

Difference between ArrayList and LinkedList

[← prev](#)[next →](#)

ArrayList and LinkedList both implements List interface and maintains insertion order. Both are non synchronized classes.

However, there are many differences between ArrayList and LinkedList classes that are given below.

ArrayList	LinkedList
1) ArrayList internally uses a dynamic array to store the elements.	LinkedList internally uses a doubly linked list to store the elements.
2) Manipulation with ArrayList is slow because it internally uses an array. If any element is removed from the array, all the bits are shifted in memory.	Manipulation with LinkedList is faster than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory.
3) An ArrayList class can act as a list only because it implements List only.	LinkedList class can act as a list and queue both because it implements List and Deque interfaces.
4) ArrayList is better for storing and accessing data.	LinkedList is better for manipulating data.



ArrayList and Vector both implements List interface and maintains insertion order.

However, there are many differences between ArrayList and Vector classes that are given below.

ArrayList	Vector
1) ArrayList is not synchronized .	Vector is synchronized .
2) ArrayList increments 50% of current array size if the number of elements exceeds from its capacity.	Vector increments 100% means doubles the array size if the total number of elements exceeds than its capacity.
3) ArrayList is not a legacy class. It is introduced in JDK 1.2.	Vector is a legacy class.
4) ArrayList is fast because it is non-synchronized.	Vector is slow because it is synchronized, i.e., in a multithreading environment, it holds the other threads in runnable or non-runnable state until current thread releases the lock of the object.
5) ArrayList uses the Iterator interface to traverse the elements.	A Vector can use the Iterator interface or Enumeration interface to traverse the elements.

Difference between Comparable and Comparator

[← prev](#)[next →](#)

Comparable and Comparator both are interfaces and can be used to sort collection elements.

However, there are many differences between Comparable and Comparator interfaces that are given below.

Comparable	Comparator
1) Comparable provides a single sorting sequence . In other words, we can sort the collection on the basis of a single element such as id, name, and price.	The Comparator provides multiple sorting sequences . In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc.
2) Comparable affects the original class , i.e., the actual class is modified.	Comparator doesn't affect the original class , i.e., the actual class is not modified.
3) Comparable provides compareTo() method to sort elements.	Comparator provides compare() method to sort elements.
4) Comparable is present in java.lang package.	A Comparator is present in the java.util package.
5) We can sort the list elements of Comparable type by Collections.sort(List) method.	We can sort the list elements of Comparator type by Collections.sort(List, Comparator) method.

HashMap

Hashtable

1) HashMap is **non synchronized**. It is not-thread safe and can't be shared between many threads without proper synchronization code.

Hashtable is **synchronized**. It is thread-safe and can be shared with many threads.

2) HashMap **allows one null key and multiple null values**.

Hashtable **doesn't allow any null key or value**.

3) HashMap is a **new class introduced in JDK 1.2**.

Hashtable is a **legacy class**.

4) HashMap is **fast**.

Hashtable is **slow**.

5) We can make the HashMap as synchronized by calling this code

```
Map m = Collections.synchronizedMap(hashMap);
```

Hashtable is internally synchronized and can't be unsynchronized.

6) HashMap is **traversed by Iterator**.

Hashtable is **traversed by Enumerator and Iterator**.

7) Iterator in HashMap is **fail-fast**.

Enumerator in Hashtable is **not fail-fast**.

8) HashMap inherits **AbstractMap** class.

Hashtable inherits **Dictionary** class.



Question: 1 Which collection class allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized? (Choose all that apply.)

Your Answer: java.util.ArrayList ✓

Correct Answer: java.util.ArrayList

Description: None

Question: 2 What is the output of the following program ?

```
import java.util.HashMap;
import java.util.Map;
public class MapTest {
    public static void main(String args[]){
        Map m= new HashMap();
        m.put(null, "Test");
        m.put(null,"Fest");
        System.out.println(m);
    }
}
```

Your Answer: {null=Fest} ✓

Correct Answer: {null=Fest}

Description: None

Question: 3 which of the following are false about Collections and Collection ?

Your Answer: Collection is an interface to Set and List 

Correct Answer: Collections is a special type of collection which holds Set of collections

Description: None

Question: 4 If you were to use a List implementation, but not sure which one to, because the requirement is not yet clear. In this case which List implementation will you use ?

Your Answer: Vector 

Correct Answer: ArrayList

Description: None

Question: 5 How do you get immutable object of a collection ? For example if you were to write an API which return a List or a Set or a Map when a method is called, but you also want that you don't want the client of your API to add or delete any object in the returned collection ?

Your Answer: None 

Correct Answer: Use the Collections.unmodifiableXxxx() method with the collection as an argument, which returns an immutable object of specific type.

Description: None

Question: 6 What is the data structure that a Set uses to store its elements ?

Your Answer: Map 

Correct Answer: Map

Description: None

Question: 7 Which of the following implementation will you use if you were to insert elements at any position in the collection ?

Your Answer: LinkedList 

Correct Answer: LinkedList

Description: None

Question: 8 Which implementation of Set would you choose if the you want the iterator of set would give you objects in the order it were inserted ?

Your Answer: LinkedHashSet 

Correct Answer: LinkedHashSet

Description: None

Question: 9 What is the output of the following program ?

```
import java.util.HashSet;
```

```
class Animal {  
    public int i=12;  
  
    public Animal(){  
        i=13;  
    }  
    public String toString(){  
        return "Animal"+i;  
    }  
}  
  
public class Test {  
    public static void main(String [] args) throws Exception {  
        HashSet s= new HashSet();  
        s.add(new Animal());  
        s.add(new Animal());  
        for(Animal a : s){  
            System.out.println(a);  
        }  
    }  
}
```

Your Answer: Runtime error about duplicate elements 

Correct Answer: Animal 13 Animal 13

Description: None

Question: 10 If you were to store objects into an implementation of List which happens only once in the entire life cycle of the product, but reading these objects inside the List implementation is quite high, then which one would you use?

Your Answer: LinkedList 

Correct Answer: ArrayList

Description: None

Question: 11 Is Hashtable synchronized?

Your Answer: Yes 

Correct Answer: Yes

Description: None

Question: 12 Set allows at most one null element ?

Your Answer: False 

Correct Answer: True

Description: None

Question: 13 Is HashMap synchronized?

Your Answer: No 

Correct Answer: No

Question: 13 Is HashMap synchronized?

Your Answer: No ✓

Correct Answer: No

Description: None

Question: 14 Enumeration is an interface helps to iterate collection, but it can't remove any element the collection it is iterating

Your Answer: True ✓

Correct Answer: True

Description: None

Question: 15 TreeSet maintains which order?

Your Answer: Ascending Order ✓

Correct Answer: Ascending Order

Description: None

- Java Interview Question (7)
- OCPJP (7)
- design pattern (7)
- JSON (6)
- homework (6)
- java concurrency tutorial (5)
- Hibernate (4)
- Web Service (4)
- jQuery (4)
- Hibernate interview Question (3)
- Java 5 tutorial (3)
- Struts (3)
- debugging (3)
- thread (3)
- Java Enum (2)
- xml (2)
- garbage collection (1)



Solution 2 - Counting word in String using StringTokenizer

Constructs a string tokenizer for the specified string. The tokenizer uses the default delimiter set, which is " \t\n\r\f": the space character, the tab character, the newline character, the carriage-return character, and the form-feed character. Delimiter characters themselves will not be treated as tokens

```
public static int countWordsUsingStringTokenizer(String sentence) {  
    if (sentence == null || sentence.isEmpty()) {  
        return 0;  
    }  
    StringTokenizer tokens = new StringTokenizer(sentence);  
    return tokens.countTokens();  
}
```

You can see that we have not given any explicit delimiter to StringTokenizer, it uses the default set of delimiter which is enough to find any whitespace and since words are separated by whitespace, the number of tokens is actually equal to the number of words in given String. See [Java How to Program by Dietel](#) for more information on StringTokenizer class in Java.

Don't be the same,
be better!
life advice

EMI options*. SEE MORE ▶
XPS 13
10TH GEN INTEL® CORE™ i7 PROCESSOR
Valid until 30th Jan 2020. T&C's apply.

Books and Resources

- Best Book to Learn Java in 2017
- 5 Books to Learn Spring MVC and Core in 2017
- 2 books to learn Hibernate in 2017
- 12 Advanced Java Programming Books for Experienced Programmers
- 5 Free JavaScript Books to download
- 5 Books to Improve Your Coding Skill
- Books Every Programmer Should Read
- Top 10 Computer Algorithm Books
- 10 Free Java Programming Books
- 5 Books to Learn Java 8 Better
- Books Every Programmer Should Read
- Top 10 Computer Algorithm Books

DON'T JUST WATCH MORE. SEE MORE.
Avail Interest Free EMI options*. SEE MORE ▶
DELL XPS 13

```
public static int count(String word) {  
    if (word == null || word.isEmpty()) {  
        return 0;  
    }  
  
    int wordCount = 0;  
  
    boolean isWord = false;  
    int endOfLine = word.length() - 1;  
    char[] characters = word.toCharArray();  
  
    for (int i = 0; i < characters.length; i++) {  
  
        // if the char is a letter, word = true.  
        if (Character.isLetter(characters[i]) && i != endOfLine) {  
            isWord = true;  
  
            // if char isn't a letter and there have been letters before,  
            // counter goes up.  
        } else if (!Character.isLetter(characters[i]) && isWord) {  
            wordCount++;  
            isWord = false;  
  
            // last word of String; if it doesn't end with a non letter, it  
            // wouldn't count without this.  
        } else if (Character.isLetter(characters[i]) && i == endOfLine) {  
            wordCount++;  
        }  
    }  
  
    return wordCount;  
}
```



- ArrayList (36)
- books (33)
- array (32)
- coding (32)
- Java Multithreading Tutorial (31)
- spring framework (28)
- sql (21)
- date and time (20)
- SQL interview Question (19)
- java io tutorial (19)
- JDBC (18)
- J2EE (17)
- Java programming Tutorial (15)
- Linux (15)
- Servlet (14)
- database (14)
- java (14)
- JSP (13)
- thread interview questions (13)
- JavaScript (12)
- OCAJP (11)
- Eclipse (10)
- Unix (9)
- binary tree (9)
- Java Interview Question (7)
- OCPJP (7)



Solution 1 - Counting word using String.split() method

In this solution, we will use the `split()` method of `java.lang.String` class to count the number of words in a given sentence. This solution uses the regular expression "`\s+`" to split the String on whitespace. The split method returns an array, the length of array is your number of words in given String.

```
public static int countWordsUsingSplit(String input) {  
    if (input == null || input.isEmpty()) {  
        return 0;  
    }  
  
    String[] words = input.split("\\s+");  
    return words.length;  
}
```

If you are new to regular expression in Java, the `\s` is a character class to detect space including tabs, since `\` needs to be escaped in Java, it becomes `\\s` and because there could be multiple spaces between words we made this regular expression greedy by adding `+`, hence `\\s+` will find one more space and split the String accordingly. See [Core Java Volume 1 - Fundamentals](#) by Cay S. Horstmann to learn more about the `split()` method of `String` class. This is also the simplest way to count the number of word in a given sentence.

Solution 2 - Counting word in String using StringTokenizer



Recommended Reading

- 10 Must Read Books for Coders of All Level
- 10 Framework Java Developer Should Learn in 2018
- 10 Books Java Programmers Should Read in 2018
- 10 Open Source Libraries and Framework for Java Developers
- 10 Programming language to Learn in 2018
- The Best Book to Learn Java in 30 days
- 10 Java Web Service Interview Questions
- Top 10 Android Interview Questions for Java Programmers
- 10 Books Every Programmer Should Read
- 5 Great Books to Learn Java 8



```
18.  
19.  
20.  
21. }
```

Output:

```
01. [[1, 2, 3], [11, 12, 13], [4, 5, 6]]
```

5.Using Arrays.asList(array) Method

- Arrays.asList(array) method we can print array elements

#5 Write a program to print array in java using `Arrays.asList(array)`

```
01. package arraysinterview;  
02. import java.util.Arrays;  
03.  
04. public class PrintArray {  
05.     /**  
06.      * How to print java array using Arrays.asList(array)  
07.      * @author www.instanceofjava.com  
08.     */  
09.     public static void main(String [] args){  
10.         int[] array ={1,2,3,4,5,6};  
11.         System.out.println(Arrays.asList(array));  
12.  
13.         String[] strarray ={"hi","array","print"};  
14.         System.out.println(Arrays.asList(strarray));  
15.  
16.     }  
17. }  
18. }
```

Output:

```
01. [[I@2a139a55]  
02. [hi, array, print]
```

Share !

#3 Write a program to print array in java using Arrays.toString(array)

```
01. package arraysinterview;
02. import java.util.Arrays;
03.
04. public class PrintArray {
05.     /**
06.      * How to print java array using Arrays.toString(array)
07.      * @author www.instanceofjava.com
08.
09.     public static void main(String [] args){
10.
11.         String[] array = { "hi", "hello", "java"};
12.
13.
14.         System.out.println(Arrays.toString(array));
15.
16.    }
17. }
```

Output:

```
01. [hi, hello, java]
```

4.Using Arrays.deepToString(array) Method

- Arrays.deepToString(array) method added in java 5 with generics and varargs.

#4 Write a program to print array in java using Arrays.deepToString(array)

```
01. package arraysinterview;
02. import java.util.Arrays;
03.
04. public class PrintArray {
05.     /**
06.      * How to print java array using Arrays.deepToString(array)
07.      * @author www.instanceofjava.com
08.
09.     public static void main(String [] args){
10.
11.         int[][] array = new int[][]{
12.             {1,2,3},
13.             {11,12,13},
14.             {4,5,6},
15.         };
16.     }
17. }
```

Total Pageviews

2 0 4 7 2 8 2 2

```
18.  
19.  
20.  
21. }
```

Output:

```
01. [[1, 2, 3], [11, 12, 13], [4, 5, 6]]
```

5.Using Arrays.asList(array) Method

- Arrays.asList(array) method we can print array elements

#5 Write a program to print array in java using `Arrays.asList(array)`

```
01. package arraysinterview;  
02. import java.util.Arrays;  
03.  
04. public class PrintArray {  
05.     /**  
06.      * How to print java array using Arrays.asList(array)  
07.      * @author www.instanceofjava.com  
08.     */  
09.     public static void main(String [] args){  
10.         int[] array ={1,2,3,4,5,6};  
11.         System.out.println(Arrays.asList(array));  
12.         String[] strarray ={"hi","array","print"};  
13.         System.out.println(Arrays.asList(strarray));  
14.     }  
15. }
```

Output:

```
01. [[I@2a139a55]  
02. [hi, array, print]
```

Share !

[Categories ▾](#)[Library](#) [Videos](#) [Q/A](#) [eBooks](#)

Search your favorite tutorials...



- [DBMS - Relational Data Model](#)
- [DBMS - Relational Algebra](#)
- [DBMS - ER to Relational Model](#)
- [DBMS - SQL Overview](#)

Relational Database Design

- [DBMS - Database Normalization](#)
- [DBMS - Database Joins](#)

Storage and File Structure

- [DBMS - Storage System](#)
- [DBMS - File Structure](#)

Indexing and Hashing

- [DBMS - Indexing](#)
- [DBMS - Hashing](#)

Transaction And Concurrency

- [DBMS - Transaction](#)
- [DBMS - Concurrency Control](#)
- [DBMS - Deadlock](#)

```
B.balance = New_Balance  
Close_Account(B)
```

ACID Properties

A transaction is a very small unit of a program and it may contain several lowlevel tasks. A transaction in a database system must maintain **Atomicity**, **Consistency**, **Isolation**, and **Durability** – commonly known as ACID properties – in order to ensure accuracy, completeness, and data integrity.

- Atomicity** – This property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in a database where a transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.
- Consistency** – The database must remain in a consistent state after any transaction. No transaction should have any adverse effect on the data residing in the database. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.
- Durability** – The database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction updates a chunk of data in a database and commits, then the database will hold the modified data. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system springs back into action.
- Isolation** – In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system. No transaction will affect the existence of any other transaction.

Serializability

Boost your career
with Simplilearn's
**Artificial
Intelligence
Engineer**
Master's Program
and Job Assist.

₹44,999[GET STARTED](#)

connectfastware

With VMware, Airtel quickly delivers new services that delight customers wherever they go.

