Coding Interview Questions:

1. divide a word and print the letters as rows.

**public** **class** PrintRows {

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

String s = "Gayathri";

**for**( **int** i = 0; i <= s.length() - 1; i++)

System.*out*.println(s.charAt(i));

}

}

Output:

G

a

y

a

t

h

r

i

1. comparison between dates and date functions.

**import** java.util.Date;

**public** **class** CompareDates {

**public** **static** **void** main(String[] args) **throws** ParseException {

String s1 = "04/20/2012";

String s2 = "04/12/2016";

SimpleDateFormat format = **new** SimpleDateFormat("dd/mm/yyyy");

Date d1 = format.parse(s1);

Date d2 = format.parse(s2);

**if**(d1.compareTo(d2) == 0)

System.*out*.println("Equal");

**else**

System.*out*.println("unequal");

**if**(d1.before(d2))

System.*out*.println("d1 is before d2");

**else** **if**(d1.after(d2))

System.*out*.println("d1 is after d2");

**else**

System.*out*.println("d1 is equal to d2");

}

}

Output:

unequal

d1 is before d2

Date Functions in Java

* after
* before
* clone
* compareTo(Date)
* compareTo(Obj) -> similar to compareTo(Date) if obj is of class Date else throws classCastException
* equals
* getTime -> returns number of milliseconds
* hashCode
* setTime
* toString

1. What's Spurious wakeups ?

For inexplicable reasons it is possible for threads to wake up even if notify() and notifyAll() has not been called. This is known as spurious wakeups. Wakeups without any reason.

Solution:

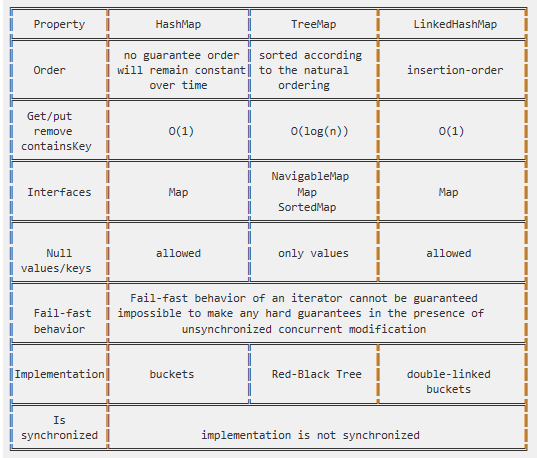
To guard against spurious wakeups the signal member variable is checked inside a while loop instead of inside an if-statement. Such a while loop is also called a spin lock. The thread awakened spins around until the condition in the spin lock (while loop) becomes false.

1. What's Grid computing ?
2. HashMap changes in Java 8 ?
3. Likedhashmap vs Treemap ?

All three classes implement the Map interface and offer mostly the same functionality. The most important difference is the order in which iteration through the entries will happen:

* HashMap makes absolutely no guarantees about the iteration order. It can (and will) even change completely when new elements are added.
* TreeMap will iterate according to the "natural ordering" of the keys according to their compareTo() method (or an externally supplied Comparator). Additionally, it implements the [SortedMap](http://java.sun.com/javase/6/docs/api/java/util/SortedMap.html) interface, which contains methods that depend on this sort order.
* LinkedHashMap will iterate in the order in which the entries were put into the map

["Hashtable"](http://en.wikipedia.org/wiki/Hashtable) is the generic name for hash-based maps. In the context of the Java API, Hashtable is an obsolete class from the days of Java 1.1 before the collections framework existed. It should not be used anymore, because its API is cluttered with obsolete methods that duplicate functionality, and its methods are synchronized (which can decrease performance and is generally useless). Use [ConcurrrentHashMap](http://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ConcurrentHashMap.html) instead of Hashtable.



1. Red black tree implementation
2. Atomicity how it works ?
3. Design pattern observer & it's intents ?
4. Concurrency ?
5. Threadpool executor rejection handler, if queue is full, if Max pool is full what will happen ?
6. what is Class.forname ?
7. Class loaders and principals ?
8. Whar are inner Classes, how to create object of inner classes what is nested inner class ?
9. Default values in…  XML spring , How will you use ternary operator ?
10. How to minimise GC cycles ?
11. Annotation ?
12. 2nd largest in array ?
13. Arraylist and linked list?
14. Array remove middle values and print with null how to remove null in print ?
15. Fibonacci ?
16. Factorial ?
17. Similarity between abstract class and interface ?
18. What is sql injection ?