**1. How do you reverse a string in Java?**

public class StringPrograms {

public static void main(String[] args) {

String str = "123";

System.out.println(reverse(str));

}

public static String reverse(String in) {

if (in == null)

throw new IllegalArgumentException("Null is not valid input");

StringBuilder out = new StringBuilder();

char[] chars = in.toCharArray();

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

out.append(chars[i]);

return out.toString();

}

}

## Write a Java program to check if a vowel is present in a string.

public class StringContainsVowels {

public static void main(String[] args) {

System.out.println(stringContainsVowels("Hello")); // true

System.out.println(stringContainsVowels("TV")); // false

}

public static boolean stringContainsVowels(String input) {

return input.toLowerCase().matches(".\*[aeiou].\*");

}

}

## How do you check whether a string is a palindrome in Java?

boolean checkPalindromeString(String input) {

boolean result = true;

int length = input.length();

for (int i = 0; i < length/2; i++) {

if (input.charAt(i) != input.charAt(length - i - 1)) {

result = false;

break;

}

}

return result;

}

## How do you remove spaces from a string in Java?

String removeWhiteSpaces(String input) {

StringBuilder output = new StringBuilder();

char[] charArray = input.toCharArray();

for (char c : charArray) {

if (!Character.isWhitespace(c))

output.append(c);

}

return output.toString();

}

## How do you sort an array in Java?

The Arrays utility class has many overloaded sort() methods to sort primitive and to object arrays. If you are sorting a primitive array in the natural order, then you can use the Arrays.sort() method, as shown in the following example:

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

Arrays.sort(array);

System.out.println(Arrays.toString(array));

## How can you find the factorial of an integer in Java?

The following example code shows how to use recursion to find the factorial of an integer:

public static long factorial(long n) {

if (n == 1)

return 1;

else

return (n \* factorial(n - 1));

}

**How do you implement a binary search in Java?**

The array elements must be sorted to implement binary search. The binary search algorithm is based on the following conditions:

* If the key is less than the middle element, then you now need to search only in the first half of the array.
* If the key is greater than the middle element, then you need to search only in the second half of the array.
* If the key is equal to the middle element in the array, then the search ends.
* Finally, if the key is not found in the whole array, then it should return -1. This indicates that the element is not present.

The following example code implements a binary search:

public static int binarySearch(int arr[], int low, int high, int key) {

int mid = (low + high) / 2;

while (low <= high) {

if (arr[mid] < key) {

low = mid + 1;

} else if (arr[mid] == key) {

return mid;

} else {

high = mid - 1;

}

mid = (low + high) / 2;

}

if (low > high) {

return -1;

}

return -1;

}

### Pyramid Pattern of Numbers

If you look at the first pattern, every row contains the same number printed the same number of times. However, every row has leading white spaces whose count is “rows-i”. Let’s look at the program to print this pattern.

package com.journaldev.patterns.pyramid;

import java.util.Scanner;

public class PyramidPattern {

private static void printPattern1(int rows) {

// for loop for the rows

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

// white spaces in the front of the numbers

int numberOfWhiteSpaces = rows - i;

//print leading white spaces

printString(" ", numberOfWhiteSpaces);

//print numbers

printString(i + " ", i);

//move to next line

System.out.println("");

}

}

//utility function to print string given times

private static void printString(String s, int times) {

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

System.out.print(s);

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Please enter the rows to print:");

int rows = scanner.nextInt();

// System.out.println("Rows = "+rows);

scanner.close();

System.out.println("Printing Pattern 1\n");

printPattern1(rows);

}

}

Notice that I have created a utility function for common string printing task. If you can divide your program into short reusable functions, then it shows that you are not only looking to write the program but also want to make sure of its quality and reusability. When we run above program, we get the following output.

Please enter the rows to print:

9

Printing Pattern 1

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

6 6 6 6 6 6

7 7 7 7 7 7 7

8 8 8 8 8 8 8 8

9 9 9 9 9 9 9 9 9

## Write Java program that checks if two arrays contain the same elements.

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public class ArraySameElements {

public static void main(String[] args) {

Integer[] a1 = {1,2,3,2,1};

Integer[] a2 = {1,2,3};

Integer[] a3 = {1,2,3,4};

System.out.println(sameElements(a1, a2));

System.out.println(sameElements(a1, a3));

}

static boolean sameElements(Object[] array1, Object[] array2) {

Set<Object> uniqueElements1 = new HashSet<>(Arrays.asList(array1));