Computer Science Questions

Operating System Quiz
Computer Architecture MCQs
Software Architecture MCQs
Software Engineering MCQs
Artificial Intelligence MCQs
LISP Programming MCQs
Database Management MCQs
Computer Network MCQs
Microprocessor MCQs

C Programming Examples

Simple C Programs
C - Arrays
C - Matrix
C - Strings
C - Bitwise Operations
C - Linked Lists
C - Stacks & Queues
C - Searching & Sorting
C - Trees
C - Strings
C - File Handling
C - Mathematical Functions
C - Puzzles & Games
C Programs - Recursion
C Programs - No Recursion

Java Algorithms

Java - Numerical Problems
Java - Combinatorial Problems
Java - Graph Problems

Years, Weeks & Days. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Year_Week_Day
3. {
4.     public static void main(String args[])
5.     {
6.         int m, year, week, day;
7.         Scanner s = new Scanner(System.in);
8.         System.out.print("Enter the number of days:");
9.
10.        m = s.nextInt();
11.        year = m / 365;
12.        m = m % 365;
13.        System.out.println("No. of years:"+year);
14.        week = m / 7;
15.        m = m % 7;
16.        System.out.println("No. of weeks:"+week);
17.        day = m;
18.        System.out.println("No. of days:"+day);
19.    }
}
```

Output:

```
$ javac Year_Week_Day.java
$ java Year_Week_Day
```

```
Enter the number of days:756
No. of years:2
No. of weeks:3
No. of days:5
```

Sanfoundry Global Education & Learning Series – 1000 Java Programs.

[Here's the list of Best Reference Books in Java Programming, Data Structures and Algorithms](#)

Ad



Make your game a success

Amazon Appstore

[DOWNLOAD](#)



Sanfoundry is No. 1 choice for Deep Hands-ON Trainings in SAN, Linux & C, Kernel Programming. Our Founder has

[Operating System Quiz](#)
[Computer Architecture MCQs](#)
[Software Architecture MCQs](#)
[Software Engineering MCQs](#)
[Artificial Intelligence MCQs](#)
[LISP Programming MCQs](#)
[Database Management MCQs](#)
[Computer Network MCQs](#)
[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)
[C - Arrays](#)
[C - Matrix](#)
[C - Strings](#)
[C - Bitwise Operations](#)
[C - Linked Lists](#)
[C - Stacks & Queues](#)
[C - Searching & Sorting](#)
[C - Trees](#)
[C - Strings](#)
[C - File Handling](#)
[C - Mathematical Functions](#)
[C - Puzzles & Games](#)
[C Programs - Recursion](#)
[C Programs - No Recursion](#)

Java Algorithms

[Java - Numerical Problems](#)
[Java - Combinatorial Problems](#)

output is also shown below.

```
1. import java.util.Scanner;
2. public class Integer_Conversion
3. {
4.     public static void main(String[] args)
5.     {
6.         int a;
7.         byte b;
8.         char c;
9.         float d;
10.        Scanner s = new Scanner(System.in);
11.        System.out.print("Enter any integer:");
12.        a = s.nextInt();
13.        b = (byte) a;
14.        System.out.println("Conversion into byte:"+b);
15.        c = (char) a;
16.        System.out.println("Conversion into char:"+c);
17.        d = a;
18.        System.out.println("Conversion into float:"+d);
19.    }
20. }
```

Output:

```
$ javac Integer_Conversion.java
$ java Integer_Conversion

Enter any integer:97
Conversion into byte:97
Conversion into char:a
Conversion into float:97.0
```



North America
from INR 58,000*

New York from
INR 58,000*

Boston from
INR 62,000*

Austin from
INR 64,000*

[Book now](#)

* Book by June 15th 2019. Travel until 30 June 2020.
T&C apply.

Lufthansa

Sanfoundry is No. 1 choice for Deep
Hands-ON Trainings in SAN, Linux & C.

Questions & Answers

[C Interview Questions](#)
[C++ Questions](#)
[Linux MCQs](#)
[C# Quiz](#)
[Java MCQs](#)
[JavaScript MCQs](#)
[SAN Questions](#)
[PHP Questions](#)
[Python Quiz](#)

Computer Science Questions

[Operating System Quiz](#)
[Computer Architecture MCQs](#)
[Software Architecture MCQs](#)
[Software Engineering MCQs](#)
[Artificial Intelligence MCQs](#)
[LISP Programming MCQs](#)
[Database Management MCQs](#)
[Computer Network MCQs](#)
[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)
[C - Arrays](#)
[C - Matrix](#)
[C - Strings](#)
[C - Bitwise Operations](#)
[C - Linked Lists](#)
[C - Stacks & Queues](#)

[« Prev Page](#)

[Next Page »](#)

Java Program to Display the IP Address of the System

This is a Java Program to Display the IP Address of the System.

InetAddress class represents an Internet Protocol (IP) address. We use the method named getHostAddress() of this class to get the IP address string in textual presentation.

Here is the source code of the Java Program to Display the IP Address of the System. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.net.InetAddress;
2. public class IP_Address
3. {
4.     public static void main(String args[]) throws Exception
5.     {
6.         InetAddress IP = InetAddress.getLocalHost();
7.         System.out.println("IP of my system is := "+IP.getHostAddress());
8.     }
9. }
```

Output:

```
$ javac IP_Address.java
$ java IP_Address
IP of my system is := 127.0.1.1
```

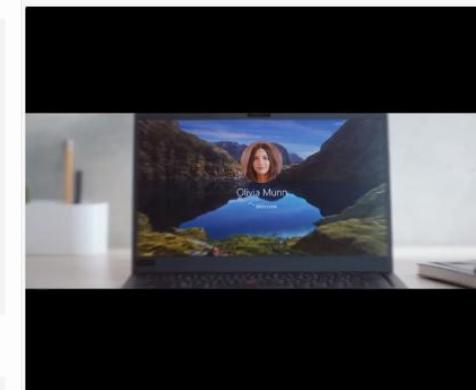
Sanfoundry Global Education & Learning Series – 1000 Java Programs.

[Here's the list of Best Reference Books in Java Programming, Data Structures and Algorithms.](#)

Best Careers

[Developer Tracks](#)
[SAN Developer](#)
[Linux Kernel Developer](#)
[Linux Driver Developer](#)
[Linux Network Developer](#)

[Live Training Photos](#)
[Mentoring](#)
[Software Productivity](#)
[GDB Assignment](#)



Sanfoundry is No. 1 choice for Deep Hands-ON Trainings in SAN, Linux & C, Kernel Programming. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent,

[C Interview Questions](#)
[C++ Questions](#)
[Linux MCQs](#)
[C# Quiz](#)
[Java MCQs](#)
[JavaScript MCQs](#)
[SAN Questions](#)
[PHP Questions](#)
[Python Quiz](#)

Computer Science Questions

[Operating System Quiz](#)
[Computer Architecture MCQs](#)
[Software Architecture MCQs](#)
[Software Engineering MCQs](#)
[Artificial Intelligence MCQs](#)
[LISP Programming MCQs](#)
[Database Management MCQs](#)
[Computer Network MCQs](#)
[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)
[C - Arrays](#)
[C - Matrix](#)
[C - Strings](#)
[C - Bitwise Operations](#)
[C - Linked Lists](#)
[C - Stacks & Queues](#)
[C - Searching & Sorting](#)
[C - Trees](#)

Java Program to Print a Semicolon without using a Semicolon anywhere in the Code

This is a Java Program to Print a Semicolon without using a Semicolon anywhere in the Code.

We take a integer variable and assign it a value of fifty nine. Now with the help Implicit Casting we convert this integer value into char and since ASCII value of semicolon(;) is fifty nine we get the output as semicolon.

Here is the source code of the Java Program to Print a Semicolon without using a Semicolon anywhere in the Code. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

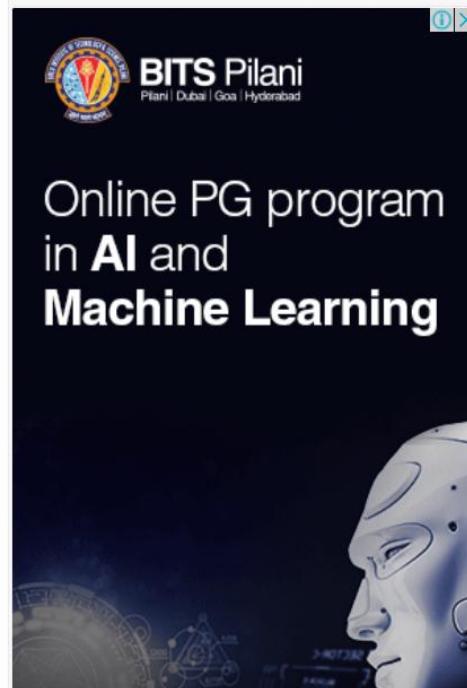
```
1. import java.util.Scanner;
2. public class Semicolon
3. {
4.     public static void main(String[] args)
5.     {
6.         int n;
7.         char a;
8.         n = 59;
9.         a = (char) n;
10.        System.out.println(a);
11.    }
12. }
```

Output:

```
$ javac Semicolon.java
$ java Semicolon
;
```

Developer tracks
[SAN Developer](#)
[Linux Kernel Developer](#)
[Linux Driver Developer](#)
[Linux Network Developer](#)

[Live Training Photos](#)
[Mentoring](#)
[Software Productivity](#)
[GDB Assignment](#)



The advertisement features the BITS Pilani logo at the top left, followed by the text "BITS Pilani" and "Plani | Dubai | Goa | Hyderabad". Below this, the text "Online PG program in AI and Machine Learning" is prominently displayed. In the background, there is a stylized profile of a human head with internal circuitry and data flow, symbolizing artificial intelligence. The overall theme is modern technology and education.

Sanfoundry Technology Education Blog
World's #1 SAN, C & Linux Kernel Bootcamp & Training by our CTO

- Home
- Test & Rank
- About
- Training
- Programming
- CS
- IT
- IS
- ECE
- EEE
- EE
- Civil
- Mechanical
- Chemical
- Metallurgy
- Instrumentation
- Aeronautical
- Aerospace
- Biotechnology
- Mining
- Marine
- Agriculture
- MCA
- BCA
- Internship
- Jobs
- Contact

Search ...



Questions & Answers

< Prev Page

Next Page >

Best Careers

C Interview Questions

C++ Questions

Linux MCQs

Off Quiz

Java MCQs

Android MCQs

SAN Questions

PHP Questions

Python Quiz

Computer Science Questions

Operating System Quiz

Computer Architecture MCQs

Software Architecture MCQs

Software Engineering MCQs

Artificial Intelligence MCQs

LISP Programming MCQs

Database Management MCQs

Computer Networks MCQs

Microprocessor MCQs

C Programming Examples

Simple C Programs

C - Arrays

C - Pointers

C - Strings

C - Bitwise Operations

C - Linked Lists

C - Stacks & Queues

C - Searching & Sorting

C - Trees

C - Strings

C - File Handling

C - Mathematical Functions

C - Puzzles & Games

C Programs - Recursion

C Programs - No Recursion

Java Algorithms

Java - Numerical Problems

Java - Combinatorial Problems

Java - Graph Problems

Java - Hard Graph Problems

Java - Computational Geometry

Java - Sets & Strings

Java - Data Structures

Java - Collection API Problems

C++ Algorithms

C++ - Numerical Problems

C++ - Combinatorial Problems

C++ - Graph Problems

C++ - Hard Graph Problems

C++ - Computation Geometry

C++ - Sets & Strings

C++ - Data Structures

C++ - STL Library

C Algorithms

C - Numerical Problems

C - Combinatorial Problems

C - Graph Problems

C - Hard Graph Problems

C - Computational Geometry

C - Sets & Strings

C - Data Structures

C - STL Library

Sanfoundry Global Education & Learning Series □ 1000 Java Programs.

Java Program to Check if a Given Character is Vowel or Consonant

This is a Java Program to Check if a Given Character is Vowel or Consonant. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.io.BufferedReader;
2. import java.io.IOException;
3. import java.io.InputStreamReader;
4. import java.util.Scanner;
5. public class Vowel_Consonant
6. {
7.     public static void main(String[] args) throws Exception
8.     {
9.         char n;
10.        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
11.        System.out.print("Enter the character you want to check:");
12.        n = (char) bf.read();
13.        switch(n)
14.        {
15.            case 'a':
16.                System.out.println("The given character '"+n+"' is vowel");
17.                break;
18.            case 'e':
19.                System.out.println("The given character '"+n+"' is vowel");
20.                break;
21.            case 'i':
22.                System.out.println("The given character '"+n+"' is vowel");
23.                break;
24.            case 'o':
25.                System.out.println("The given character '"+n+"' is vowel");
26.                break;
27.            case 'u':
28.                System.out.println("The given character '"+n+"' is vowel");
29.                break;
30.            default:
31.                System.out.println("The given character '"+n+"' is consonant");
32.                break;
33.        }
34.    }
35. }
```

Output:
\$ java Vowel_Consonant.java
\$ java Vowel_Consonant

Enter the character you want to check:
The given character b is consonant

Sanfoundry Global Education & Learning Series □ 1000 Java Programs.

< Prev Page - Java Program to Find the Biggest of 3 Numbers

Next Page - Java Program to Find Reverse of a Number using Recursion



Developer Tracks
SAN Developer
Linux Kernel Developer
Linux Driver Developer
Linux Network Developer

Live Training Photos
Meetings
Software Productivity
COB Assignment



Get Hired

VISIT SITE

Sanfoundry is No. 1 choice for Deep Hands-On Trainings in SAN, Linux & C Kernel Programming. Our Founder has trained employees of almost all Top Companies in India such as VMware, Ctrix, Oracle, Motorola, Ericsson, Alcatel, HP, Intel, Microsoft, Oracle, SII, Infosys, Siemens, Symantec, Redhat, Chelsio, Cavium, STMicro, Samsung, LG-Sat, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elexis, Tata VSNL, Mindtree, Cognizant and Startups.

Best Trainings

SAN I - Technology
SAN II - Admin
Linux Fundamentals
Advanced Programming
Linux Debugging
System Programming
Network Programming
Linux Threads
Kernel Programming
Kernel Debugging
Linux Device Drivers



Get Hired

AMCAT

VISIT SITE

Deep Dive @ Sanfoundry:

1. C Programming Examples on Strings
2. Java Programming Examples on Hard Graph Problems & Algorithms
3. Java Programming Examples on Collection API
4. Java Programming Examples on Networking
5. Java Programming Examples on Classes
6. Java Programming Examples on JavaLang
7. Java Programming Examples on File Handling
8. Java Programming Examples on Inheritance
9. Java Programming Examples on String Handling
10. Simple Java Programs

Are You as Tech Savvy as You Think?

In Saudi Arabia, hacking is punished with the removal of one of the offender's hands

0/10 SCORE

TRUE ✓ FALSE ✗

YOU DON'T HAVE TO BE PERFECT TO BE A PARENT.

Powered by Catebollo

Manish Bhojania
Founder & CTO
Sanfoundry

[View profile](#) [LinkedIn](#)

Advanced C Programming, SAN Storage Technologies, SCSI Internals & Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him @ [LinkedIn](#) | [Facebook](#) | [Twitter](#)

Manish Bhojania, a technology veteran with 20+ years @ Cisco & Wipro, is Founder and CTO at Sanfoundry. He is Linux Kernel Developer & SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and currently works as a consultant for various IT organizations. He has contributed articles on Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage, Advanced C Programming, SAN Storage Technologies, SCSI Internals & Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him @ [LinkedIn](#) | [Facebook](#) | [Twitter](#)

All Stream Best Books

1000 C Questions & Answers

1000 C++ Questions & Answers

1000 Java Questions & Answers

1000 Linux Questions & Answers

1000 Python Questions

1000 PHP Questions & Answers

1000 MySQL Questions & Answers

Cloud Computing Questions

Computer Science Questions

All Stream Questions & Answers

1000 Questions & Answers

1000 MCQs

1000 Interview Questions

1000 Case Studies

1000 Project Ideas

1000 Research Papers

1000 Books

1000 Videos

1000 Slides

1000 PDFs

1000 eBooks

1000 Infographics

1000 Diagrams

1000 Charts

1000 Tables

1000 Figures

1000 Graphs

1000 Animations

1000 Videos

1000 Scripts

[SAN Questions](#)
[PHP Questions](#)
[Python Quiz](#)

Computer Science Questions

[Operating System Quiz](#)
[Computer Architecture MCQs](#)
[Software Architecture MCQs](#)
[Software Engineering MCQs](#)
[Artificial Intelligence MCQs](#)
[LISP Programming MCQs](#)
[Database Management MCQs](#)
[Computer Network MCQs](#)
[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)
[C - Arrays](#)
[C - Matrix](#)
[C - Strings](#)
[C - Bitwise Operations](#)
[C - Linked Lists](#)
[C - Stacks & Queues](#)
[C - Searching & Sorting](#)
[C - Trees](#)
[C - Strings](#)
[C - File Handling](#)
[C - Mathematical Functions](#)
[C - Puzzles & Games](#)
[C Programs - Recursion](#)
[C Programs - No Recursion](#)

Enter any character as input. After that we match the ASCII value of that character against the given three cases. If that value matches then we get the output accordingly.

Here is the source code of the Java Program to Check if given Alphabets are Uppercase or Lowercase or Digits. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.io.BufferedReader;
2. import java.io.IOException;
3. import java.io.InputStreamReader;
4. public class Alphabet_Check
5. {
6.     public static void main(String args[]) throws IOException
7.     {
8.         char m;
9.         BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
10.        System.out.print("Enter any alphabet:");
11.        m = (char) bf.read();
12.        if(m >= 97 && m <= 123)
13.        {
14.            System.out.println("Lower Case");
15.        }
16.        else if(m >= 65 && m <= 96)
17.        {
18.            System.out.println("Upper Case");
19.        }
20.        else if(m >= 48 && m <= 57)
21.        {
22.            System.out.println("Digit");
23.        }
24.    }
25. }
```

Output:

[Live Training Photos](#)
[Mentoring](#)
[Software Productivity](#)
[GDB Assignment](#)



New ŠKODA Rapid Monte Carlo



[BOOK NOW](#)

Sanfoundry is No. 1 choice for Deep Hands-ON Trainings in SAN, Linux & C, Kernel Programming. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.

