



## Career Services Assignment 3 – Java Flash Cards

**Instructions:** Research common Java interview questions online and create 20 flash cards from the information you find. Study your flash cards regularly to better prepare for interviews. Fill out the table below with the information you put on each of your flash cards.

Front of Card	Back of Card		
<a href="#">Explain JDK, JRE and JVM?</a>	JDK	JRE	JVM
	It stands for Java Development Kit.	It stands for Java Runtime Environment.	It stands for Java Virtual Machine.
	It is the tool necessary to compile, document and package Java programs.	JRE refers to a runtime environment in which Java bytecode can be executed.	It is an abstract machine. It is a specification that provides a run-time environment in which Java bytecode can be executed.
	It contains JRE + development tools.	It's an implementation of the JVM which physically exists.	JVM follows three notations: Specification, <b>Implementation</b> , and <b>Runtime Instance</b> .
<a href="#">Explain public static void main(String args[]) in Java</a>	<ul style="list-style-type: none"><li>• <b>public:</b> Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.</li><li>• <b>static:</b> It is a keyword in java which identifies it is class-based. main() is made static in Java so that it can be accessed without creating the instance of a Class. In case, main is not made static then the compiler will throw an error as <b>main()</b> is called by the JVM before any objects are made and only static methods can be directly invoked via the class.</li></ul>		



## Career Services Assignment 3 – Java Flash Cards

	<ul style="list-style-type: none"><li>• <b>void:</b> It is the return type of the method. Void defines the method which will not return any value.</li><li>• <b>main:</b> It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occurs.</li><li>• <b>String args[]:</b> It is the parameter passed to the main method.</li></ul>
<a href="#">Why Java is platform independent?</a>	Java is called platform independent because of its byte codes which can run on any system irrespective of its underlying operating system.
<a href="#">Why Java is not 100% Object-oriented?</a>	Java is not 100% Object-oriented because it makes use of eight primitive data types such as boolean, byte, char, int, float, double, long, short which are not objects.
<a href="#">What are wrapper classes in Java?</a>	Wrapper classes convert the Java primitives into the reference types (objects). Every primitive data type has a class dedicated to it. These are known as wrapper classes because they “wrap” the primitive data type into an object of that class.
<a href="#">What are constructors in Java?</a>	<p>In Java, constructor refers to a block of code which is used to initialize an object. It must have the same name as that of the class. Also, it has no return type and it is automatically called when an object is created.</p> <p>There are two types of constructors:</p> <ol style="list-style-type: none"><li>1. <b>Default Constructor:</b> In Java, a default constructor is the one which does not take any inputs. In other words, default constructors are the no argument constructors which will be created by default in case you no other constructor is defined by the user. Its main purpose is to initialize the instance variables with the default values. Also, it is majorly used for object creation.</li></ol>



## Career Services Assignment 3 – Java Flash Cards

	2. <b>Parameterized Constructor:</b> The parameterized constructor in Java, is the constructor which is capable of initializing the instance variables with the provided values. In other words, the constructors which take the arguments are called parameterized constructors.												
<a href="#">What is singleton class in Java and how can we make a class singleton?</a>	Singleton class is a class whose only one instance can be created at any given time, in one JVM. A class can be made singleton by making its constructor private.												
<a href="#">What is the difference between Array list and vector in Java?</a>	<table><thead><tr><th>ArrayList</th><th>Vector</th></tr></thead><tbody><tr><td>Array List is not synchronized.</td><td>Vector is synchronized.</td></tr><tr><td>Array List is fast as it's non-synchronized.</td><td>Vector is slow as it is thread safe.</td></tr><tr><td>If an element is inserted into the Array List, it increases its Array size by 50%.</td><td>Vector defaults to doubling size of its array.</td></tr><tr><td>Array List does not define the increment size.</td><td>Vector defines the increment size.</td></tr><tr><td>Array List can only use Iterator for traversing an Array List.</td><td>Vector can use both Enumeration and Iterator for traversing.</td></tr></tbody></table>	ArrayList	Vector	Array List is not synchronized.	Vector is synchronized.	Array List is fast as it's non-synchronized.	Vector is slow as it is thread safe.	If an element is inserted into the Array List, it increases its Array size by 50%.	Vector defaults to doubling size of its array.	Array List does not define the increment size.	Vector defines the increment size.	Array List can only use Iterator for traversing an Array List.	Vector can use both Enumeration and Iterator for traversing.
ArrayList	Vector												
Array List is not synchronized.	Vector is synchronized.												
Array List is fast as it's non-synchronized.	Vector is slow as it is thread safe.												
If an element is inserted into the Array List, it increases its Array size by 50%.	Vector defaults to doubling size of its array.												
Array List does not define the increment size.	Vector defines the increment size.												
Array List can only use Iterator for traversing an Array List.	Vector can use both Enumeration and Iterator for traversing.												
<a href="#">What is the difference between equals()</a>	<b>Q9. What is the difference between equals() and == in Java?</b>  Equals() method is defined in Object class in Java and used for checking equality of two objects defined by business logic.												