Set<Object> uniqueElements2 = new HashSet<>(Arrays.asList(array2));

// if size is different, means there will be a mismatch

if (uniqueElements1.size() != uniqueElements2.size()) return false;

for (Object obj : uniqueElements1) {

// element not present in both?

if (!uniqueElements2.contains(obj)) return false;

}

return true;

}

}

Output

true

false

## How do you find the second largest number in an array in Java?

private static int findSecondHighest(int[] array) {

int highest = Integer.MIN\_VALUE;

int secondHighest = Integer.MIN\_VALUE;

for (int i : array) {

if (i > highest) {

secondHighest = highest;

highest = i;

} else if (i > secondHighest) {

secondHighest = i;

}

}

return secondHighest;

}

## How can you find a string in a text file in Java?

The following example code shows how to use the Scanner class to read the file contents line by line and then use the String contains() method to check if the string is present in the file:

boolean findStringInFile(String filePath, String str) throws FileNotFoundException {

File file = new File(filePath);

Scanner scanner = new Scanner(file);

// read the file line by line

while (scanner.hasNextLine()) {

String line = scanner.nextLine();

if (line.contains(str)) {

scanner.close();

return true;

}

}

scanner.close();

return false;

}

## How do you print a date in specific format in Java?

The following example code shows how to use the SimpleDateFormat class to format the date string:

String pattern = "MM-dd-yyyy";

SimpleDateFormat simpleDateFormat = new SimpleDateFormat(pattern);

String date = simpleDateFormat.format(new Date());

System.out.println(date); // 06-23-2020

## How do you merge two lists in Java?

The following example code shows how to use the addAll() method to merge multiple lists in Java:

List<String> list1 = new ArrayList<>();

list1.add("1");

List<String> list2 = new ArrayList<>();

list2.add("2");

List<String> mergedList = new ArrayList<>(list1);

mergedList.addAll(list2);

System.out.println(mergedList); // [1, 2]

## How do you remove all occurrences of a given character from an input string in Java?

The String class doesn’t have a method to remove characters. The following example code shows how to use the replace() method to create a new string without the given character:

String str1 = "abcdABCDabcdABCD";

str1 = str1.replace("a", "");

System.out.println(str1); // bcdABCDbcdABCD

## How do you get distinct characters and their count in a string in Java?

You can create the character array from the string. Then iterate over it and create a HashMap with the character as key and their count as value. The following example code shows how to extract and count the characters of a string:

String str1 = "abcdABCDabcd";

char[] chars = str1.toCharArray();

Map<Character, Integer> charsCount = new HashMap<>();

for (char c : chars) {

if (charsCount.containsKey(c)) {

charsCount.put(c, charsCount.get(c) + 1);

} else

charsCount.put(c, 1);

}

System.out.println(charsCount); // {a=2, A=1, b=2, B=1, c=2, C=1, d=2, D=1}

## Can you write some code to showcase inheritance in Java?

The following example code shows how to use the extends keyword to create a subclass of the class Animal. The new class Cat inherits the variable from the Animal class and adds more code that only belongs to the Cat class.

class Animal {

String color;

}

class Cat extends Animal {

void meow() {

System.out.println("Meow");

}

}

## How do you illustrate a try catch example in Java?

The following example code shows an example of try-catch:

try {

FileInputStream fis = new FileInputStream("test.txt");

} catch(FileNotFoundException e) {

e.printStackTrace();

}

## How do you use the forEach() method in Java?

The forEach() method provides a shortcut to perform an action on all the elements of an iterable. The following example code shows how to iterate over the list elements and print them:

List<String> list = new ArrayList<>();

Iterator<String> it = list.iterator();

while (it.hasNext()) {

System.out.println(it.next());

}

You can use the forEach() method with a lambda expression to reduce the code size, as shown in the following example code:

List<String> list = new ArrayList<>();

list.forEach(System.out::print);

## Show examples of overloading and overriding in Java.

When a class has two or more methods with the same name, they are called overloaded methods. The following example code shows as overloaded method called print:

class Foo {

void print(String s) {

System.out.println(s);

}

void print(String s, int count) {

while (count > 0) {

System.out.println(s);

count--;

}

}

}

When a superclass method is also implemented in the child class, it’s called overriding. The following example code shows how to annotate the printname() method that’s implemented in both classes:

class Base {

void printName() {

System.out.println("Base Class");

}

}

class Child extends Base {

@Override

void printName() {

System.out.println("Child Class");