Computer Science Questions

Operating System Quiz
Computer Architecture MCQs
Software Architecture MCQs
Software Engineering MCQs
Artificial Intelligence MCQs
LISP Programming MCQs
Database Management MCQs
Computer Network MCQs
Microprocessor MCQs

C Programming Examples

Simple C Programs
C - Arrays
C - Matrix
C - Strings
C - Bitwise Operations
C - Linked Lists
C - Stacks & Queues
C - Searching & Sorting
C - Trees
C - Strings
C - File Handling
C - Mathematical Functions
C - Puzzles & Games
C Programs - Recursion
C Programs - No Recursion

Java Algorithms

Here is the source code of the Java Program to Illustrate Use of Binary Literals. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Binary_Literal
3. {
4.     public static void main(String[] args)
5.     {
6.         byte aB = 0b00100001;
7.         short aS = 0b10100010100;
8.         int a1 = 0b10110;
9.         int a2 = 0b101;
10.        int a3 = 0b1011;
11.        int aI=a2+a3;
12.        System.out.println("Byte value:"+aB);
13.        System.out.println("Short value:"+aS);
14.        System.out.println("Integer value:"+a1);
15.        System.out.println("Result:"+aI);
16.    }
17. }
```

Output:

```
$ javac Binary_Literal.java
$ java Binary_Literal

Byte value:33
Short value:1300
Integer value:22
Result:16
```

Sanfoundry Global Education & Learning Series – 1000 Java Programs

GDB Assignment



The Machine Learning Course that gets you Hired. 15+ industry projects taught end to end.

Applied AI Course

VISIT SITE

Sanfoundry is No. 1 choice for Deep Hands-ON Trainings in SAN, Linux & C, Kernel Programming. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.



World's #1 SAN, C & Linux Kernel Bootcamp & Training by our CTO

Home Test & Rank About Training Programming CS IT IS ECE EEE EE Civil Mechanical Chemical Metallurgy
Instrumentation Aeronautical Aerospace Biotechnology Mining Marine Agriculture MCA BCA Internship Jobs Contact
Search ...

Be a Certified CLOUD EXPERT
Accredited By IBM, MICROSOFT

splunk> AWS ENROLL NOW

intelliPaat

Questions & Answers « Prev Page Next Page » Best Careers

Java Program to Find if a Given Year is a Leap Year

This is a Java Program to Find if a Given Year is a Leap Year.

Enter any year as an input. We first check whether the given year is divisible by 400 or not. If it is divisible then it is a leap year else we check for further conditions. Now if it is divisible by 100 then it is not a leap year or else we further divide it by 4. If it is divisible then it is a leap year else its not.

Here is the source code of the Java Program to Find if a Given Year is a Leap Year. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
import java.util.Scanner;
public class Check_Leap_Year {
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter any year:");
        int year = s.nextInt();
        boolean flag = false;
        if(year % 400 == 0)
        {
            flag = true;
        }
        else if (year % 100 == 0)
        {
            flag = false;
        }
        else if (year % 4 == 0)
        {
            flag = true;
        }
        else
        {
            flag = false;
        }
        if(flag)
            System.out.println("Year "+year+" is a Leap Year");
        else
            System.out.println("Year "+year+" is not a Leap Year");
    }
}
```

Output:

```
$ java Check_Leap_Year
java Check_Leap_Year

Enter any year:1800
Year 1800 is not a Leap Year

Enter any year:2000
Year 2000 is a Leap Year
```

Sanfoundry Global Education & Learning Series – 1000 Java Programs.

Here's the list of Best Reference Books in Java Programming, Data Structures and Algorithms.

« Prev Page Java Programs to Check Whether Given Number is Divisible by 5
» Next Page Java Programs to Display Numbers from 1 to 10 Using For Loop

[BOOK NOW](#)
Travel now! 10% off till Mar 2020
| TAC apply

C++ Interview Questions
C++ Questions
Linux MCQs
C# Quiz
Java MCQs
JavaScript MCQs
SAP MCQs
PHP Questions
Python Quiz

Computer Science Questions
Operating System MCQs
Computer Architecture MCQs
Software Architecture MCQs
Software Engineering MCQs
Artificial Intelligence MCQs
UML Programming MCQs
Database Management MCQs
Computer Network MCQs
Microprocessor MCQs

C Programming Examples
Simple C Programs
C - Arrays
C - Matrix
C - Strings
C - Data Operations
C - Linked Lists
C - Stacks & Queues
C - Searching & Sorting
C - Trees
C - Strings
C - File Handling
C - Mathematical Functions
C - Functions & Games
C Programs - Recursion
C Programs - No Recursion

Java Algorithm
Java - Numerical Problems
Java - Combinatorial Problems
Java - Graph Problems
Java - Hard Graph Problems
Java - Computation Geometry
Java - Sets & Strings
Java - Data Structures
Java - Collection API Problems

C++ Algorithms
C++ - Numerical Problems
C++ - Combinatorial Problems
C++ - Graph Problems
C++ - Hard Graph Problems
C++ - Computation Geometry
C++ - Sets & Strings
C++ - Data Structures
C++ - STL Library

C Algorithms
C - Numerical Problems
C - Combinatorial Problems
C - Graph Problems
C - Hard Graph Problems
C - Computation Geometry
C - Sets & Strings
C - Data Structures

Deep Dive @ Sanfoundry:
1. Java Programming Examples on Graph Problems & Algorithms
2. Java Programming Examples on Numerical Problems & Algorithms
3. Java Programming Examples on Combinatorial Problems & Algorithms
4. Java Programming Examples on Computational Geometry Problems & Algorithms
5. Java Programming Examples on Hard Graph Problems & Algorithms
6. Simple C Programs
7. Java Programming Examples on Data-Structures
8. Java Algorithms, Problems & Programming Examples
9. C# Basic Programming Examples
10. Java Programming Examples on Collection API

Are These Made-Up Microsoft Facts?
Halo was actually first designed for Mac OS, but Microsoft bought it in 2000 and made it Xbox exclusive

0/10 SCORE
Powered by Gabekello

LinkedIn Ads
Ad - Sanfoundry

Reach 55M+ professionals in India with LinkedIn Ads

LinkedIn Ads
Learn More

LinkedIn Ads
Learn More

Manish Bhajoria, a technology veteran with 20+ years of experience. He is Founder and CEO of Sanfoundry. He is Linux Kernel Developer & SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and delivers focused training sessions to IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage, Advanced C Programming, SAN Storage Technologies, SCSI Internals & Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him @ LinkedIn | Facebook | Twitter

Best Reference Books
Questions and Answers
India Internships
About Sanfoundry

[PHP Questions](#)
[Python Quiz](#)

Computer Science Questions

[Operating System Quiz](#)
[Computer Architecture MCQs](#)
[Software Architecture MCQs](#)
[Software Engineering MCQs](#)
[Artificial Intelligence MCQs](#)
[LISP Programming MCQs](#)
[Database Management MCQs](#)
[Computer Network MCQs](#)
[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)
[C - Arrays](#)
[C - Matrix](#)
[C - Strings](#)
[C - Bitwise Operations](#)
[C - Linked Lists](#)
[C - Stacks & Queues](#)
[C - Searching & Sorting](#)
[C - Trees](#)
[C - Strings](#)
[C - File Handling](#)
[C - Mathematical Functions](#)
[C - Puzzles & Games](#)
[C Programs - Recursion](#)
[C Programs - No Recursion](#)

Here is the source code of the Java Program to Extract Last two Digits of a given Year. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Extract_Digit
3. {
4.     public static void main(String[] args)
5.     {
6.         int x, y, i = 0;
7.         String z = "";
8.         Scanner s = new Scanner(System.in);
9.         System.out.print("Enter Year:");
10.        x = s.nextInt();
11.        while(i < 2)
12.        {
13.            y = x % 10;
14.            z = y + "" + z;
15.            x = x / 10;
16.            i++;
17.        }
18.        System.out.println("Last two digits:" +z);
19.    }
20. }
```

Output:

```
$ javac Extract_Digit .java
$ java Extract_Digit
```

```
Enter Year:2014
```

```
Last two digits:14
```

Mentoring
Software Productivity
GDB Assignment



We are Animation Training Institute provide Practical Courses with Job opportunities



Arena Bhawarkuan Indore

[VISIT SITE](#)

Sanfoundry is No. 1 choice for Deep Hands-ON Trainings in SAN, Linux & C, Kernel Programming. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.

Operating System Quiz
Computer Architecture MCQs
Software Architecture MCQs
Software Engineering MCQs
Artificial Intelligence MCQs
LISP Programming MCQs
Database Management MCQs
Computer Network MCQs
Microprocessor MCQs

C Programming Examples

Simple C Programs
C - Arrays
C - Matrix
C - Strings
C - Bitwise Operations
C - Linked Lists
C - Stacks & Queues
C - Searching & Sorting
C - Trees
C - Strings
C - File Handling
C - Mathematical Functions
C - Puzzles & Games
C Programs - Recursion
C Programs - No Recursion

Java Algorithms

Java - Numerical Problems
Java - Combinatorial Problems

```
1. import java.util.Scanner;
2. public class Strings_Switch
3. {
4.     public static void main(String[] args)
5.     {
6.         String week;
7.         Scanner s = new Scanner(System.in);
8.         System.out.print("Enter choice:");
9.         week = s.nextLine();
10.        switch (week)
11.        {
12.            case "Monday":
13.                System.out.print("Day: Monday");
14.                break;
15.
16.            case "Tuesday":
17.                System.out.print("Day: Tuesday");
18.                break;
19.
20.            case "Wednesday":
21.                System.out.print("Day: Wednesday");
22.                break;
23.
24.            case "Thursday":
25.                System.out.print("Day: Thursday");
26.                break;
27.
28.            case "Friday":
29.                System.out.print("Day: Friday");
30.                break;
31.
32.            case "Saturday":
33.                System.out.print("Day: Saturday");
```



Sanfoundry is No. 1 choice for Deep Hands-ON Trainings in SAN, Linux & C, Kernel Programming. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.



C - Bitwise Operations
C - Linked Lists
C - Stacks & Queues
C - Searching & Sorting
C - Trees
C - Strings
C - File Handling
C - Mathematical Functions
C - Puzzles & Games
C Programs - Recursion
C Programs - No Recursion

Java Algorithms

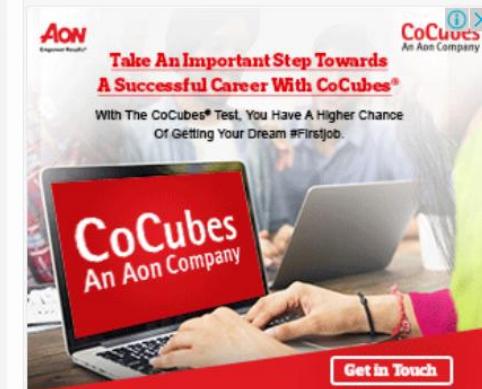
Java - Numerical Problems
Java - Combinatorial Problems
Java - Graph Problems
Java - Hard Graph Problems
Java - Computation Geometry
Java - Sets & Strings
Java - Data-Structures
Java - Collection API Problems

C++ Algorithms

C++ - Numerical Problems
C++ - Combinatorial Problems
C++ - Graph Problems
C++ - Hard Graph Problems
C++ - Computation Geometry
C++ - Sets & Strings
C++ - Data-Structures
C++ - STL Library

```
2. //Java Program to Find the Roots of a Quadratic Equation
3.
4. import java.io.BufferedReader;
5. import java.io.InputStreamReader;
6.
7. public class Quadratic {
8.     // Function to find and display the roots of the equation.
9.     public static void main(String[] args) {
10.         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
11.         double a,b,c;
12.         try{
13.             System.out.println("Enter the coefficients of the quadratic equation");
14.             a = Double.parseDouble(br.readLine());
15.             b = Double.parseDouble(br.readLine());
16.             c = Double.parseDouble(br.readLine());
17.         }catch (Exception e){
18.             System.out.println("An error occurred");
19.             return;
20.         }
21.         double determinant = Math.pow(b,2) - 4*a*c;
22.         if(determinant > 0){
23.             System.out.println("Roots are " + (-b+Math.sqrt(determinant))/(2*a)
24.                               + " and " + (-b-Math.sqrt(determinant))/(2*a));
25.         }else if (determinant == 0){
26.             System.out.println("Roots are " + -b/(2*a));
27.         }
28.         else{
29.             System.out.println("Roots are " + -b/(2*a) + "+i" +
30.                               Math.sqrt(-determinant)/(2*a) + " and "
31.                               + -b/(2*a) + "-i" + Math.sqrt(-determinant)/(2*a));
32.         }
33.     }
34. }
```

trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.

An advertisement for CoCubes, an Aon company. It features a person working on a laptop with the CoCubes logo on the screen. The text reads: "Take An Important Step Towards A Successful Career With CoCubes®. With The CoCubes® Test, You Have A Higher Chance Of Getting Your Dream #FirstJob." There is a "Get in Touch" button at the bottom right.

Aon
CoCubes
An Aon Company

Take An Important Step Towards
A Successful Career With CoCubes®

With The CoCubes® Test, You Have A Higher Chance
Of Getting Your Dream #FirstJob.

CoCubes
An Aon Company

Get in Touch

Best Trainings

SAN I - Technology
SAN II - Admin
Linux Fundamentals
Advanced C Training
Linux-C Debugging
System Programming
Network Programming
Linux Threads
Kernel Programming

Implementation

```
/*Function to left rotate arr[] of size n by d*/
void leftRotate(int arr[], int d, int n)
{
    int i, j, k, temp;
    for (i = 0; i < gcd(d, n); i++)
    {
        /* move i-th values of blocks */
        temp = arr[i];
        j = i;
        while(1)
        {
            k = j + d;
            if (k >= n)
                k = k - n;
            if (k == i)
                break;
            arr[j] = arr[k];
            j = k;
        }
        arr[j] = temp;
    }
}
```

the number of sets formed. The time complexity of this algorithm is O(n)

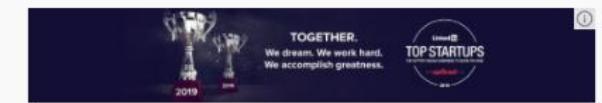
4:36 / 4:48

Array Rotation | GeeksforGeeks

105,229 views • Aug 22, 2017

1,304 205 SHARE SAVE ...

SUBSCRIBE



Global MBA
Ad upGrad.com

LEARN MORE

Up next

AUTOPLAY



ARRAY ROTATION IN-PLACE [JUGGLING ALGORITHM]
CodeWhoop 14:42



How to attempt GATE 2020 | Last Minute GATE Exam Tips |...
Gate Lectures by Ravindraba... 5:12
Recommended for you
New



The best software Interview material - Prepare in less than ...
Gaurav Sen 17:23
235K views



IELTS Speaking Score 7.5
GeneralIELTSHelp 12:30
Recommended for you



Print a given matrix in spiral form | GeeksforGeeks

Java program to remove duplicates in array using LinkedHashSet. In given example, `numbers` is an integer array which has duplicate numbers 1, 3 and 5. We add all elements to LinkedHashSet, and then get back the content in array. The result array does not have duplicate integers.

ArrayExample.java

