumber(178));
    }
}

Output:
Result:871
```

**Program: Write a program to convert decimal number to binary format.**

Description:

Write a program to convert decimal number to binary format using numeric operations. Below example shows how to convert decimal number to binary format using numeric operations.

```
 package com.java2novice.algos;

public class DecimalToBinary {
    public void printBinaryFormat(int number){
        int binary[] = new int[25];
        int index = 0;
        while(number > 0){
            binary[index++] = number%2;
            number = number/2;
        }
        for(int i = index-1;i >= 0;i--){
            System.out.print(binary[i]);
        }
    }

    public static void main(String a[]){
        DecimalToBinary dtb = new DecimalToBinary();
        dtb.printBinaryFormat(25);
    }
}
```

Output:

```
11001
```

**Program: Write a program to find top two maximum numbers in a array**

Description:
Write a program to find top two maximum numbers in the given array. You should not use any sorting functions. You should iterate the array only once. You should not use any kind of collections in java.

```
package com.java2novice.algos;

public class TwoMaxNumbers {

    public void printTwoMaxNumbers(int[] nums){
        int maxOne = 0;
        int maxTwo = 0;
        for(int n:nums){
            if(maxOne < n){
                maxTwo = maxOne;
                maxOne =n;
            } else if(maxTwo < n){
                maxTwo = n;
            }
        }
        System.out.println("First Max Number: "+maxOne);
        System.out.println("Second Max Number: "+maxTwo);
    }

    public static void main(String a[]){
        int num[] = {5,34,78,2,45,1,99,23};
        TwoMaxNumbers tmn = new TwoMaxNumbers();
        tmn.printTwoMaxNumbers(num);
    }
}
```

Output:

```
First Max Number: 99
Second Max Number: 78
```

Program: How to swap two numbers without using temporary variable?

```
package com.java2novice.algos;

public class MySwapingTwoNumbers {

    public static void main(String a[]){
        int x = 10;
        int y = 20;
        System.out.println("Before swap:");
        System.out.println("x value: "+x);
        System.out.println("y value: "+y);
        x = x+y;
        y=x-y;
        x=x-y;
        System.out.println("After swap:");
        System.out.println("x value: "+x);
        System.out.println("y value: "+y);
    }
}
```

Output:

```
Before swap:
x value: 10
y value: 20
After swap:
x value: 20
y value: 10
```

Program: Write a program to convert binary to decimal number.

```
package com.java2novice.algos;

public class BinaryToDecimal {

    public int getDecimalFromBinary(int binary){

        int decimal = 0;
        int power = 0;
        while(true){
            if(binary == 0){
                break;
            } else {
                int tmp = binary%10;
                decimal += tmp*Math.pow(2, power);
                binary = binary/10;
                power++;
            }
        }
        return decimal;
    }

    public static void main(String a[]){
        BinaryToDecimal bd = new BinaryToDecimal();
        System.out.println("11 ===> "+bd.getDecimalFromBinary(11));
        System.out.println("110 ===> "+bd.getDecimalFromBinary(110));
        System.out.println("100110 ===> "+bd.getDecimalFromBinary(100110));
    }
}
```

```
Output:
11 ===> 3
110 ===> 6
100110 ===> 38
```

**Program: Write a program to find the sum of the first 1000 prime numbers.**

```
package com.primesum;

public class Main {

    public static void main(String args[]){

        int number = 2;
        int count = 0;
        long sum = 0;
        while(count < 1000){
            if(isPrimeNumber(number)){
                sum += number;
                count++;
            }
            number++;
        }
        System.out.println(sum);
    }

    private static boolean isPrimeNumber(int number){

        for(int i=2; i<=number/2; i++){
            if(number % i == 0){
                return false;
            }
        }
        return true;
    }
}
```

Output:

```
3682913
```

**Program: Write a program to check the given number is a prime number or not?**

Description:

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. A natural number greater than 1 that is not a prime number is called a composite number. For example, 5 is prime, as only 1 and 5 divide it, whereas 6 is composite, since it has the divisors 2 and 3 in addition to 1 and 6. The fundamental theorem of arithmetic establishes the central role of primes in number theory: any integer greater than 1 can be expressed as a product of primes that is unique up to ordering. This theorem requires excluding 1 as a prime.

```
 package com.java2novice.algos;

public class MyPrimeNumCheck {

    public boolean isPrimeNumber(int number){

        for(int i=2; i<=number/2; i++){
            if(number % i == 0){
                return false;
            }
        }
        return true;
    }

    public static void main(String a[]){
        MyPrimeNumCheck mpc = new MyPrimeNumCheck();
        System.out.println("Is 17 prime number? "+mpc.isPrimeNumber(17));
        System.out.println("Is 19 prime number? "+mpc.isPrimeNumber(19));
        System.out.println("Is 15 prime number? "+mpc.isPrimeNumber(15));
    }
}
```

Output:

```
Is 17 prime number? true
Is 19 prime number? true
Is 15 prime number? false
```

**Program: Write a program to find sum of each digit in the given number using recursion. -**

```
package com.java2novice.algos;

public class MyNumberSumRec {

   int sum = 0;

   public int getNumberSum(int number){

      if(number == 0){
         return sum;
      } else {
         sum += (number%10);
         getNumberSum(number/10);
      }
      return sum;
   }

   public static void main(String a[]){
      MyNumberSumRec mns = new MyNumberSumRec();
      System.out.println("Sum is: "+mns.getNumberSum(223));
   }
}
```

**Java example program to demonstrate super call execution from sub class constructor to super class constructor**