}

}

**Anagrams program in java**

|  |  |  |  |
| --- | --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 Remove Vowels from String in Java The question is, *write a Java program to remove all vowels from a string. The string must be received by user at run-time.* The program given below is its answer:  import java.util.Scanner;  public class CodesCracker  {  public static void main(String[] args)  {  String str, strRes, vowels;  char ch;  int i, count, k;  Scanner scan = new Scanner(System.in);    System.out.print("Enter the String: ");  str = scan.nextLine();    strRes="";  vowels = "aeiouAEIOU";  for(i=0; i<str.length(); i++)  {  count=0;  ch = str.charAt(i);  for(k=0; k<vowels.length(); k++)  {  if(ch==vowels.charAt(k))  count++;  }  if(count==0)  strRes = strRes + ch;  }    System.out.println("\nString without Vowels = " +strRes);  }  }   |  | | --- | | Print Star Pattern in Java - Pattern No.1 The question is, *write a Java program to print star pattern.* The program given below is its answer:  public class CodesCracker  {  public static void main(String[] args)  {  int i, j;  for(i=0; i<5; i++)  {  for(j=0; j<=i; j++)  {  System.out.print("\* ");  }  System.out.print("\n");  }  }  } | | **Linear Search in Java**  The question is, *write a Java program to perform linear search.* The program given below is its answer:  import java.util.Scanner;  public class CodesCracker  {  public static void main(String[] args)  {  int i, num, pos=0;  int[] arr = new int[10];  Scanner s = new Scanner(System.in);    System.out.print("Enter 10 Elements: ");  for(i=0; i<10; i++)  arr[i] = s.nextInt();    System.out.print("Enter an Element to Search: ");  num = s.nextInt();    for(i=0; i<10; i++)  {  if(num==arr[i])  {  pos = i+1;  break;  }  }  if(pos==0)  System.out.println("\nThe element not found!");  else  System.out.println("\nThe element found at position: " +pos);  }  }  The snapshot given below shows the sample run of above program, with user input **1, 3, 4, 6, 7, 8, 10, 12, 15, 18** as 10 elements and **10** as element to search:  linear search program in java | | import java.util.\*;  public class Main {      public static void main(String[] args) {          Scanner sc= new Scanner(System.in);          System.out.print("Please give First String : ");          String str1= sc.nextLine();          Scanner sc1= new Scanner(System.in);          System.out.print("Please give Second String : ");          String str2= sc.nextLine();          if(anagramCheck(str1,str2)){              System.out.print("String are anagram");          }else{              System.out.print("String are not anagram");          }      }      public static boolean anagramCheck(String str1, String str2){          boolean status = false;          if (str1.length() != str2.length()) {              status = false;          } else {              char[] arr1 = str1.toLowerCase().toCharArray();              char[] arr2 = str2.toLowerCase().toCharArray();              Arrays.sort(arr1);              Arrays.sort(arr2);              status = Arrays.equals(arr1, arr2);          }          return status;       }  } |

## Remove Vowels from String in Java

The question is, *write a Java program to remove all vowels from a string. The string must be received by user at run-time.* The program given below is its answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String str, strRes, vowels;

char ch;

int i, count, k;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the String: ");

str = scan.nextLine();

strRes="";

vowels = "aeiouAEIOU";

for(i=0; i<str.length(); i++)

{

count=0;

ch = str.charAt(i);

for(k=0; k<vowels.length(); k++)

{

if(ch==vowels.charAt(k))

count++;

}

if(count==0)

strRes = strRes + ch;

}

System.out.println("\nString without Vowels = " +strRes);

}

}

**Remove Word from String in Java - Character by Character**

The question is, *write a Java program to remove specified word from a given string. Both, string and word must be received by user at run-time.* The program given below is its answer. This program uses character by character way to remove a given word from a given string:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String str, word;

int strLen, wordLen, i, temp, j, k=0;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the String: ");

str = scan.nextLine();

System.out.print("Enter the Word to Remove: ");

word = scan.nextLine();

strLen = str.length();

wordLen = word.length();

char[] strChars = new char[strLen];

char[] wordChars = new char[wordLen];

strChars = str.toCharArray();

wordChars = word.toCharArray();

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

{

temp = i;

for(j=0; j<wordLen; j++)

{

if(strChars[i]==wordChars[j])

i++;

}

k = i-temp;

if(k==wordLen)

{

for(j=temp; j<(strLen-wordLen); j++)

strChars[j] = strChars[j+wordLen];

strLen = strLen-wordLen;