```
import java.util.Arrays;
import java.util.LinkedHashSet;

public class ArrayExample
{
    public static void main(String[] args) throws CloneNotSupportedException
    {
        //Array with duplicate elements
        Integer[] numbers = new Integer[] {1,2,3,4,5,1,3,5};

        //This array has duplicate elements
        System.out.println( Arrays.toString(numbers) );

        //Create set from array elements
        LinkedHashSet<Integer> linkedHashSet = new LinkedHashSet<>( Arrays.asList(numbers) );

        //Get back the array without duplicates
        Integer[] numbersWithoutDuplicates = linkedHashSet.toArray(new Integer[] {});

        //Verify the array content
        System.out.println( Arrays.toString(numbersWithoutDuplicates) );
    }
}
```

Program Output.

Console

```
[1, 2, 3, 4, 5, 1, 3, 5]
[1, 2, 3, 4, 5]
```

```

{
    // Array with duplicate elements
    Integer[] origArray = new Integer[] { 1, 1, 2, 3, 3, 3, 4, 5, 6, 6, 6, 7, 8 };

    // This array has duplicate elements
    System.out.println(Arrays.toString(origArray));

    Integer[] tempArray = removeDuplicates(origArray);

    // Verify the array content
    System.out.println(Arrays.toString(tempArray));
}

private static Integer[] removeDuplicates(Integer[] origArray) {

    Integer[] tempArray = new Integer[origArray.length];

    int indexJ = 0;
    for (int indexI = 0; indexI < origArray.length - 1; indexI++) {
    {
        Integer currentElement = origArray[indexI];

        if (currentElement != origArray[indexI+1]) {
            tempArray[indexJ++] = currentElement;
        }
    }

    tempArray[indexJ++] = origArray[origArray.length-1];

    return tempArray;
}
}

```

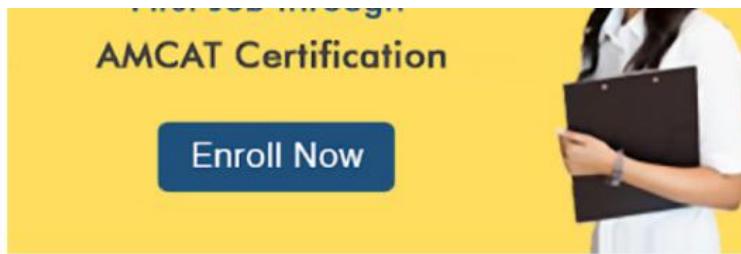
Program Output.

Console

```

[1, 1, 2, 3, 3, 3, 4, 5, 6, 6, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8, null, null, null, null, null]

```



3.What will happen if we extends a class which is having private constructor.

- If we try to extend a class which is having private constructor compile time error will come.
- Implicit super constructor is not visible for default constructor. Must define an explicit constructor

```
package privateConstructorInterviewQuestions;
public class Sample {
    private Sample(){}
    public static void main(String[] args) {
        Sample obj= new Sample();
    }
}
package privateConstructorInterviewQuestions;
public class Test extends Sample {
    /**
     * @ www.
     */
    public static void main(String[] args) {
    }
}
```

The code shows a Java class named Sample with a private constructor. It also contains a main method that creates an instance of Sample. Below this, another class named Test is defined, which extends Sample. The main method of Test also creates an instance of Sample. A tooltip appears over the line "Implicit super constructor Sample() is not visible for default constructor. Must define an explicit constructor" with the text "Press 'F2' for focus".



Singleton Design pattern:

A screenshot of the InstanceOfJava.com Facebook page. The page has 3,802 likes. It features a profile picture of a coffee cup, the page name "InstanceOfJava.com", and a bio "We will help you in Learning". There are links to "Like Page" and "Contact Us". A message box at the bottom says "Be the first of your friends to like this".

```
13. {
14.     if (object == null)
15.     {
16.         System.out.println("getInstance(): First time getInstance was called and object created !");
17.         object = new SingletonClass ();
18.     }
19. }
20. return object;
21. }
22. }
23. return object;
24. }
25. }
26. }
```

- [Java Online Training](#) >
- [Artificial Intelligence Programs](#) >
- [Java Coding Course](#) >
- [Java Certification Courses](#) >
- [Learn Java Online](#) >
- [Advanced Java Tutorial](#) >



```
01. package instanceofjava;
02.
03. public class SingletonObjectDemo {
04.
05.     public static void main(String args[]) {
06.
07.         SingletonClass s1= SingletonClass .getInstance();
08.         SingletonClass s2= SingletonClass .getInstance();
09.         System.out.println(s1.hashCode());
10.         System.out.println(s2.hashCode());
11.
12.     }
13. }
```

Output:

```
01. getInstance(): First time getInstance was called and object created !
02. Singleton(): Private constructor invoked
03. 655022016
04. 655022016
```

```
01. package com.privateConstructorSingleton;
02.
03. public class SingletonClass {
04.
05.     private static SingletonClass object;
06.
07.     private SingletonClass () {
08.         System.out.println("Singleton(): Private constructor invoked");
09.     }
10.
11.     public static SingletonClass getInstance()
12.     {
13.
14.         if (object == null)
15.         {
16.
17.             System.out.println("getInstance(): First time getInstance was called and object created !");
18.             object = new SingletonClass ();
19.         }
20.
21.     }
22.
23.     return object;
24.
25. }
26.
27. }
```

Total Pageviews

2 0 4 7 1 8 6 8

- Java Online Training >
- Java Coding Course >
- Learn Java Online >
- Artificial Intelligence Programs >
- Java Certification Courses >
- Advanced Java Tutorial >



```
01. package instanceofjava;
02.
03. public class SingletonObjectDemo {
04.
05.     public static void main(String args[]) {
06.
07.         SingletonClass s1= SingletonClass .getInstance();
08.         SingletonClass s2= SingletonClass .getInstance();
09.         System.out.println(s1.hashCode());
10.         System.out.println(s2.hashCode());
```

```
01. array  
02. arraylist  
03. conversion  
04. javaprogram
```

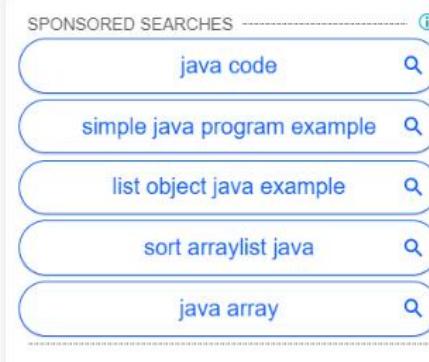
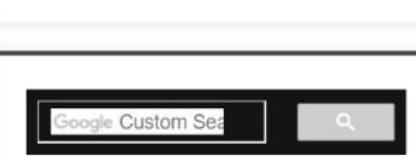
Program #2: Java example program to covert Integer arraylist to int array using toArray() method

- In this case toArray() method gives Integer array so we need to convert again Integer array to int array.

```
01. package arraysinterview;  
02. import java.util.ArrayList;  
03. import java.util.List;  
04.  
05. public class ArrayListTOArray {  
06.     public static void main(String[] args) {  
07.         List<Integer> list = new ArrayList<Integer>();  
08.  
09.         list.add(10);  
10.         list.add(20);  
11.         list.add(30);  
12.         list.add(40);  
13.         list.add(50);  
14.  
15.         Object[] integers = list.toArray();  
16.  
17.         int[] intarray = new int[integers.length];  
18.         int i = 0;  
19.         for (Object n : integers) {  
20.             intarray[i++] = (Integer) n;  
21.             System.out.println(i);  
22.         }  
23.  
24.     }  
25.  
26. }  
27.  
28. }
```

Output:

```
01. 1  
02. 2  
03. 3  
04. 4  
05. 5
```



[Enroll Now](#)

- Lets see a java program on how to convert arraylist to array.

Program #1: Java example program to covert arraylist to array using toArray() method

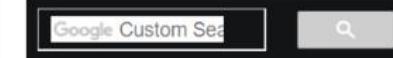
```
01. package arraysinterview;
02. import java.util.ArrayList;
03. import java.util.List;
04.
05. public class ArrayListTOArray {
06.
07.     public static void main(String[] args) {
08.         List<String> list = new ArrayList<String>();
09.
10.        list.add("array");
11.        list.add("arraylist");
12.        list.add("conversion");
13.        list.add("javaprogram");
14.
15.        String [] str = list.toArray(new String[list.size()]);
16.
17.        for (int i = 0; i < str.length; i++) {
18.            System.out.println(str[i]);
19.        }
20.
21.    }
22.
23. }
```

Output:

```
01. array
02. arraylist
03. conversion
04. javaprogram
```

Program #2: Java example program to covert Integer arraylist to int array using toArray() method

- In this case toArray() method gives Integer array so we need to convert again Integer array to int array.





59. What will be the output of following programs?

1. static method in class

```
package com.journaldev.util;

public class Test {

    public static String toString(){
        System.out.println("Test toString called");
        return "";
    }

    public static void main(String args[]){
        System.out.println(toString());
    }
}
```

Answer: The code won't compile because we can't have an `Object` class method with static keyword. Note that `Object` class has `toString()` method. You will get a compile-time error as "This static method cannot hide the instance method from Object". The reason is that static method belongs to the class and since every class base is `Object`, we can't have the same method in the instance as well as in class. You won't get this error if you change the method name from `toString()` to something else that is not present in superclass `Object`.

2. static method invocation

```
package com.journaldev.util;

public class Test {
```





the method name from `toString()` to something else that is not present in superclass `Object`.

2. static method invocation

```
package com.journaldev.util;

public class Test {

    public static String foo(){
        System.out.println("Test foo called");
        return "";
    }

    public static void main(String args[]){
        Test obj = null;
        System.out.println(obj.foo());
    }
}
```

Copy

Answer: Well this is a strange situation. We all have seen `NullPointerException` when we invoke a method on the object that is NULL. But here this program will work and prints "Test foo called".

The reason for this is the Java compiler code optimization. When the Java code is compiled to produce byte code, it figures out that `foo()` is a static method and should be called using class. So it changes the method call `obj.foo()` to `Test.foo()` and hence no `NullPointerException`.

I must admit that it's a very tricky question and if you are interviewing someone, this will blow his mind off. 😊

I will keep on adding more questions to the list, if you think I missed any important one, please let me know through comments.



[File I/O](#)

[Generics](#)

[Java Collections](#)

[JDBC](#)

[Interview Programs](#)

[Interview Q&A](#)

[x Vs y](#)

[Java 8](#)

Collection Interface :

Collection is a root level interface of the Java Collection Framework. Most of the classes in Java Collection Framework inherit from this interface. **List**, **Set** and **Queue** are main sub interfaces of this interface. JDK doesn't provide any direct implementations of this interface. But, JDK provides direct implementations of it's sub interfaces. **ArrayList**, **Vector**, **HashSet**, **LinkedHashSet**, **PriorityQueue** are some indirect implementations of Collection interface. **Map interface**, which is also a part of java collection framework, doesn't inherit from Collection interface. Collection interface is a member of `java.util` package.



[Click here for more info on Collection interface in java.](#)

Collections Class:

Collections is an utility class in `java.util` package. It consists of only static methods which are used to operate on objects

Subscribe To Our Latest Updates

Join 11,855 other subscribers

Email Address

Subscribe

Java 8 Tutorials

[Java 8 Lambda Expressions](#)

[Java 8 Functional Interfaces](#)

[Java 8 Method References](#)

[Java 8 Interface Changes](#)

[Java 8 Streams](#)

[Java 8 Collectors](#)

[Interface Vs Abstract Class After Java 8](#)

[Collections Vs Streams](#)

[Solving Real Time Queries Using Java 8 - Employee Management System](#)

Nested Classes

Java Enums

Wrapper Classes

Arrays

Strings

Exception Handling

Threads

File I/O

Generics

Java Collections

JDBC

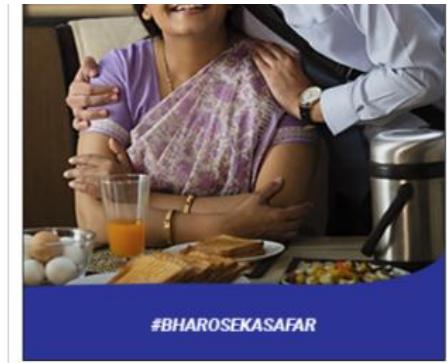
Interview Programs

Interview Q&A

x Vs y

Java 8

utility class. In this article, we will discuss the differences between Collection and Collections in java.



Collection Interface :

Collection is a root level interface of the Java Collection Framework. Most of the classes in Java Collection Framework inherit from this interface. **List**, **Set** and **Queue** are main sub interfaces of this interface. JDK doesn't provide any direct implementations of this interface. But, JDK provides direct implementations of it's sub interfaces. **ArrayList**, **Vector**, **HashSet**, **LinkedHashSet**, **PriorityQueue** are some indirect implementations of Collection interface. **Map interface**, which is also a part of java collection framework, doesn't inherit from Collection interface. Collection interface is a member of `java.util` package.



Subscribe To Our Latest Updates

Join 11,855 other subscribers

Email Address

Subscribe

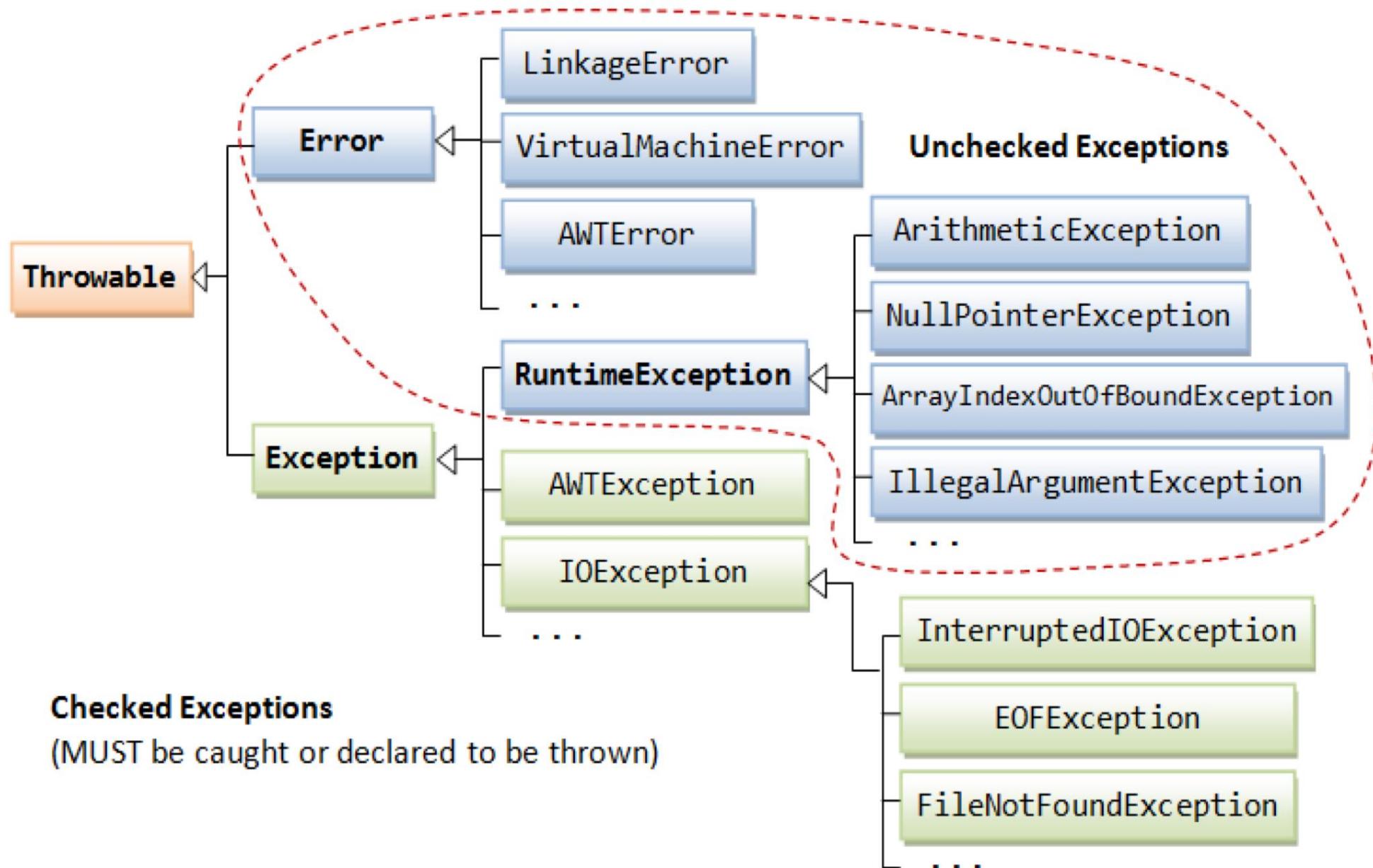
Java 8 Tutorials

Java 8 Lambda Expressions

Java 8 Functional Interfaces

Java 8 Method References

Exception Handling



Computer Science Questions

[Operating System Quiz](#)

[Computer Architecture MCQs](#)

[Software Architecture MCQs](#)

[Software Engineering MCQs](#)

[Artificial Intelligence MCQs](#)

[LISP Programming MCQs](#)

[Database Management MCQs](#)

[Computer Network MCQs](#)

[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)

[C - Arrays](#)

[C - Matrix](#)

[C - Strings](#)

[C - Bitwise Operations](#)

[C - Linked Lists](#)

[C - Stacks & Queues](#)

[C - Searching & Sorting](#)

[C - Trees](#)

[C - Strings](#)

[C - File Handling](#)

[C - Mathematical Functions](#)

[C - Puzzles & Games](#)

[C Programs - Recursion](#)

[C Programs - No Recursion](#)

Java Algorithms

[Java - Numerical Problems](#)

[Java - Combinatorial Problems](#)

[Java - Graph Problems](#)

[Java - Hard Graph Problems](#)

[Java - Computation Geometry](#)

[Java - Sets & Strings](#)

[Java - Data-Structures](#)

[Java - Collection API Problems](#)

C++ Algorithms

Here is the source code of the Java Program to Check if a given Bit Position is set to One or not. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Bit_Postion
3. {
4.     public static void main(String[] args)
5.     {
6.         int n, m;
7.         String x = "";
8.         Scanner s = new Scanner(System.in);
9.         System.out.print("Enter any Decimal Number:");
10.        n = s.nextInt();
11.        while(n > 0)
12.        {
13.            int a = n % 2;
14.            x = a + x;
15.            n = n / 2;
16.        }
17.        System.out.print("Enter the position where you want to check:");
18.        int l = x.length();
19.        m = s.nextInt();
20.        if((l - m) >= 0 && (x.charAt(l - m) == '1'))
21.        {
22.            System.out.println("1 is present at given bit position");
23.        }
24.        else
25.        {
26.            System.out.println("0 is present at given bit position");
27.        }
28.    }
29. }
```