```
package com.superkeywordinjava;
 public Class SuperDemo{

SuperDemo(){
System.out.println("Inside super class constructor");
}

}
```

```
package com.superkeywordinjava;
public Class Subdemo extends SuperDemo{

Subdemo(){
System.out.println("Inside sub class constructor");
}

public static void main (String args[]) {
 Subdemo obj= new Subdemo();


}
}
```

**Output:**

```
Inside super class constructor
Inside sub class constructor
```

**Java interview programming question on this keyword.**

```
package thiskeywordinterviewprograms.java;
public class ThisDemo {

    int a;
    int b;

ThisDemo(int a, int b){

    this.a=a;
    this.b=b;

}

public static void main(String[] args) {

    ThisDemo obj = new ThisDemo(10, 20);

     System.out.println(obj.a);
     System.out.println(obj.b);

}

}
```

o/p

```
10
20
```

Super keyword java programs for interview

```
package com.superkeywordinjava;
```

```
 public Class SuperDemo{

int a,b;

}
package com.superkeywordinjava;
public Class Subdemo extends SuperDemo{
int a,b;
void disply(){

super.a=10;
super.b=20;

System.out.println(a);
System.out.println(b);
System.out.println(super.a);
System.out.println(super.b);

}

public static void main (String args[]) {
 Subdemo obj= new Subdemo();

obj.a=1;
obj.b=2;

obj.disply();



}
}
```

o/p

```
1
2
10
20
```

```
/ Java program for implementation of Selection Sort
class SelectionSort
{
   void sort(int arr[])
```

```
    {
        int n = arr.length;

        // One by one move boundary of unsorted subarray
        for (int i = 0; i < n-1; i++)
        {
            // Find the minimum element in unsorted array
            int min_idx = i;
            for (int j = i+1; j < n; j++)
                if (arr[j] < arr[min_idx])
                    min_idx = j;

            // Swap the found minimum element with the first
            // element
            int temp = arr[min_idx];
            arr[min_idx] = arr[i];
            arr[i] = temp;
        }
    }

    // Prints the array
    void printArray(int arr[])
    {
        int n = arr.length;
        for (int i=0; i<n; ++i)
            System.out.print(arr[i]+" ");
        System.out.println();
    }

    // Driver code to test above
    public static void main(String args[])
    {
        SelectionSort ob = new SelectionSort();
        int arr[] = {64,25,12,22,11};
        ob.sort(arr);
        System.out.println("Sorted array");
        ob.printArray(arr);
    }
}
```

Output:

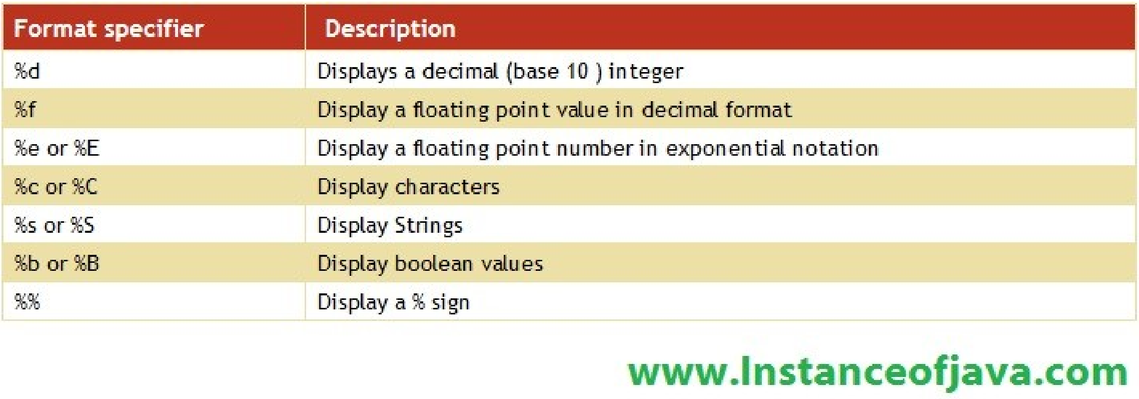
```
Sorted array:
11 12 22 25 64
```

## Format text using printf() method in java

```
package printfinjava;
/**
 * How to format text using java printf() method
 * @author www.instanceofjava.com
 */

public class PrintfMethod {

    public static void main(String[] args) {
        String str="java printf double";

        System.out.println ("String is "+str);
        System.out.printf ("String is %s", str);

    }

}
```

**Output:**

```
String is java printf double
String is java printf double
```



| Format specifier | Description |
|---|---|
| %d | Displays a decimal (base 10 ) integer |
| %f | Display a floating point value in decimal format |
| %e or %E | Display a floating point number in exponential notation |
| %c or %C | Display characters |
| %s or %S | Display Strings |
| %b or %B | Display boolean values |
| %% | Display a % sign |

www.Instanceofjava.com

## Array of objects example program

```
package basics;

public class Arrayelements {

    public static void main(String[] args) {

        int[] a= new int[2];
        a[0]=1;
        a[1]=2;

        System.out.println(a[0]);
        System.out.println(a[1]);
        int[] var= {1,2,3,4,5};


        for (int i = 0; i < var.length; i++) {
           System.out.println(var[i]);
                                                               }
   int[][] array=new int[2][2];

        array[0][0]=1;
        array[0][1]=2;
        array[1][0]=3;
        array[1][1]=4;

    for(int i=0; i<array.length; i++) {
                for(int j=0; j<array[1].length; j++)
                      System.out.print(array[i][j] + " ");
             System.out.println();

                }
      }

}
```

o/p

```
1
2
1
2
3
4
5
1 2
3 4
```