}

}

System.out.print("\nThe new string is: ");

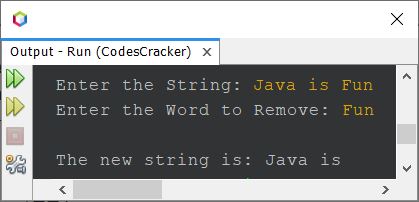
for(i=0; i<strLen; i++)

System.out.print(strChars[i]);

}

}

The sample run of above program with user input **Java is Fun** as string and **Fun** as word to delete, is shown in the snapshot given below:



**Concatenate Two Strings in Java using concat()**

The question is, *write a Java program to concatenate or append the second string into the first. Both the string must be received by user at run-time.* The program given below is its answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String a, b;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the First String: ");

a = scan.nextLine();

System.out.print("Enter the Second String: ");

b = scan.nextLine();

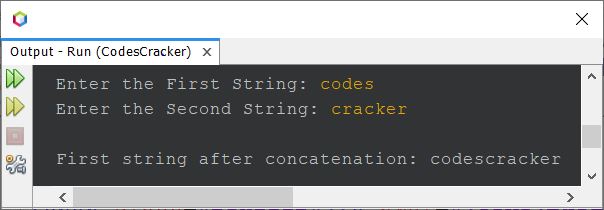
a = a.concat(b);

System.out.println("\nFirst string after concatenation: " +a);

}

}

The snapshot given below shows the sample run of above Java program on string concatenation, with user input **codes** as first and **cracker** as second string:



# Java Program to Sort Strings in Alphabetical Order

This article is created to cover a program in Java, that sorts strings, entered by user at run-time of the program, in alphabetical order.

The question is, *write a Java program to sort string or names in alphabetical order. Strings must be received by user at run-time.* The program given below is its answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String[] names = new String[5];

String temp;

int i, j;

Scanner scan = new Scanner(System.in);

System.out.print("Enter 5 Names: ");

for(i=0; i<5; i++)

names[i] = scan.nextLine();

// sorting names in alphabetical order

for(i=0; i<5; i++)

{

for(j=1; j<5; j++)

{

if(names[j-1].compareTo(names[j])>0)

{

temp=names[j-1];

names[j-1]=names[j];

names[j]=temp;

}

}

}

System.out.println("\nNames in Alphabetical Order:");

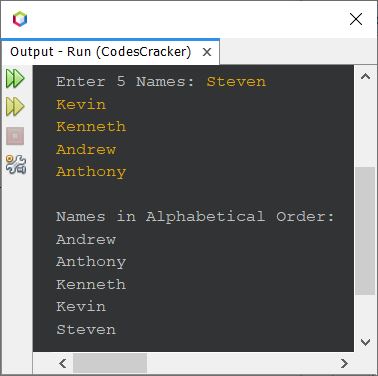
for(i=0;i<5;i++)

System.out.println(names[i]);

}

}

The snapshot given below shows the sample run of above program with user input **Steven**, **Kevin**, **Kenneth**, **Andrew**, and **Anthony** as five names to sort and print names in alphabetical order:



**Find Occurrence of a Word in String - Basic Version**

The question is, *write a Java program to find and print the occurrence of a word in a string. Both word and string must be received by user at run-time of the program.* The program given below is its answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String str, word;

int wordsLen, i, count=0;

Scanner s = new Scanner(System.in);

System.out.print("Enter the String: ");

str = s.nextLine();

System.out.print("\nEnter a Word to Find its Occurrence: ");

word = s.next();

String words[] = str.split(" ");

wordsLen = words.length;

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

{

if(word.equals(words[i]))

count++;

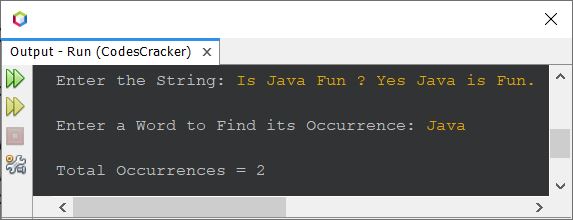
}

System.out.println("\nTotal Occurrences = " +count);

}

}

The snapshot given below shows the sample run of above Java program on finding the occurrence of a given word in given string, with user input **Is Java Fun ? Yes Java is Fun.** as string and **Java** as word:



# Java Program to Count the Occurrence of Each Character in String

This article is created to cover a program in Java, to count the occurrence of each character in a string entered by user at run-time of the program.