Output:

```
$ javac Bit_Postion.java
$ java Bit_Postion
```

```
Enter any Decimal Number:6
Enter the position where you want to check:2
1 is present at given bit position
```

GDB Assignment



Earn \$35 Per Page

Online Career Inc.

[VISIT SITE](#)

Sanfoundry is **No. 1** choice for Deep Hands-ON Trainings in **SAN, Linux & C, Kernel Programming**. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.



Enter your Birth infos now & get Precise Details about Yourself.

 in-astrology.com

[VISIT SITE](#)

Python Quiz

Computer Science Questions

Operating System Quiz

Computer Architecture MCQs

Software Architecture MCQs

Software Engineering MCQs

Artificial Intelligence MCQs

LISP Programming MCQs

Database Management MCQs

Computer Network MCQs

Microprocessor MCQs

C Programming Examples

Simple C Programs

C - Arrays

C - Matrix

C - Strings

C - Bitwise Operations

C - Linked Lists

C - Stacks & Queues

C - Searching & Sorting

C - Trees

C - Strings

C - File Handling

C - Mathematical Functions

C - Puzzles & Games

C Programs - Recursion

C Programs - No Recursion

Java Algorithms

Java - Numerical Problems

Java - Combinatorial Problems

Java - Graph Problems

Java - Hard Graph Problems

Java - Computation Geometry

Using Data Structures

Source code of this program to get the count of 1's as the output.

Here is the source code of the Java Program to Count the Number of Bits set to One. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Count_One
3. {
4.     public static void main(String[] args)
5.     {
6.         int n, m, count = 0;
7.         String x = "";
8.         Scanner s = new Scanner(System.in);
9.         System.out.print("Enter the Decimal Number:");
10.        n = s.nextInt();
11.        while(n > 0)
12.        {
13.            int a = n % 2;
14.            x = a + x;
15.            n = n / 2;
16.        }
17.        int l = x.length();
18.        for(int i = 0; i < l; i++)
19.        {
20.            if(x.charAt(i) == '1')
21.            {
22.                count++;
23.            }
24.        }
25.        System.out.println("No. of 1's are:"+count);
26.    }
27. }
```

Output:

```
$ javac Count_One.java
$ java Count_One
```

Enter the Decimal Number:15

No. of 1's are:4

Marketing

Software Productivity
GDB Assignment



Earn \$35 Per Page

Online Career Inc.

VISIT SITE

Sanfoundry is **No. 1** choice for Deep Hands-ON Trainings in **SAN, Linux & C, Kernel Programming**. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.



Enter your Birth infos now & get Precise

Python Quiz

Computer Science Questions

[Operating System Quiz](#)

[Computer Architecture MCQs](#)

[Software Architecture MCQs](#)

[Software Engineering MCQs](#)

[Artificial Intelligence MCQs](#)

[LISP Programming MCQs](#)

[Database Management MCQs](#)

[Computer Network MCQs](#)

[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)

[C - Arrays](#)

[C - Matrix](#)

[C - Strings](#)

[C - Bitwise Operations](#)

[C - Linked Lists](#)

[C - Stacks & Queues](#)

[C - Searching & Sorting](#)

[C - Trees](#)

[C - Strings](#)

[C - File Handling](#)

[C - Mathematical Functions](#)

[C - Puzzles & Games](#)

[C Programs - Recursion](#)

[C Programs - No Recursion](#)

Java Algorithms

[Java - Numerical Problems](#)

[Java - Combinatorial Problems](#)

[Java - Graph Problems](#)

[Java - Hard Graph Problems](#)

[Java - Computation Geometry](#)

[Java - Sets & Strings](#)

Swap the first variable with second variable. Now again we perform bitwise XOR operation on the same variables and store it in second variable. Finally bitwise XOR is performed on first and second variable and result is stored in first variable. Hence we get the swapped value as an output.

Here is the source code of the Java Program to Swap the Contents of two Numbers using Bitwise XOR Operation. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Swap_BitwiseXOR
3. {
4.     public static void main(String args[])
5.     {
6.         int m, n;
7.         Scanner s = new Scanner(System.in);
8.         System.out.print("Enter the first number:");
9.         m = s.nextInt();
10.        System.out.print("Enter the second number:");
11.        n = s.nextInt();
12.        m = m ^ n;
13.        n = m ^ n;
14.        m = m ^ n;
15.        System.out.println("After Swapping");
16.        System.out.println("First number:"+m);
17.        System.out.println("Second number:"+n);
18.    }
19. }
```

Output:

```
$ javac Swap_BitwiseXOR.java
$ java Swap_BitwiseXOR
```

```
Enter the first number:6
Enter the second number:7
After Swapping
First number:7
Second number:6
```

Sanfoundry Global Education & Learning Series – 1000 Java Programs.

Software Productivity
GDB Assignment

bytecode.in

Get 100 % Job
Placement

Learn Data Science, Data
Visualization Skills,
Machine Learning from
World-Class Mentors.

OPEN

Sanfoundry is **No. 1** choice for Deep
Hands-ON Trainings in **SAN, Linux & C,**
Kernel Programming. Our Founder has
trained employees of almost all Top
Companies in India such as VMware,

[C Interview Questions](#)
[C++ Questions](#)
[Linux MCQs](#)
[C# Quiz](#)
[Java MCQs](#)
[JavaScript MCQs](#)
[SAN Questions](#)
[PHP Questions](#)
[Python Quiz](#)

Computer Science Questions

[Operating System Quiz](#)
[Computer Architecture MCQs](#)
[Software Architecture MCQs](#)
[Software Engineering MCQs](#)
[Artificial Intelligence MCQs](#)
[LISP Programming MCQs](#)
[Database Management MCQs](#)
[Computer Network MCQs](#)
[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)
[C - Arrays](#)
[C - Matrix](#)
[C - Strings](#)
[C - Bitwise Operations](#)
[C - Linked Lists](#)
[C - Stacks & Queues](#)
[C - Searching & Sorting](#)
[C - Trees](#)
[C - Strings](#)
[C - File Handling](#)
[C - Mathematical Functions](#)
[C - Puzzles & Games](#)
[C Programs - Recursion](#)
[C Programs - No Recursion](#)

Java Program to Find Largest Between Three Numbers Using Ternary Operator

This is a Java Program to Find Largest Between Three Numbers Using Ternary Operator.

Enter any three integer numbers as an input from which you want to find the largest integer. After that we use ternary operator(?, >, :) to find largest number as the output.

Here is the source code of the Java Program to Find Largest Between Three Numbers Using Ternary Operator. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Largest_Ternary
3. {
4.     public static void main(String[] args)
5.     {
6.         int a, b, c, d;
7.         Scanner s = new Scanner(System.in);
8.         System.out.println("Enter all three numbers:");
9.         a = s.nextInt();
10.        b = s.nextInt();
11.        c = s.nextInt();
12.        d = c > (a > b ? a : b) ? c : ((a > b) ? a : b);
13.        System.out.println("Largest Number:" +d);
14.    }
15. }
```

Output:

```
$ javac Largest_Ternary.java
$ java Largest_Ternary

Enter all three numbers:
5
6
7
Largest Number:7
```

Developer Tracks
[SAN Developer](#)
[Linux Kernel Developer](#)
[Linux Driver Developer](#)
[Linux Network Developer](#)

Live Training Photos
Mentoring
Software Productivity
GDB Assignment



Boost your career with Simplilearn's **Artificial Intelligence Engineer** Master's Program and Job Assist. ₹44,999 GET STARTED

Sanfoundry is **No. 1** choice for Deep Hands-ON Trainings in **SAN, Linux & C, Kernel Programming**. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.

[PHP Questions](#)

[Python Quiz](#)

Computer Science Questions

[Operating System Quiz](#)

[Computer Architecture MCQs](#)

[Software Architecture MCQs](#)

[Software Engineering MCQs](#)

[Artificial Intelligence MCQs](#)

[LISP Programming MCQs](#)

[Database Management MCQs](#)

[Computer Network MCQs](#)

[Microprocessor MCQs](#)

C Programming Examples

[Simple C Programs](#)

[C - Arrays](#)

[C - Matrix](#)

[C - Strings](#)

[C - Bitwise Operations](#)

[C - Linked Lists](#)

[C - Stacks & Queues](#)

[C - Searching & Sorting](#)

[C - Trees](#)

[C - Strings](#)

[C - File Handling](#)

[C - Mathematical Functions](#)

[C - Puzzles & Games](#)

[C Programs - Recursion](#)

[C Programs - No Recursion](#)

Java Algorithms

[Java - Numerical Problems](#)

[Java - Combinatorial Problems](#)

[Java - Graph Problems](#)

[Java - Hard Graph Problems](#)

[Java - Computation Geometry](#)

Enter any binary number as an input. After that we perform operations like modulo and division to convert it into gray code.

Here is the source code of the Java Program to Convert Binary Code of a Number into its Equivalent Gray's Code Without Using Recursion. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import static java.lang.StrictMath.pow;
2. import java.util.Scanner;
3. public class Binary_Gray
4. {
5.     public static void main(String[] args)
6.     {
7.         int a, b, x, result = 0, i = 0;
8.         Scanner s = new Scanner(System.in);
9.         System.out.print("Enter Binary number:");
10.        x = s.nextInt();
11.        while(x != 0)
12.        {
13.            a = x % 10;
14.            x = x / 10;
15.            b = x % 10;
16.            if((a & ~b) == 1 || (~a & b) == 1)
17.            {
18.                result = (int) (result + pow(10,i));
19.            }
20.            i++;
21.        }
22.        System.out.println("Gray Code:"+result);
23.    }
24. }
```

Output:

```
$ javac Binary_Gray.java
$ java Binary_Gray
```

```
Enter Binary number:1001
Gray Code:1101
```

[mentoring](#)

[Software Productivity](#)

[GDB Assignment](#)

Sanfoundry is **No. 1** choice for Deep Hands-ON Trainings in **SAN, Linux & C, Kernel Programming**. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.

Home
Fundamentals
Constructors
Exception Handling
Threads
String Functions
Generics
Collections & Util Package
Nested Classes
Networking
File I/O Operations
Java Annotations
JDBC Examples
Spring Core
Spring Boot
Hibernate
Java Interview Questions
Java Interview Programs
Java Restful Web Services
Java 8 Features
JSON in Java

Program: How to get distinct elements from an array by avoiding duplicate elements?

Description:

The below example shows how to avoid duplicate elements from an array and display only distinct elements. Please use only arrays to process it.

Code:

```
1 package com.java2novice.algos;
2
3 public class MyDistinctElements {
4
5     public static void printDistinctElements(int[] arr){
6
7         for(int i=0;i<arr.length;i++){
8             boolean isDistinct = false;
9             for(int j=0;j<i;j++){
10                 if(arr[i] == arr[j]){
11                     isDistinct = true;
12                     break;
13                 }
14             }
15             if(!isDistinct){
16                 System.out.print(arr[i]+ " ");
17             }
18         }
19     }
20
21     public static void main(String a[]){
22
23         int[] nums = {5,2,7,2,4,7,8,2,3};
24         MyDistinctElements.printDistinctElements(nums);
25     }
26 }
```

Output:

5 2 7 4 8 3

 Google Custom Search

Boost your career
with Simplilearn's

Artificial Intelligence Engineer

Master's Program
and Job Assist.

GET STARTED

simplilearn



Java2Novice - YouTube Channel

Python Tutorial for Beginners



Level up your code
with a Pro Java IDE

Advertise Here

```

// Sorting array to bring duplicates together
Arrays.sort(numbersWithDuplicates);

int[] result = new int[numbersWithDuplicates.length];
int previous = numbersWithDuplicates[0];
result[0] = previous;

for (int i = 1; i < numbersWithDuplicates.length; i++) {
    int ch = numbersWithDuplicates[i];

    if (previous != ch) {
        result[i] = ch;
    }
    previous = ch;
}

return result;
}
}

Output :
Initial Array : [1, 1, 2, 2, 3, 4, 5]
After removing Duplicates : [1, 0, 2, 0, 3, 4, 5]
Initial Array : [1, 1, 1, 1, 1, 1, 1]
After removing Duplicates : [1, 0, 0, 0, 0, 0, 0]
Initial Array : [1, 2, 3, 4, 5, 6, 7]
After removing Duplicates : [1, 2, 3, 4, 5, 6, 7]

```

How to Implement Binary Search Tree in Java? Example

[Read Next Story >](#)

A STRUCTURE
ALGORITHM
IN JAVA
SECOND EDITION



Build your API with SPRING



Download
The E-book

Building a REST API with
Spring 4?

Email address...