## Career Services Assignment 3 – Java Flash Cards

<a href="#">and == in java?</a>	<p>"==" or equality operator in Java is a binary operator provided by Java programming language and used to compare primitives and objects. <i>public boolean equals(Object o)</i> is the method provided by the Object class. The default implementation uses == operator to compare two objects. For example: method can be overridden like String class. equals() method is used to compare the values of two objects.</p>
---------------------------------	---



## Career Services Assignment 3 – Java Flash Cards

When can you use the super keyword	<p>In Java, the super keyword is a reference variable that refers to an immediate parent class object.</p> <p>When you create a subclass instance, you're also creating an instance of the parent class, which is referenced to by the super reference variable.</p> <p>The uses of the Java super Keyword are-</p> <ol style="list-style-type: none"><li>1. To refer to an immediate parent class instance variable, use super.</li><li>2. The keyword super can be used to call the method of an immediate parent class.</li><li>3. Super() can be used to call the constructor of the immediate parent class.</li></ol>
------------------------------------	--



## Career Services Assignment 3 – Java Flash Cards

What makes a HashSet different from a TreeSet?	HashSet	TreeSet
	It is implemented through a hash table.	TreeSet implements SortedSet Interface that uses trees for storing data.
	It permits the null object.	It does not allow the null object.
	It is faster than TreeSet especially for search, insert, and delete operations.	It is slower than HashSet for these operations.
	It does not maintain elements in an ordered way.	The elements are maintained in a sorted order.
	It uses equals() method to compare two objects.	It uses compareTo() method for comparing two objects.
	It does not permit a heterogenous object.	It permits a heterogenous object.
What are the differences between HashMap and Hashtable in Java	HashMap	Hashtable
	It is non synchronized. It cannot be shared between many threads without proper synchronization code.	It is synchronized. It is thread-safe and can be shared with many threads.
	It permits one null key and multiple null values.	It does not permit any null key or value.
	is a new class introduced in JDK 1.2.	It was present in earlier versions of java as well.
	It is faster.	It is slower.
	It is traversed through the iterator.	It is traversed through Enumerator and Iterator.
	It uses fail fast iterator.	It uses an enumerator which is not fail fast.
	It inherits AbstractMap class.	It inherits Dictionary class.



## Career Services Assignment 3 – Java Flash Cards

What is the importance of reflection in Java	Reflection is a runtime API for inspecting and changing the behavior of methods, classes, and interfaces. Java Reflection is a powerful tool that can be really beneficial. Java Reflection allows you to analyze classes, interfaces, fields, and methods during runtime without knowing what they are called at compile time. Reflection can also be used to create new objects, call methods, and get/set field values. External, user-defined classes can be used by creating instances of extensibility objects with their fully-qualified names. Debuggers can also use reflection to examine private members of classes.
How to not allow serialization of attributes of a class in Java	The Non-Serialized attribute can be used to prevent member variables from being serialized. You should also make an object that potentially contains security-sensitive data non-serializable if possible. Apply the Non-Serialized attribute to certain fields that store sensitive data if the object must be serialized. If you don't exclude these fields from serialization, the data they store will be visible to any programs with serialization permission.
Can you call a constructor of a class inside another constructor	Yes, we can call a constructor of a class inside another constructor. This is also called as constructor chaining. Constructor chaining can be done in 2 ways-  <ol style="list-style-type: none"><li>1. <b>Within the same class:</b> For constructors in the same class, the this() keyword can be used.</li><li>2. <b>From the base class:</b> The super() keyword is used to call the constructor from the base class. The constructor chaining follows the process of inheritance. The constructor of the sub class first calls the constructor of the super class. Due to this, the creation of sub class's object starts with the initialization of the data members of the super class. The constructor chaining works similarly with any number of classes. Every constructor keeps calling the chain till the top of the chain.</li></ol>



## Career Services Assignment 3 – Java Flash Cards

Contiguous memory locations are usually used for storing actual values in an array but not in ArrayList. Explain	An array generally contains elements of the primitive data types such as int, float, etc. In such cases, the array directly stores these elements at contiguous memory locations. While an ArrayList does not contain primitive data types. An ArrayList contains the reference of the objects at different memory locations instead of the object itself. That is why the objects are not stored at contiguous memory locations.
How is the creation of a String using new() different from that of a literal?	When we create a string using new(), a new object is created. Whereas, if we create a string using the string literal syntax, it may return an already existing object with the same name.
Why is synchronization necessary? Explain with the help of a relevant example	<p>Java allows multiple threads to execute. They may be accessing the same variable or object. Synchronization helps to execute threads one after another.</p> <p>It is important as it helps to execute all concurrent threads while being in sync. It prevents memory consistency errors due to access to shared memory. An example of synchronization code is-</p> <pre>1 public synchronized void increment() 2 { 3     a++; 4 }</pre> <p>As we have synchronized this function, this thread can only use the object after the previous thread has used it.</p>





## Career Services Assignment 3 – Java Flash Cards

Explain the term “Double Brace Initialization” in Java	Double Brace Initialization is a Java term that refers to the combination of two independent processes. There are two braces used in this. The first brace creates an anonymous inner class. The second brace is an initialization block. When these both are used together, it is known as Double Brace Initialization. The inner class has a reference to the enclosing outer class, generally using the ‘this’ pointer. It is used to do both creation and initialization in a single statement. It is generally used to initialize collections. It reduces the code and also makes it more readable.
Why is it said that the length() method of String class doesn't return accurate results	<p>The length() method of String class doesn't return accurate results because it simply takes into account the number of characters within in the String. In other words, code points outside of the BMP (Basic Multilingual Plane), that is, code points having a value of U+10000 or above, will be ignored.</p> <p>The reason for this is historical. One of Java's original goals was to consider all text as Unicode; yet, Unicode did not define code points outside of the BMP at the time. It was too late to modify char by the time Unicode specified such code points.</p>

Taken from <https://www.edureka.co/blog/interview-questions/java-interview-questions/>