The question is, *write a Java program to count and print the occurrence of each characters available in a string. The string must be received by user at run-time.* The program given below is its answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

int len, i, k, count, prevLen, arri=0, arrLen=0;

String str;

char ch;

Scanner s = new Scanner(System.in);

System.out.print("Enter the String: ");

str = s.nextLine();

len = str.length();

char[][] arr = new char[len][2];

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

{

ch = str.charAt(i);

count = 0;

for(k=0; k<arrLen; k++)

{

if(ch==arr[k][0])

{

prevLen = (arr[k][1]);

prevLen = prevLen+1;

arr[k][1] = (char)prevLen;

count++;

break;

}

}

if(count==0)

{

arr[arri][0] = ch;

arr[arri][1] = '1';

arrLen++;

arri++;

}

}

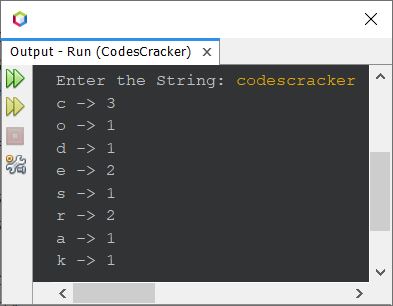
for(i=0; i<arrLen; i++)

System.out.println(arr[i][0]+ " -> " +arr[i][1]);

}

}

Here is its sample run with user input **codescracker** as string to count occurrence of each character available in this given string:



# Java Program to Count Number of Repeated Characters in a String

This article is created to cover a program in Java that count and prints the number of repeated or duplicate characters available in a given string. For example, if given string is **Java Programming**, then the output will be **4**. Because the character **a**, **r**, **g**, and **m** are available for more than one times.

## Count Number of Repeated Characters in String - Basic Version

The question is, *write a Java program to count the number of repeated characters in a string. The string must be received by user at run-time of the program.* Here is its sample run:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String str;

char ch;

int strLen, i, count, j, k, repChars=0;

Scanner s = new Scanner(System.in);

System.out.print("Enter the String: ");

str = s.nextLine();

strLen = str.length();

char[] arr = new char[strLen];

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

arr[i] = str.charAt(i);

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

{

ch = arr[i];

count = 0;

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

{

if(ch==arr[j])

{

count++;

for(k=j; k<(strLen-1); k++)

arr[k] = arr[k+1];

strLen--;

j--;

}

}

if(count>0)

repChars++;

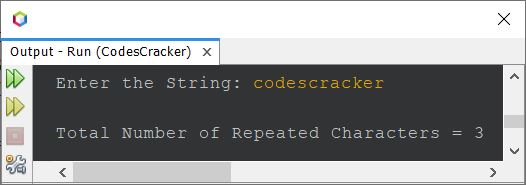
}

System.out.println("\nTotal Number of Repeated Characters = " +repChars);

}

}

The snapshot given below shows the sample run of above program with user input **codescracker** as string to count and print the total number of duplicate characters available in it:



That is, the character **c**, **e**, and **r** are the three characters available in repeated number of times. Therefore the output was 3.

# Java Program to Count Number of Words in a String

This article is created to cover a program in Java to count the total number of words available in a string entered by user.

## Count Words in String - Basic Version

The question is, *write a Java program to count the number of words in a string. The string must be entered by user at run-time of the program.* The program given below is the answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String str;

int totalWords;

Scanner s = new Scanner(System.in);

System.out.print("Enter the String: ");

str = s.nextLine();

String words[] = str.split(" ");

totalWords = words.length;

System.out.println("\nTotal Number of Words = " +totalWords);

}

}

# Java Program to Check Anagram or Not

This article is created to cover a program in Java that checks whether two strings entered by user at run-time of the program, are anagram or not.

Two strings are anagram, if one string can be re-arranged to form the other. For example, **listen** and **silent** are anagram strings. Because **listen** can be re-arranged to form **silent** and vice-versa.