Submit

Categories

- ▶ [courses](#) (61)
- ▶ [SQL](#) (55)
- ▶ [linux](#) (38)
- ▶ [database](#) (34)
- ▶ [Eclipse](#) (28)
- ▶ [Java Certification OCPJP SCJP](#) (24)
- ▶ [JVM Internals](#) (22)
- ▶ [jQuery](#) (17)
- ▶ [REST](#) (16)
- ▶ [Maven](#) (12)
- ▶ [Testing](#) (11)
- ▶ [general](#) (10)

Blog Archive

- ▶ [2019](#) (112)
- ▶ [2018](#) (108)

String Functions
Generics
Collections & Util Package
Nested Classes
Networking
File I/O Operations
Java Annotations
JDBC Examples
Spring Core
Spring Boot
Hibernate
Java Interview Questions
Java Interview Programs
Java Restful Web Services
Java 8 Features
JSON in Java
JUnit
Java Design Patterns
Search Algorithms
Sorting Algorithms
Data Structures

```

2
3 import java.util.Stack;
4
5 public class StackSort {
6
7     public static Stack<Integer> sortStack(Stack<Integer> input){
8
9         Stack<Integer> tmpStack = new Stack<Integer>();
10        System.out.println("===== debug logs =====");
11        while(!input.isEmpty()) {
12            int tmp = input.pop();
13            System.out.println("Element taken out: "+tmp);
14            while(!tmpStack.isEmpty() && tmpStack.peek() > tmp) {
15                input.push(tmpStack.pop());
16            }
17            tmpStack.push(tmp);
18            System.out.println("input: "+input);
19            System.out.println("tmpStack: "+tmpStack);
20        }
21        System.out.println("===== debug logs ended =====");
22        return tmpStack;
23    }
24
25    public static void main(String a[]){
26
27        Stack<Integer> input = new Stack<Integer>();
28        input.add(34);
29        input.add(3);
30        input.add(31);
31        input.add(98);
32        input.add(92);
33        input.add(23);
34        System.out.println("input: "+input);
35        System.out.println("final sorted list: "+sortStack(input));
36    }
37}

```

Output:

```

input: [34, 3, 31, 98, 92, 23]
===== debug logs =====
Element taken out: 23
input: [34, 3, 31, 98, 92]
tmpStack: [23]
Element taken out: 92
input: [34, 3, 31, 98]
tmpStack: [23, 92]

```

Java2Novice - YouTube Channel

Python Tutorial for Beginners



Level up your code
with a Pro Java IDE
IntelliJ IDEA
[Get It now](#)

Advertise Here

Knowledge Centre

Purpose of garbage collection

The garbage collection process is to identify the objects which are no longer referenced or needed by a program so that their resources can be reclaimed and reused. These identified objects will be discarded.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

25. What is difference between Array and ArrayList? When will you use Array over ArrayList?

Arrays can contain primitive or Objects whereas ArrayList can contain only Objects.

Arrays are fixed size whereas ArrayList size is dynamic.

Arrays doesn't provide a lot of features like ArrayList, such as addAll, removeAll, iterator etc.

Although ArrayList is the obvious choice when we work on list, there are few times when array are good to use.

- If the size of list is fixed and mostly used to store and traverse them.
- For list of primitive data types, although Collections use autoboxing to reduce the coding effort but still it makes them slow when working on fixed size primitive data types.
- If you are working on fixed multi-dimensional situation, using `[][]` is far more easier than `List<List<>>`

26. What is difference between ArrayList and LinkedList?

ArrayList and LinkedList both implement List interface but there are some differences between them.

1. ArrayList is an index based data structure backed by Array, so it provides random access to its elements with performance as $O(1)$ but LinkedList stores data as list of nodes where every node is linked to its previous and next node. So even though there is a method to get the element using index, internally it traverse from start to reach at the index node and then return the element, so performance is $O(n)$ that is slower than ArrayList.
2. Insertion, addition or removal of an element is faster in LinkedList compared to ArrayList because there is no concept of resizing array or updating index when element is added in middle.
3. LinkedList consumes more memory than ArrayList because every node in LinkedList stores reference of previous and next elements.

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

It does not support the add or addAll operations.

22. What is difference between HashMap and Hashtable?

HashMap and Hashtable both implements Map interface and looks similar, however there are following difference between HashMap and Hashtable.

1. HashMap allows null key and values whereas Hashtable doesn't allow null key and values.
2. Hashtable is synchronized but HashMap is not synchronized. So HashMap is better for single threaded environment, Hashtable is suitable for multi-threaded environment.
3. **LinkedHashMap** was introduced in Java 1.4 as a subclass of HashMap, so incase you want iteration order, you can easily switch from HashMap to LinkedHashMap but that is not the case with Hashtable whose iteration order is unpredictable.
4. HashMap provides Set of keys to iterate and hence it's fail-fast but Hashtable provides Enumeration of keys that doesn't support this feature.
5. Hashtable is considered to be legacy class and if you are looking for modifications of Map while iterating, you should use ConcurrentHashMap.

23. How to decide between HashMap and TreeMap?

For inserting, deleting, and locating elements in a Map, the HashMap offers the best alternative. If, however, you need to traverse the keys in a sorted order, then TreeMap is your better alternative.

Depending upon the size of your collection, it may be faster to add elements to a HashMap, then convert the map to a TreeMap for sorted key traversal.

24. What are similarities and difference between ArrayList and Vector?

ArrayList and Vector are similar classes in many ways.

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

iterating, you should use ConcurrentHashMap.

23. How to decide between HashMap and TreeMap?

For inserting, deleting, and locating elements in a Map, the HashMap offers the best alternative. If, however, you need to traverse the keys in a sorted order, then TreeMap is your better alternative.

Depending upon the size of your collection, it may be faster to add elements to a HashMap, then convert the map to a TreeMap for sorted key traversal.

24. What are similarities and difference between ArrayList and Vector?

ArrayList and Vector are similar classes in many ways.

1. Both are index based and backed up by an array internally.
2. Both maintains the order of insertion and we can get the elements in the order of insertion.
3. The iterator implementations of ArrayList and Vector both are fail-fast by design.
4. ArrayList and Vector both allows null values and random access to element using index number.

These are the differences between ArrayList and Vector.

1. Vector is synchronized whereas ArrayList is not synchronized. However if you are looking for modification of list while iterating, you should use CopyOnWriteArrayList.
2. ArrayList is faster than Vector because it doesn't have any overhead because of synchronization.
3. ArrayList is more versatile because we can get synchronized list or read-only list from it easily using Collections utility class.

25. What is difference between Array and ArrayList? When will you use Array over ArrayList?

Is there a better company
you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

25. What is difference between Array and ArrayList? When will you use Array over ArrayList?

Arrays can contain primitive or Objects whereas ArrayList can contain only Objects.

Arrays are fixed size whereas ArrayList size is dynamic.

Arrays doesn't provide a lot of features like ArrayList, such as addAll, removeAll, iterator etc.

Although ArrayList is the obvious choice when we work on list, there are few times when array are good to use.

- If the size of list is fixed and mostly used to store and traverse them.
- For list of primitive data types, although Collections use autoboxing to reduce the coding effort but still it makes them slow when working on fixed size primitive data types.
- If you are working on fixed multi-dimensional situation, using `[][]` is far more easier than `List<List<>>`

26. What is difference between ArrayList and LinkedList?

ArrayList and LinkedList both implement List interface but there are some differences between them.

1. ArrayList is an index based data structure backed by Array, so it provides random access to its elements with performance as $O(1)$ but LinkedList stores data as list of nodes where every node is linked to its previous and next node. So even though there is a method to get the element using index, internally it traverse from start to reach at the index node and then return the element, so performance is $O(n)$ that is slower than ArrayList.
2. Insertion, addition or removal of an element is faster in LinkedList compared to ArrayList because there is no concept of resizing array or updating index when element is added in middle.
3. LinkedList consumes more memory than ArrayList because every node in LinkedList stores reference of previous and next elements.

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

It does not support the add or addAll operations.

22. What is difference between HashMap and Hashtable?

HashMap and Hashtable both implements Map interface and looks similar, however there are following difference between HashMap and Hashtable.

1. HashMap allows null key and values whereas Hashtable doesn't allow null key and values.
2. Hashtable is synchronized but HashMap is not synchronized. So HashMap is better for single threaded environment, Hashtable is suitable for multi-threaded environment.
3. **LinkedHashMap** was introduced in Java 1.4 as a subclass of HashMap, so incase you want iteration order, you can easily switch from HashMap to LinkedHashMap but that is not the case with Hashtable whose iteration order is unpredictable.
4. HashMap provides Set of keys to iterate and hence it's fail-fast but Hashtable provides Enumeration of keys that doesn't support this feature.
5. Hashtable is considered to be legacy class and if you are looking for modifications of Map while iterating, you should use ConcurrentHashMap.

23. How to decide between HashMap and TreeMap?

For inserting, deleting, and locating elements in a Map, the HashMap offers the best alternative. If, however, you need to traverse the keys in a sorted order, then TreeMap is your better alternative.

Depending upon the size of your collection, it may be faster to add elements to a HashMap, then convert the map to a TreeMap for sorted key traversal.

24. What are similarities and difference between ArrayList and Vector?

ArrayList and Vector are similar classes in many ways.

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

iterating, you should use ConcurrentHashMap.

23. How to decide between HashMap and TreeMap?

For inserting, deleting, and locating elements in a Map, the HashMap offers the best alternative. If, however, you need to traverse the keys in a sorted order, then TreeMap is your better alternative.

Depending upon the size of your collection, it may be faster to add elements to a HashMap, then convert the map to a TreeMap for sorted key traversal.

24. What are similarities and difference between ArrayList and Vector?

ArrayList and Vector are similar classes in many ways.

1. Both are index based and backed up by an array internally.
2. Both maintains the order of insertion and we can get the elements in the order of insertion.
3. The iterator implementations of ArrayList and Vector both are fail-fast by design.
4. ArrayList and Vector both allows null values and random access to element using index number.

These are the differences between ArrayList and Vector.

1. Vector is synchronized whereas ArrayList is not synchronized. However if you are looking for modification of list while iterating, you should use CopyOnWriteArrayList.
2. ArrayList is faster than Vector because it doesn't have any overhead because of synchronization.
3. ArrayList is more versatile because we can get synchronized list or read-only list from it easily using Collections utility class.

25. What is difference between Array and ArrayList? When will you use Array over ArrayList?

Is there a better company
you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

25. What is difference between Array and ArrayList? When will you use Array over ArrayList?

Arrays can contain primitive or Objects whereas ArrayList can contain only Objects.

Arrays are fixed size whereas ArrayList size is dynamic.

Arrays doesn't provide a lot of features like ArrayList, such as addAll, removeAll, iterator etc.

Although ArrayList is the obvious choice when we work on list, there are few times when array are good to use.

- If the size of list is fixed and mostly used to store and traverse them.
- For list of primitive data types, although Collections use autoboxing to reduce the coding effort but still it makes them slow when working on fixed size primitive data types.
- If you are working on fixed multi-dimensional situation, using `[][]` is far more easier than `List<List<>>`

26. What is difference between ArrayList and LinkedList?

ArrayList and LinkedList both implement List interface but there are some differences between them.

1. ArrayList is an index based data structure backed by Array, so it provides random access to its elements with performance as $O(1)$ but LinkedList stores data as list of nodes where every node is linked to its previous and next node. So even though there is a method to get the element using index, internally it traverse from start to reach at the index node and then return the element, so performance is $O(n)$ that is slower than ArrayList.
2. Insertion, addition or removal of an element is faster in LinkedList compared to ArrayList because there is no concept of resizing array or updating index when element is added in middle.
3. LinkedList consumes more memory than ArrayList because every node in LinkedList stores reference of previous and next elements.

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

It does not support the add or addAll operations.

22. What is difference between HashMap and Hashtable?

HashMap and Hashtable both implements Map interface and looks similar, however there are following difference between HashMap and Hashtable.

1. HashMap allows null key and values whereas Hashtable doesn't allow null key and values.
2. Hashtable is synchronized but HashMap is not synchronized. So HashMap is better for single threaded environment, Hashtable is suitable for multi-threaded environment.
3. **LinkedHashMap** was introduced in Java 1.4 as a subclass of HashMap, so incase you want iteration order, you can easily switch from HashMap to LinkedHashMap but that is not the case with Hashtable whose iteration order is unpredictable.
4. HashMap provides Set of keys to iterate and hence it's fail-fast but Hashtable provides Enumeration of keys that doesn't support this feature.
5. Hashtable is considered to be legacy class and if you are looking for modifications of Map while iterating, you should use ConcurrentHashMap.

23. How to decide between HashMap and TreeMap?

For inserting, deleting, and locating elements in a Map, the HashMap offers the best alternative. If, however, you need to traverse the keys in a sorted order, then TreeMap is your better alternative.

Depending upon the size of your collection, it may be faster to add elements to a HashMap, then convert the map to a TreeMap for sorted key traversal.

24. What are similarities and difference between ArrayList and Vector?

ArrayList and Vector are similar classes in many ways.

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

iterating, you should use ConcurrentHashMap.

23. How to decide between HashMap and TreeMap?

For inserting, deleting, and locating elements in a Map, the HashMap offers the best alternative. If, however, you need to traverse the keys in a sorted order, then TreeMap is your better alternative.

Depending upon the size of your collection, it may be faster to add elements to a HashMap, then convert the map to a TreeMap for sorted key traversal.

24. What are similarities and difference between ArrayList and Vector?

ArrayList and Vector are similar classes in many ways.

1. Both are index based and backed up by an array internally.
2. Both maintains the order of insertion and we can get the elements in the order of insertion.
3. The iterator implementations of ArrayList and Vector both are fail-fast by design.
4. ArrayList and Vector both allows null values and random access to element using index number.

These are the differences between ArrayList and Vector.

1. Vector is synchronized whereas ArrayList is not synchronized. However if you are looking for modification of list while iterating, you should use CopyOnWriteArrayList.
2. ArrayList is faster than Vector because it doesn't have any overhead because of synchronization.
3. ArrayList is more versatile because we can get synchronized list or read-only list from it easily using Collections utility class.

25. What is difference between Array and ArrayList? When will you use Array over ArrayList?

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.



LEARN
FILMMAKING
BY MAKING FILMS

AT THE
NEW YORK
FILM ACADEMY

[LEARN MORE](#)

25. What is difference between Array and ArrayList? When will you use Array over ArrayList?

Arrays can contain primitive or Objects whereas ArrayList can contain only Objects.

Arrays are fixed size whereas ArrayList size is dynamic.

Arrays doesn't provide a lot of features like ArrayList, such as addAll, removeAll, iterator etc.

Although ArrayList is the obvious choice when we work on list, there are few times when array are good to use.

- If the size of list is fixed and mostly used to store and traverse them.
- For list of primitive data types, although Collections use autoboxing to reduce the coding effort but still it makes them slow when working on fixed size primitive data types.
- If you are working on fixed multi-dimensional situation, using `[][]` is far more easier than `List<List<>>`

26. What is difference between ArrayList and LinkedList?

ArrayList and LinkedList both implement List interface but there are some differences between them.

1. ArrayList is an index based data structure backed by Array, so it provides random access to its elements with performance as $O(1)$ but LinkedList stores data as list of nodes where every node is linked to its previous and next node. So even though there is a method to get the element using index, internally it traverse from start to reach at the index node and then return the element, so performance is $O(n)$ that is slower than ArrayList.
2. Insertion, addition or removal of an element is faster in LinkedList compared to ArrayList because there is no concept of resizing array or updating index when element is added in middle.
3. LinkedList consumes more memory than ArrayList because every node in LinkedList stores reference of previous and next elements.

Is there a better company you could be working for?

JOB SEARCH

Click here to find out.

upGrad

In association
with



Job
Guarantee
with
HDFC Life

Tell Me How!

```
public class CreateNewFile {  
  
    /**  
     * This class shows how to create a File in Java  
     * @param args  
     * @throws IOException  
     */  
    public static void main(String[] args) throws IOException {  
        String fileSeparator = System.getProperty("file.separator");  
  
        //absolute file name with path  
        String absoluteFilePath =  
fileSeparator+"Users"+fileSeparator+"pankaj"+fileSeparator+"file.txt";  
        File file = new File(absoluteFilePath);  
        if(file.createNewFile()){  
            System.out.println(absoluteFilePath+" File Created");  
        }else System.out.println("File "+absoluteFilePath+" already exists");  
  
        //file name only  
        file = new File("file.txt");  
        if(file.createNewFile()){  
            System.out.println("file.txt File Created in Project root directory");  
        }else System.out.println("File file.txt already exists in the project root  
directory");  
    }  
}
```

When we execute the above program from Eclipse IDE for the first time, the below output is produced.