Now the question is, *write a Java program to check whether strings are anagram or not.* The program given below is its answer:

import java.util.Scanner;

public class CodesCracker

{

public static void main(String[] args)

{

String strOne, strTwo;

int lenOne, lenTwo, i, j, found=0, not\_found=0;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the First String: ");

strOne = scan.nextLine();

System.out.print("Enter the Second String: ");

strTwo = scan.nextLine();

lenOne = strOne.length();

lenTwo = strTwo.length();

if(lenOne == lenTwo)

{

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

{

found = 0;

for(j=0; j<lenOne; j++)

{

if(strOne.charAt(i) == strTwo.charAt(j))

{

found = 1;

break;

}

}

if(found == 0)

{

not\_found = 1;

break;

}

}

if(not\_found == 1)

System.out.println("\nStrings are not Anagram");

else

System.out.println("\nStrings are Anagram");

}

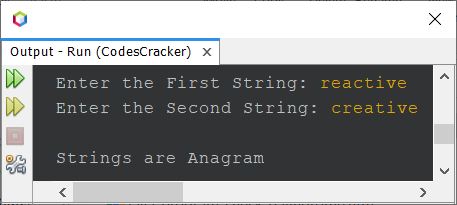
else

System.out.println("\nLength of Strings Mismatched!");

}

}

Here is its sample run with user input **reactive** as first and **creative** as second string:



# Java Program to Generate Random Numbers

This article is created to cover multiple programs in Java that generate and prints random number(s). Here are the list of programs covered in this article:

* Generate and print a random number in Java using **Math.random()**
* Generate and print **n** random numbers in Java
* Generate and print **n** random numbers between 1 to 100 in Java
* Generate and print **n** random numbers between given range in Java
* Generate and print random numbers in Java using **Random** class

## Generate and Print a Random Number in Java

The question is, *write a Java program to generate a random number.* The program given below is its answer:

import java.lang.Math;

public class CodesCracker

{

public static void main(String[] args)

{

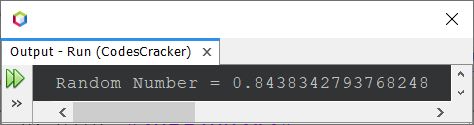
double num = Math.random();

System.out.println("Random Number = " +num);

}

}

The snapshot given below shows the sample output produced by above Java program:



# Java Program to Print Date and Time

This article covers multiple programs in Java that prints the current date and time. Here are the list of programs covered in this article:

* Get and print the complete current date in default format
* Get and print the current date, in specified format
* Get and print the current time, in specified format
* Print today's day
* Print time, day, and date in specified format

## Get and Print Current Date in Java

The question is, *write a Java program to print the current date.* The program given below is its answer. This program prints the complete current date in default format.

import java.util.Date;

public class CodesCracker

{

public static void main(String[] args)

{

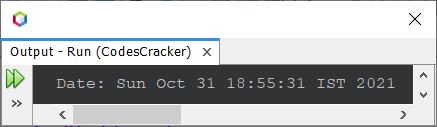
Date d = new Date();

System.out.println("Date: " +d);

}

}

The snapshot given below shows the sample output produced by above program, on printing the current date:



**Java program to delete vowels from string using replaceAll method.**

Here we are going to use replaceAll() method of String class. This method replaces any occurrence of any vowel character(lowercase or uppercase) with "". We are using "[AEIOUaeiou]" regular expression, which matches with any vowel alphabet.

|  |
| --- |
| package com.tcc.java.programs;    import java.util.Scanner;    public class DeleteVowels {      public static void main(String args[]) {          String str, output;          Scanner scanner = new Scanner(System.in);            System.out.println("Enter a String");          str = scanner.nextLine();            output = str.replaceAll("[AEIOUaeiou]",             "");            System.out.println("Output String\n"             + output);      }  } |

**Java program to remove vowel characters from string without using library function.**

|  |
| --- |
| package com.tcc.java.programs;    import java.util.Scanner;    public class DeleteVowelsLoop {      public static void main(String args[]) {          String str, output = "";          int length, i;          Scanner scanner = new Scanner(System.in);            System.out.println("Enter a String");          str = scanner.nextLine();          length = str.length();            // Deleting vowel alphabets from input string          for (i = 0; i < length; i++) {              if ("AEIOUaeiou".indexOf(str.charAt(i)) == -1) {                  output = output + str.charAt(i);              }          }            System.out.println("Output String : " + output);      }  } |

Output

Enter a String

Apple

Output String : ppl

Enter a String

TechCrashCourse

Output String : TchCrshCrs

|  |
| --- |
| / Java Program to Remove Duplicate Elements  // From the Array using extra space    public class Main {        public static int removeduplicates(int a[], int n)      {          if (n == 0 || n == 1) {              return n;          }            // creating another array for only storing          // the unique elements          int[] temp = new int[n];          int j = 0;            for (int i = 0; i < n - 1; i++) {              if (a[i] != a[i + 1]) {                  temp[j++] = a[i];              }          }            temp[j++] = a[n - 1];            // Changing the original array          for (int i = 0; i < j; i++) {              a[i] = temp[i];          }            return j;      }      public static void main(String[] args)      {          int a[] = { 1, 1, 2, 2, 2 };          int n = a.length;            n = removeduplicates(a, n);            // Printing The array elements          for (int i = 0; i < n; i++)              System.out.print(a[i] + " ");      }  } |

**Output**

1 2

# Java Program to Count Words in a Sentence

Here is a Java program to count words in a sentence using split method. To count the number of words in a sentence, we first take a sentence as input from user and store it in a String object. Words in a sentence are separated by space character(" "), hence we can use space as a delimiter to split given sentence into words. To split a string to multiple words separated by spaces, we will call split() method.

**public String[] split(String regex);**

split() method returns an array of Strings, after splitting string based of given regex(delimiters). To fine the count of words in sentence, we will find the length of String array returned by split method.

**Java program to find the count of words in a sentence**

|  |
| --- |
| package com.tcc.java.programs;    import java.util.Scanner;    public class WordCount {      public static void main(String args[]) {          String str;          Scanner scanner = new Scanner(System.in);            System.out.println("Enter a Sentence");          str = scanner.nextLine();            // Printing number of words in given sentence          System.out.println("Number of Words = "              + str.split(" ").length);      }  } |

Output

Enter a Sentence

I Love Java Programming

Number of Words = 4

# Java Program to Reverse Words of a Sentence

Here is a java program to reverse words of a sentence. Given a sentence, we have to reverse the sequence of words in given sentences. Words of the given sentence are separated by one or multiple space characters.

For Example,  
Input Sentence : I love Java Programming  
Output Sentence : Programming Java love I  
  
To split a string to multiple words separated by spaces, we will call split() method.

**public String[] split(String regex);**

split() method returns an array of Strings, after splitting string based of given regex(delimiters).

**Java program to reverse words of a sentence**

|  |
| --- |
| package com.tcc.java.programs;    import java.util.Scanner;    public class ReverseSentence {      public static void main(String[] args) {          String input;          String[] words;          int i;          Scanner scanner = new Scanner(System.in);            System.out.println("Enter a Sentence");          input = scanner.nextLine();          words = input.split(" ");            // Now, Print the sentence in reverse order          System.out.println("Reversed Sentence");          for (i = words.length - 1; i >= 0; i--) {              System.out.print(words[i] + " ");          }      }  } |

Output

Enter a Sentence

I love Java Programming

Reversed Sentence

Programming Java love I

# Java Program to Count Occurrence of Each Character in a String

Here is a Java program to count the occurrence of each character if a string. In this java program, we have to count the frequency of occurrence of each character of a string and then print it on screen.

For Example,

Input String : Apple

A : 1 times

e : 1 times

l : 1 times

p : 2 times

To count the frequency of each alphabet, we will first take a string as input from user. We will use an integer array of length 256 to count the frequency of characters. Initialize frequency array element with zero, which means initially the count of all characters are zero.  
Using a for loop, traverse input string and increment the count of every character of input string. Finally, traverse the frequency array and print the frequency of every character.

**Java program to count each character of a string**

|  |
| --- |
| package com.tcc.java.programs;    import java.util.Scanner;    public class CharacterCount {      public static void main(String args[]) {          String str;          int i, length, counter[] = new int[256];            Scanner scanner = new Scanner(System.in);          System.out.println("Enter a String");          str = scanner.nextLine();            length = str.length();            for (i = 0; i < length; i++) {              counter[(int) str.charAt(i)]++;          }            for (i = 0; i < 256; i++) {              if (counter[i] != 0) {                  System.out.println((char) i + "                    --> " + counter[i]);              }          }      }  } |

Output

Enter a String

APPLE

A --> 1

E --> 1

L --> 1

P --> 2

**Removing Duplicate Elements In Java Array**

To remove the duplicate elements present in an array and get a unique array, we use multiple methods and procedures. The most important ones are given below:

**Method 1**

In this method, we remove the duplicate elements by using a temporary array.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | public class Main{  public static int removeDuplicates(int array[], int n){  if(n==0 || n==1){  return n;  }  int[] temp = new int[n];  int j = 0;  for(int i=0; i<n-1; i++){  if(array[i] != array[i+1]){  temp[j++] = array[i];  }  }  temp[j++] = array[n-1];  //Changing the original array  for(int i=0; i<j; i++){  array[i] = temp[i];  }  return j;  }  public static void main (String[] args) {  int array[] = {18,18,25,25,25,28,28,29};  int length = array.length;  length = removeDuplicates(array, length);  //Printing The array elements  for(int i=0; i<length; i++)  System.out.print(array[i]+" ");    }    } |

We create a temporary array to store the unique elements. The initial array is traversed, and the unique elements are copied to the temporary array. Track of count of the unique element is kept using “j”. The value present in j is then copied from the temporary array to the initial array, after which j is returned.