In association
with



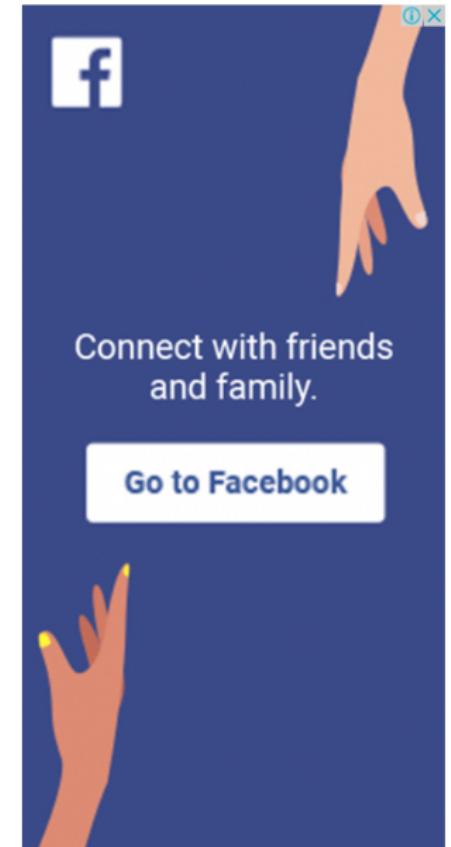
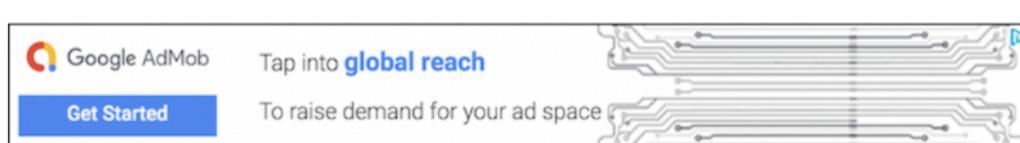
Job
Guarantee
with
HDFC Life

Tell Me How!

```
//file name only
file = new File("file.txt");
if(file.createNewFile()){
    System.out.println("file.txt File Created in Project root directory");
}else System.out.println("File file.txt already exists in the project root
directory");

//relative path
String relativePath = "tmp"+fileSeparator+"file.txt";
file = new File(relativePath);
if(file.createNewFile()){
    System.out.println(relativePath+" File Created in Project root
directory");
}else System.out.println("File "+relativePath+" already exists in the
project root directory");
}
```

When we execute the above program from Eclipse IDE for the first time, the below output is produced.



upGrad

In association
with



Job
Guarantee
with
HDFC Life

Tell Me How!

```
/Users/pankaj/file.txt File Created
file.txt File Created in Project root directory
Exception in thread "main"
java.io.IOException: No such file or directory
    at java.io.UnixFileSystem.createFileExclusively(Native Method)
    at java.io.File.createNewFile(File.java:947)
    at com.journaldev.files.CreateNewFile.main(CreateNewFile.java:32)
```

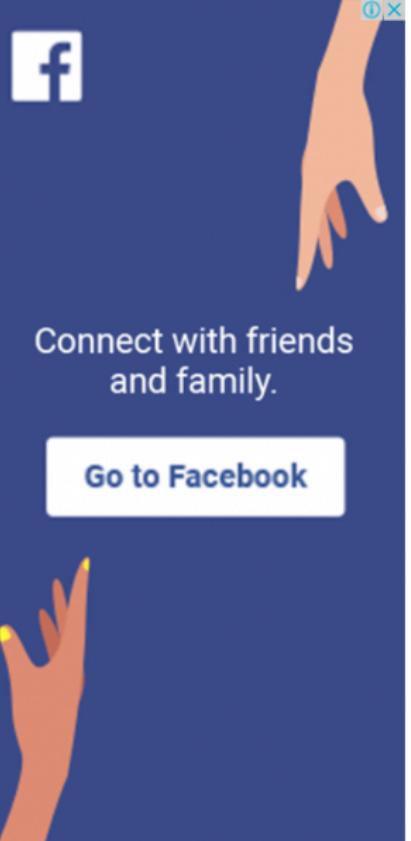
For the relative path, it throws IOException because `tmp` directory is not present in the project root folder.

So it's clear that `createNewFile()` just tries to create the file and any directory either absolute or relative should be present already, else it throws `IOException`.

So I created "tmp" directory in the project root and executed the program again, here is the output.

```
File /Users/pankaj/file.txt already exists
File file.txt already exists in the project root directory
tmp/file.txt File Created in Project root directory
```

First two files were already present, so `createNewFile()` returns `false`, third file was created in tmp directory and returns `true`.





Job
Guarantee
with
HDFC Life

Tell Me How!

```
File /Users/pankaj/file.txt already exists
File file.txt already exists in project root directory
file.createNewFile();
```

2. FileOutputStream.write(byte[] b)

If you want to create a new file and at the same time write some data into it, you can use [FileOutputStream](#) write method. Below is a simple code snippet to show its usage. The rules for absolute path and relative path discussed above is applicable in this case too.

```
String fileData = "Pankaj Kumar";
FileOutputStream fos = new FileOutputStream("name.txt");
fos.write(fileData.getBytes());
fos.flush();
fos.close();
```

3. Java NIO Files.write()

We can use [Java NIO Files](#) class to create a new file and write some data into it. This is a good option because we don't have to worry about closing IO resources.

```
String fileData = "Pankaj Kumar";
Files.write(Paths.get("name.txt"), fileData.getBytes());
```

That's all for creating a new file in the java program.



IT Help Desk Software
ManageEngine ServiceDesk Plus Try for free

Best IT HelpDesk Software

ManageEngine
ServiceDesk Plus

100,000+ Successful IT Help Desks use ServiceDesk Plus to Supercharge their IT Help Desk



upGrad
In association with
HDFC Life
Sarvashaa ke jyoti!

Job
Guarantee
with
HDFC Life

Tell Me How!

① X

There are three popular methods to create file in java. Let's look at them one by one.

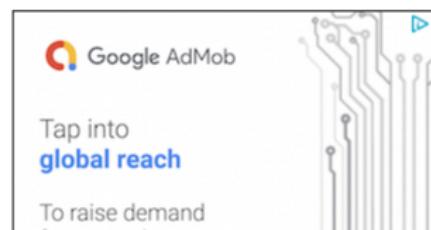
1. `File.createNewFile()`

`java.io.File` class can be used to create a new File in Java. When we initialize File object, we provide the file name and then we can call `createNewFile()` method to create new file in Java.

File `createNewFile()` method returns `true` if new file is created and `false` if file already exists. This method also throws `java.io.IOException` when it's not able to create the file. The files created is empty and of zero bytes.

When we create the File object by passing the file name, it can be with **absolute path**, or we can only provide the file name or we can provide the relative path.

For a non-absolute path, File object tries to locate files in the project root directory. If we run the program from command line, for the non-absolute path, File object tries to locate files from the current directory.



Share your family photos on Facebook.

f

① X

BRANDER



РАЗРАБОТКА
МОБИЛЬНЫХ



```
//relative path  
file = new File("temp/file.txt");  
if(file.delete()){  
    System.out.println("temp/file.txt File deleted from Project root  
directory");  
}else System.out.println("File temp/file.txt doesn't exist in the project root  
directory");  
  
//delete empty directory  
file = new File("temp");  
if(file.delete()){  
    System.out.println("temp directory deleted from Project root directory");  
}else System.out.println("temp directory doesn't exist or not empty in the  
project root directory");  
  
//try to delete directory with files  
file = new File("/Users/pankaj/project");  
if(file.delete()){  
    System.out.println("/Users/pankaj/project directory deleted from Project  
root directory");  
}else System.out.println("/Users/pankaj/project directory doesn't exist or not  
empty");
```

Below is the output when we execute the above java delete file example program for the first time.

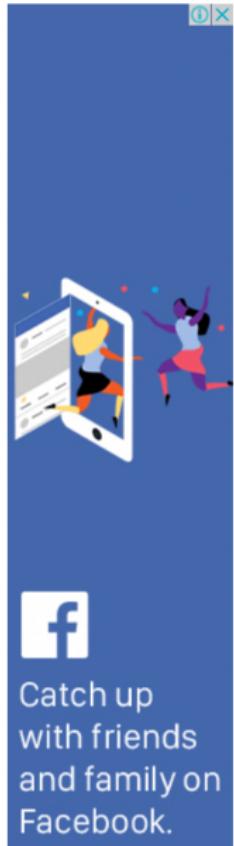


Best IT HelpDesk Software

ManageEngine
ServiceDesk Plus

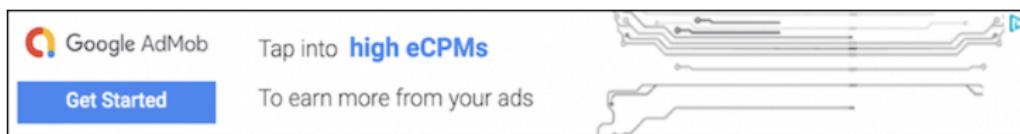
100,000+ Successful IT Help
Desks use ServiceDesk Plus to
Supercharge their IT Help Desk





```
if(file.delete()){
    System.out.println("/Users/pankaj/project directory deleted from Project
root directory");
} else System.out.println("/Users/pankaj/project directory doesn't exist or not
empty");
```

Below is the output when we execute the above java delete file example program for the first time.



```
/Users/pankaj/file.txt File deleted
file.txt File deleted from Project root directory
temp/file.txt File deleted from Project root directory
temp directory deleted from Project root directory
/Users/pankaj/project directory doesn't exist or not empty
```

Note that temp directory had file.txt and it got deleted first and then directory was empty and got deleted successfully. /Users/pankaj/project was not empty and hence not deleted.

In the subsequent run of the same program, the output is:

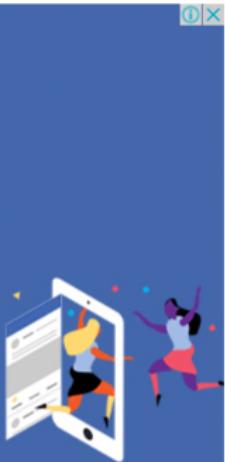


Best IT HelpDesk Software

ManageEngine
ServiceDesk Plus

100,000+ Successful IT Help Desks use ServiceDesk Plus to Supercharge their IT Help Desk





Catch up
with friends
and family on
Facebook.

```
package com.journaldev.files;

import java.io.File;

public class DeletefileJava {

    /**
     * This class shows how to delete a File in Java
     * @param args
     */
    public static void main(String[] args) {
        //absolute file name with path
        File file = new File("/Users/pankaj/file.txt");
        if(file.delete()){
            System.out.println("/Users/pankaj/file.txt File deleted");
        }else System.out.println("File /Users/pankaj/file.txt doesn't exist");

        //file name only
        file = new File("file.txt");
        if(file.delete()){
            System.out.println("file.txt File deleted from Project root directory");
        }else System.out.println("File file.txt doesn't exist in the project root
    }
}
```

Below is the output when we execute the above java delete file example program for the first time.

An advertisement for ManageEngine ServiceDesk Plus. It features three green tags with the word 'FREE' and a map background. Text includes 'IT Help Desk Software', 'ManageEngine ServiceDesk Plus', and a 'Try for free' button.

Best IT HelpDesk Software



5 Things you Need to Know About Java Thread

5. Explain Java 7 ARM Feature and multi-catch block?

If you are catching a lot of exceptions in a single try block, you will notice that catch block code looks very ugly and mostly consists of redundant code to log the error, keeping this in mind Java 7 one of the feature was multi-catch block where we can catch multiple exceptions in a single catch block. The catch block with this feature looks like below:

```
catch(IOException | SQLException | Exception ex){  
    logger.error(ex);  
    throw new MyException(ex.getMessage());  
}
```

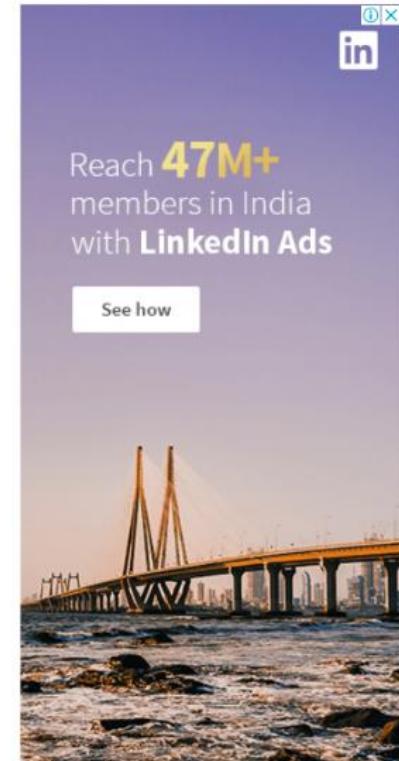
Most of the time, we use finally block just to close the resources and sometimes we forget to close them and get runtime exceptions when the resources are exhausted. These exceptions are hard to debug and we might need to look into each place where we are using that type of resource to make sure we are closing it. So java 7 one of the improvement was **try-with-resources** where we can create a resource in the try statement itself and use it inside the try-catch block. When the execution comes out of try-catch block, runtime environment automatically close these resources. Sample of try-catch block with this improvement is:

```
try (MyResource mr = new MyResource()) {  
    System.out.println("MyResource created in try-with-resources");  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

Read more about this at [Java 7 ARM](#).

6. What is difference between Checked and Unchecked Exception in Java?

1. Checked Exceptions should be handled in the code using try-catch block or else method should use throws keyword to let the caller know about the checked exceptions that might be thrown from the method. Unchecked



Read more about this at [Java 7 ARM](#).

6. What is difference between Checked and Unchecked Exception in Java?

1. Checked Exceptions should be handled in the code using try-catch block or else method should use throws keyword to let the caller know about the checked exceptions that might be thrown from the method. Unchecked Exceptions are not required to be handled in the program or to mention them in throws clause of the method.
2. **Exception** is the super class of all checked exceptions whereas **RuntimeException** is the super class of all unchecked exceptions. Note that RuntimeException is the child class of Exception.
3. Checked exceptions are error scenarios that requires to be handled in the code, or else you will get compile time error. For example, if you use FileReader to read a file, it throws **FileNotFoundException** and we must catch it in the try-catch block or throw it again to the caller method. Unchecked exceptions are mostly caused by poor programming, for example **NullPointerException** when invoking a method on an object reference without making sure that it's not null. For example, I can write a method to remove all the vowels from the string. It's the caller responsibility to make sure not to pass null string. I might change the method to handle these scenarios but ideally the caller should take care of this.

7. What is difference between throw and throws keyword in Java?

throws keyword is used with method signature to declare the exceptions that the method might throw whereas throw keyword is used to disrupt the flow of program and handing over the exception object to runtime to handle it.

8. How to write custom exception in Java?

We can extend **Exception** class or any of its subclasses to create our custom exception class. The custom exception class can have its own variables and methods that we can use to pass error codes or other exception related information to the exception handler.

A simple example of custom exception is shown below.



12. What happens when exception is thrown by main method?

When exception is thrown by main() method, Java Runtime terminates the program and print the exception message and stack trace in system console.

13. Can we have an empty catch block?

We can have an empty catch block but it's the example of worst programming. We should never have empty catch block because if the exception is caught by that block, we will have no information about the exception and it will be a nightmare to debug it. There should be at least a logging statement to log the exception details in console or log files.

14. Provide some Java Exception Handling Best Practices?

Some of the best practices related to Java Exception Handling are:

- Use Specific Exceptions for ease of debugging.
- Throw Exceptions Early (Fail-Fast) in the program.
- Catch Exceptions late in the program, let the caller handle the exception.
- Use Java 7 ARM feature to make sure resources are closed or use finally block to close them properly.
- Always log exception messages for debugging purposes.
- Use multi-catch block for cleaner close.
- Use custom exceptions to throw single type of exception from your application API.
- Follow naming convention, always end with Exception.
- Document the Exceptions Thrown by a method using @throws in javadoc.
- Exceptions are costly, so throw it only when it makes sense. Else you can catch them and provide null or empty response.

Read more about them in detail at [Java Exception Handling Best Practices](#).

15. What is the problem with below programs and how do we fix it?

In this section, we will look into some programming questions related to java exceptions.

1 What is the problem with below program?

Questions

- › Core Java Interview Questions
- › String Interview Questions
- › Multithreading Interview Questions
- › Collections Interview Questions
- › Exception Interview Questions
- › Java Programming Interview Questions
- › Java 8 Interview Questions Part 1
- › Java 8 Interview Questions Part 2
- › Servlet Interview Questions
- › JSP Interview Questions
- › Struts 2 Interview Questions
- › JDBC Interview Questions
- › Spring Interview Questions
- › Hibernate Interview Questions
- › JSF Interview Questions
- › Web Services Interview Questions
- › Scala Basic Interview Questions
- › Scala Intermediate Interview Questions
- › Scala Advanced Interview Questions
- › Scala Interview Questions
- Summary
- › Common Job Interview Questions

Miscellaneous

- › Java ClassLoader
- › String StringBuffer StringBuilder
- › Java is Pass By Value
- › Java Heap vs Stack Memory

Recommended Tutorials

- + Java Tutorials

12. What happens when exception is thrown by main method?
13. Can we have an empty catch block?
14. Provide some Java Exception Handling Best Practices?
15. What is the problem with below programs and how do we fix it?

1. What is Exception in Java?

Exception is an error event that can happen during the execution of a program and disrupts its normal flow. Exception can arise from different kind of situations such as wrong data entered by user, hardware failure, network connection failure etc.

Whenever any error occurs while executing a java statement, an exception object is created and then **JRE** tries to find exception handler to handle the exception. If suitable exception handler is found then the exception object is passed to the handler code to process the exception, known as **catching the exception**. If no handler is found then application throws the exception to runtime environment and JRE terminates the program.

Java Exception handling framework is used to handle runtime errors only, compile time errors are not handled by exception handling framework.



2. What are the Exception Handling Keywords in Java?

There are four keywords used in java exception handling.