**Method 2: Removing Duplicate Elements In Java Array**

In this method, a separate index is used.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | public class Main{  public static int removeDuplicates(int array[], int n){  if(n==0 || n==1){  return n;  }  int j = 0;//for next element  for (int i=0; i < n-1; i++){  if (array[i] != array[i+1]){  array[j++] = array[i];  }  }  array[j++] = array[n-1];  return j;  }  public static void main (String[] args) {  int array[] = {18,18,25,25,25,28,28,29};  int length = array.length;  length = removeDuplicates(array, length);  //printing array elements  for(int i=0; i<length; i++)  System.out.print(array[i]+" ");  }  } |

18,25,28,29

**1) Remove Duplicate Element in Array using Temporary Array**

1. public class RemoveDuplicateInArrayExample{
2. public static int removeDuplicateElements(int arr[], int n){
3. if (n==0 || n==1){
4. return n;
5. }
6. int[] temp = new int[n];
7. int j = 0;
8. for (int i=0; i<n-1; i++){
9. if (arr[i] != arr[i+1]){
10. temp[j++] = arr[i];
11. }
12. }
13. temp[j++] = arr[n-1];
14. // Changing original array
15. for (int i=0; i<j; i++){
16. arr[i] = temp[i];
17. }
18. return j;
19. }
21. public static void main (String[] args) {
22. int arr[] = {10,20,20,30,30,40,50,50};
23. int length = arr.length;
24. length = removeDuplicateElements(arr, length);
25. //printing array elements
26. for (int i=0; i<length; i++)
27. System.out.print(arr[i]+" ");
28. }
29. }

[Test it Now](https://compiler.javatpoint.com/opr/test.jsp?filename=RemoveDuplicateInArrayExample)

Output:

10 20 30 40 50

**2) Remove Duplicate Element in Array using separate index**

1. public class RemoveDuplicateInArrayExample2{
2. public static int removeDuplicateElements(int arr[], int n){
3. if (n==0 || n==1){
4. return n;
5. }
6. int j = 0;//for next element
7. for (int i=0; i < n-1; i++){
8. if (arr[i] != arr[i+1]){
9. arr[j++] = arr[i];
10. }
11. }
12. arr[j++] = arr[n-1];
13. return j;
14. }
16. public static void main (String[] args) {
17. int arr[] = {10,20,20,30,30,40,50,50};
18. int length = arr.length;
19. length = removeDuplicateElements(arr, length);
20. //printing array elements
21. for (int i=0; i<length; i++)
22. System.out.print(arr[i]+" ");
23. }
24. }

[Test it Now](https://compiler.javatpoint.com/opr/test.jsp?filename=RemoveDuplicateInArrayExample2)

Output:

10 20 30 40 50

## Remove Duplicate Elements in Unsorted Array

If you have unsorted array, you need to sort it first. To do so, use Arrays.sort(arr) method.

1. mport java.util.Arrays;
2. public class RemoveDuplicateInArrayExample3{
3. public static int removeDuplicateElements(int arr[], int n){
4. if (n==0 || n==1){
5. return n;
6. }
7. int[] temp = new int[n];
8. int j = 0;
9. for (int i=0; i<n-1; i++){
10. if (arr[i] != arr[i+1]){
11. temp[j++] = arr[i];
12. }
13. }
14. temp[j++] = arr[n-1];
15. // Changing original array
16. for (int i=0; i<j; i++){
17. arr[i] = temp[i];
18. }
19. return j;
20. }
22. public static void main (String[] args) {
23. int arr[] = {10,70,30,90,20,20,30,40,70,50};//unsorted array
24. Arrays.sort(arr);//sorting array
25. int length = arr.length;
26. length = removeDuplicateElements(arr, length);
27. //printing array elements
28. for (int i=0; i<length; i++)
29. System.out.print(arr[i]+" ");
30. }
31. }

[Test it Now](https://compiler.javatpoint.com/opr/test.jsp?filename=RemoveDuplicateInArrayExample3)

Output:

10 20 30 40 50 70 90