```
01. package arraysInterviewQuestions;
02. public class PrintMissingNumbers {
03.
04.     private static void findMissingNumber(int[] number){
05.
06.         // take max length as last number in array
07.         int k[] = new int[number[number.length-1]];
08.
09.         int m=0;
10.
11.         if(number[0]!=1){
12.             for (int x = 1; x < number[0]; x++) {
13.                 k[m] = x;
14.                 m++;
15.             }
16.         }
17.
18.         for (int i = 0; i < number.length -1; i++) {
19.
20.             int j = i+1;
21.             int difference = number[j] - number[i] ;
22.
23.
24.             if(difference != 1 ){
25.
26.                 for (int x = 1; x < difference; x++) {
27.
28.                     k[m] = number[i] + x;
29.                     m++;
30.                 }
31.
32.             }
33.         }
34.
35.
36.         System.out.println("missing numbers in array ::");
37.
38.         for (int l = 0; l < m ; l++) {
39.             System.out.println( k[l]+ " ");
40.         }
41.
42.
43.
44.     public static void main(String[] args) {
45.
46.         int a[]={2,4,6,9,10,20};
47.
48.         //if Array is not sorted :To sort array use Arrays.sort(a);
49.
50.         findMissingNumber(a);
51.
52.
```

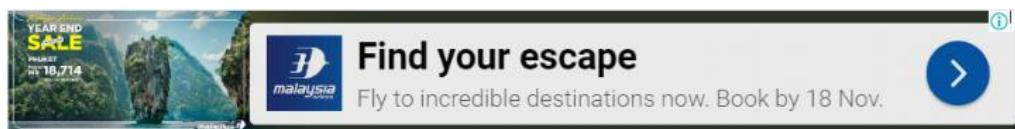


Google Custom Sea

Java Program to find missing numbers in an array

Posted by: InstanceOfJava Posted date: Jun 18, 2016 / comment : 0

- To find missing numbers in an array first we need to make sure that array is sorted.
- After sorting we need to check that array each element with next element then we can find the difference.
- if Array is not sorted :To sort array use Arrays.sort(array);
- If difference is 1 then no need to do any thing because numbers are in order.
- If difference is not equal to 1 then we need to print all those numbers or pick those numbers and place it in one array.
- this would be the logic to find missing numbers in an array
- Here there may be a chance of array not starts with 1 then we need to check first itself whether array starts with 1 or not if not we need to print 1 to starting element of array.
- for example int a[]={4,5,6,8}; then we need to print 1 2 3 7.



Lets see a java example program to find missing numbers in an array.

```
01. package arraysInterviewQuestions;  
02. public class PrintMissingNumbers {  
03.     private static void findMissingNumber(int[] number){  
04.  
05. }
```

An advertisement for AMCAT featuring a smiling man in a blue shirt flexing his right arm. The text "AMCAT Test" is at the top, followed by "Trusted by 20 Lac+ Freshers". A large orange button at the bottom right says "Enroll Now". The AMCAT logo is in the top right corner.

```
40. }
41. }
```

Output:

```
01. Enter a String to find out shortest palindrome
02. java
03. Shortest palindrome of java is avajava
```

```
1 package shortestpalindromeexample.java;
2 import java.util.Scanner;
3
4 public class ShortestPalindromeDemo {
5
6@ public static String shortestPalindrome(String str) {
7     int x=0;
8     int y=str.length()-1;
9
10    while(y>=0){
11        if(str.charAt(x)==str.charAt(y)){ // www.instanceofjava.com
12            x++;
13        }
14        y--;
15    }
16
17    if(x==str.length())
18    return str;
19
20    String suffix = str.substring(x);
21    String prefix = new StringBuilder(suffix).reverse().toString();
22    String mid = shortestPalindrome(str.substring(0, x));
23    return prefix+mid+suffix;
24 }
25
26@ public static void main(String[] args) {
27
28 Scanner in = new Scanner(System.in);
29
30 System.out.println("Enter a String to find out shortest palindrome");
31
32 String str=in.nextLine();
33
34
35 System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));
36 }
37 }
38 }
```

www.InstanceOfJava.com

Google Custom Search

SPONSORED SEARCHES

- [find smallest palindrome string](#)
- [python find in string](#)
- [java string](#)
- [java code](#)
- [learn to program java](#)

Let's be friends



Be the first of your friends to like this

```
40. }
41. }
```

Output:

```
01. Enter a String to find out shortest palindrome
02. java
03. Shortest palindrome of java is avajava
```

```
1 package shortestpalindromeexample.java;
2 import java.util.Scanner;
3
4 public class ShortestPalindromeDemo {
5
6@ public static String shortestPalindrome(String str) {
7     int x=0;
8     int y=str.length()-1;
9
10    while(y>=0){
11        if(str.charAt(x)==str.charAt(y)){ // www.instanceofjava.com
12            x++;
13        }
14        y--;
15    }
16
17    if(x==str.length())
18    return str;
19
20    String suffix = str.substring(x);
21    String prefix = new StringBuilder(suffix).reverse().toString();
22    String mid = shortestPalindrome(str.substring(0, x));
23    return prefix+mid+suffix;
24 }
25
26@ public static void main(String[] args) {
27
28 Scanner in = new Scanner(System.in);
29
30 System.out.println("Enter a String to find out shortest palindrome");
31
32 String str=in.nextLine();
33
34
35 System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));
36 }
37 }
38 }
```

www.InstanceOfJava.com

Google Custom Search

SPONSORED SEARCHES

- [find smallest palindrome string](#)
- [python find in string](#)
- [java string](#)
- [java code](#)
- [learn to program java](#)

Let's be friends



Be the first of your friends to like this

#1: Java Example program to write or save string to file using PrintWriter and java 7 try with resources.

```
1 package com.instanceofjava.writetofile;
2 import java.io.FileNotFoundException;
3 import java.io.PrintWriter;
4 /**
5  *
6  * @author www.instanceofjava.com
7  * @category: java example program
8  *
9  * Description: Write a java example program to write string to files using PrintWriter
10 *
11 */
12 public class WriteStringToFile {
13
14     public static void main(String[] args) {
15         String str="Write String to file";
16         try( PrintWriter out = new PrintWriter("data.txt") )
17     {
18
19             out.println( str );
20
21         } catch (FileNotFoundException e)
22         {
23             e.printStackTrace();
24
25         }
26     }
27 }
```

```
1 package com.instanceofjava.writetofile;
2
3 import java.io.FileNotFoundException;
4
5 /**
6  *
7  * @author www.instanceofjava.com
8  * @category: java example programs
9  *
10 * Description: Write a java example program to save string to file using PrintWriter
11 *
12 */
13
14 public class SaveStringToFile {
15
16     public static void main(String[] args) {
17         String str="Write String to file";
18         try( PrintWriter out = new PrintWriter("E:\\data.txt") ){

```



Java program to write a string to file using PrintWriter

Posted by: InstanceOfJava | Posted date: Jan 30, 2018 | comment : 0

- We can write a string to text file in java in various ways.
- Using PrintWriter we can write or append a string to the text file.
- Using println() method of PrintWriter we can save or write string to text file.
- After completion of using PrintWriter you need to close it by calling close() method.
- If you are using java 7 , by using try with resources we can implement it and closing will be taken care automatically.
- Now we will see an example program on how to write or save a string / string variable to a text file using java 7.

#1: Java Example program to write or save string to file using PrintWriter and java 7 try with resources.

```
01. package com.instanceofjava.writetofile;
02. import java.io.FileNotFoundException;
03. import java.io.PrintWriter;
04. /**
05. *
06. * @author www.instanceofjava.com
07. * @category: java example program
08. *
09. * Description: Write a java example program to write string to files using PrintWriter
10. *
11. */
12. public class WriteStringToFile {
13.
14.     public static void main(String[] args) {
15.         String str="Write String to file";
16.         try( PrintWriter out = new PrintWriter("data.txt") )
17.         {
18.             out.println(str);
19.         }
19.     }
}
```



> Java Programming Interview Questions

> Java 8 Interview Questions Part 1
> Java 8 Interview Questions Part 2

> Servlet Interview Questions

> JSP Interview Questions

> Struts 2 Interview Questions

> JDBC Interview Questions

> Spring Interview Questions

> Hibernate Interview Questions

> JSF Interview Questions

> Web Services Interview Questions

> Scala Basic Interview Questions

> Scala Intermediate Interview Questions

Questions

> Scala Advanced Interview Questions

> Scala Interview Questions

Summary

> Common Job Interview Questions

Miscellaneous

> Java ClassLoader

> String StringBuffer StringBuilder

> Java is Pass By Value

> Java Heap vs Stack Memory

Recommended Tutorials

+ Java Tutorials

+ Java EE Tutorials



1. Java Programming Interview Question 1

What is the output of the below statements?

```
String s1 = "abc";
String s2 = "abc";
System.out.println("s1 == s2 is:" + s1 == s2);
```

2. Java Programming Interview Question 2

What is the output of the below statements?

```
import
urllib2
learning

#Download SAP HANA, express edition
free_download = urllib2.urlopen(hana_express)

new_skills = ['machine learning', 'predictive analytics', 'geospatial']

upgraded_skills = learning.install_db(free_download, new_skills)
```

SAP HANA, express edition

[Free Download](#)

```
String s3 = "JournalDev";
int start = 1;
char end = 5;
System.out.println(start + end);
System.out.println(s3.substring(start, end));
```

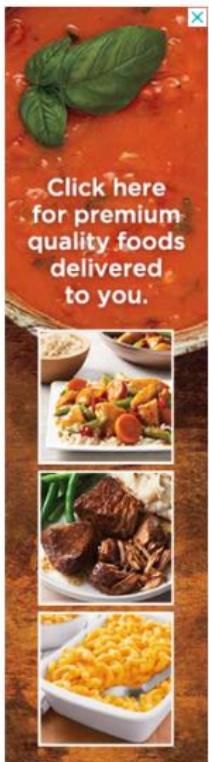
Is there a better company you could be working for?



[Click here to find out.](#)

Recommended Tutorials

- + Java Tutorials
- + Java EE Tutorials



```
String s3 = "JournalDev";
int start = 1;
char end = 5;
System.out.println(start + end);
System.out.println(s3.substring(start, end));
```

3. Java Programming Interview Question 3

What is the output of the below statements?

```
HashSet shortSet = new HashSet();
for (short i = 0; i < 100; i++) {
    shortSet.add(i);
    shortSet.remove(i - 1);
}
System.out.println(shortSet.size());
```

4. Java Programming Interview Question 4

What will be the boolean "flag" value to reach the finally block?

```
try {
    if (flag) {
        while (true) {
        }
    } else {
        System.exit(1);
    }
} finally {
```





```
try {
    if (flag) {
        while (true) {
        }
    } else {
        System.exit(1);
    }
} finally {
    System.out.println("In Finally");
}
```

5. Java Programming Interview Question 5

What will be the output of the below statements?

```
String str = null;
String str1="abc";
System.out.println(str1.equals("abc") | str.equals(null));
```

6. Java Programming Interview Question 6

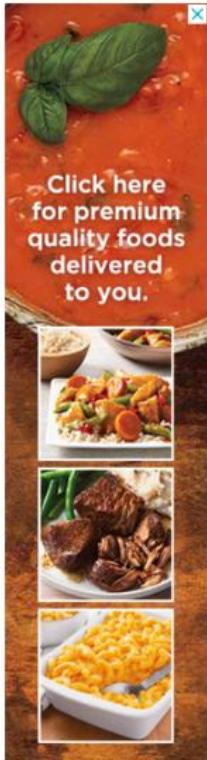
What will be the output of the below statements?

```
String x = "abc";
String y = "abc";
x.concat(y);
System.out.print(x);
```



What will be the output of below program?

```
public class MathTest {  
  
    public void main(String[] args) {  
  
        int x = 10*10-10;  
  
        System.out.println(x);  
    }  
  
}
```

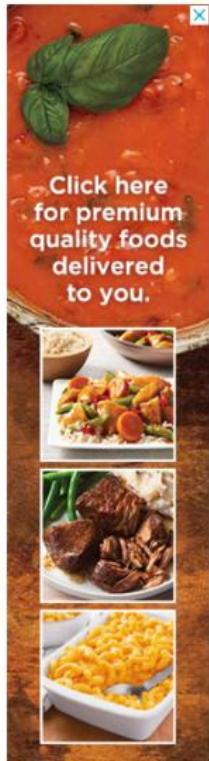


8. Java Programming Interview Question 8

What will be the output when below java program is compiled and executed?

```
public class Test {  
  
    public static void main(String[] args) {  
        try {  
            throw new IOException("Hello");  
        } catch(IOException | Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Java Programming Interview Questions Answers



1. Java Programming Interview Question 1 Answer and Explanation

The given statements output will be "false" because in java + operator precedence is more than == operator. So the given expression will be evaluated to "s1 == s2 is:abc" == "abc" i.e false.

2. Java Programming Interview Question 2 Answer and Explanation

The given statements output will be "ourn". First character will be automatically type caste to int. After that since in java first character index is 0, so it will start from 'o' and print till 'n'. Note that in String `substring` function it leaves the end index.

3. Java Programming Interview Question 3 Answer and Explanation

The size of the shortSet will be 100. Java Autoboxing feature has been introduced in JDK 5, so while adding the short to HashSet<Short> it will automatically convert it to Short object. Now "i-1" will be converted to an int while evaluation and after that it will autoboxed to Integer object but there are no Integer object in the HashSet, so it will not remove anything from the HashSet and finally its size will be 100.

4. Java Programming Interview Question 4 Answer and Explanation

The finally block will never be reached here. If flag will be TRUE, it will go into an infinite loop and if its false its exiting the JVM. So finally block will never be reached here.

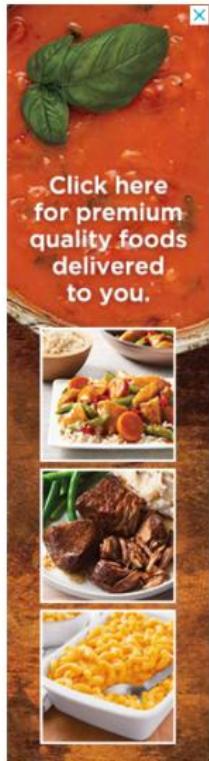
5. Java Programming Interview Question 5 Answer and Explanation

The given print statement will throw `java.lang.NullPointerException` because while evaluating the OR logical operator it will first evaluate both the literals and since str is null, `equals()` method will throw exception. Its always advisable to use short circuit logical operators i.e "||" and "&&" which evaluates the literals values from left and since the first literal will return true, it will skip the second literal evaluation.

6. Java Programming Interview Question 6 Answer and Explanation

The statements will print abc. Notice that `x.concat(y);` will create a new string but it's not assigned to x, so value of x





6. Java Programming Interview Question 6 Answer and Explanation

The statements will print `abc`. Notice that `x.concat(y);` will create a new string but it's not assigned to `x`, so value of `x` is not changed.

7. Java Programming Interview Question 7 Answer and Explanation

This is a tricky question, it looks like the test is about the order of execution of the mathematical operators and syntax of main method will get overlooked. It will produce Runtime error because main method is not static, something like below.

```
pankaj:bin pankaj$ java MathTest  
Error: Main method is not static in class MathTest, please define the main method as:  
public static void main(String[] args)
```

8. Java Programming Interview Question 8 Answer and Explanation

No, it won't print Hello. It will be a compile time error as `The exception IOException is already caught by the alternative Exception.`

I hope that the above scenarios will help a bit in understanding some of the java concepts. Please try these java programming interview questions before going to the solution and comment to let me know your score. 😊

UPDATE: Head over to some more [java coding interview questions](#).

Recently I have created YouTube videos for tricky programs in java, you should check them out. Also subscribe to my [YouTube Channel](#) to get notified when I add new videos.

```
1 Test.java 1/23 Java Tricky Program 1- Java Co... Watch later Share  
File Edit Source Refresh Help  
package com.journaldev.java;  
public class Test {  
    public static void main(String[] args) {  
        System.out.println("Hello");  
    }  
}
```

