General Questions:

1. What is polymorphism?
2. What is the difference between function overloading and function overriding?
3. What is encapsulation? What is Inheritance? When will you choose encapsulation/inheritance?
4. What is the difference between a class and a structure?
5. What is the difference between a static method and a non-static method?
6. What is reflection?
7. What is the difference between static binding and dynamic binding?
8. What is a singleton and how can you define a singleton?

Java Questions:

1. What is Java virtual machine and how it is considered in context of Java’s platform independent feature.
2. What does the “transient” modifier do?
3. Is there any case when finally will not be executed?
4. Why string objects are immutable in Java?
5. What is the basic difference between string and stringbuffer object?
6. What does the “volatile” modifier do? What is an AtomicInteger?
7. What is inner class? What is the anonymous inner class? Under which circumstances do we use it?
8. Why the first sentence in sub-class constructor must be a call to super constructor?
9. List up the status of a thread, explain some of them?
10. What is synchronized modifier? What does it put lock on?
11. How HashMap works?
12. What is finalize()?
13. What is java.library.path? What is classpath?
14. How many references types are there in Java? What is the difference between them? List up several use case for each of them?
15. Describe some GC strategies.
16. Talk something about Java concurrent package.
17. Talk something about JVM components, e.g heap, method stack, classloader, etc…
18. Describe the generic type in Java? How JVM handles them?
19. If you are the designer of JDK, what kind of improvement you want to implmenet in the next version of Java?
20. If a Java program got something wrong (e.g very slow in production env) what will you do?

**NTSH 2015 Intern Java Screening Questions**

1. What is Encapsulation?

Encapsulation provides objects with the ability to hide their internal characteristics and behavior. Each object provides a number of methods, which can be accessed by other objects and change its internal data.

1. What is Polymorphism?

Polymorphism is the ability of programming languages to present the same interface for differing underlying data types. A polymorphic type is a type whose operations can also be applied to values of some other type.

1. What **is the Difference between JDK and JRE ?**

The Java Runtime Environment (JRE) is basically the Java Virtual Machine (JVM) where your Java programs are being executed. It also includes browser plugins for applet execution. The Java Development Kit (JDK) is the full featured Software Development Kit for Java, including the JRE, the compilers and tools (like [JavaDoc](http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html), and [Java Debugger](http://docs.oracle.com/javase/7/docs/technotes/tools/windows/jdb.html)), in order for a user to develop, compile and execute Java applications.

1. **What is Autoboxing and Unboxing ?**

Autoboxing is the [automatic conversion made by the Java compiler](http://www.javacodegeeks.com/2013/07/java-generics-tutorial-example-class-interface-methods-wildcards-and-much-more.html) between the primitive types and their corresponding object wrapper classes. For example, the compiler converts an int to an [Integer](http://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html?is-external=true), a double to a [Double](http://docs.oracle.com/javase/7/docs/api/java/lang/Double.html), and so on. If the conversion goes the other way, this operation is called unboxing.

1. **Does Java support multiple inheritance ?**

No, Java does not support multiple inheritance. Each class is able to extend only on one class, but is able to implement more than one interfaces.

1. **What is the difference between processes and threads ?**

A process is an execution of a program, while a [Thread](http://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html) is a single execution sequence within a process. A process can contain multiple threads. A [Thread](http://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html) is sometimes called a lightweight process.

1. **What is the difference between a synchronized method and a synchronized block ?**

A thread can acquire the lock for an object by using the synchronized keyword. The synchronized keyword can be applied in a method level (coarse grained lock) or block level of code (fine grained lock).

1. **What’s a deadlock ?**

A condition that occurs when [two processes are waiting for each other to complete](http://www.javacodegeeks.com/2013/01/java-deadlock-example-how-to-analyze-deadlock-situation.html), before proceeding. The result is that both processes wait endlessly.

1. **What are the basic interfaces of Java Collections Framework ?**

[Java Collections Framework](http://docs.oracle.com/javase/7/docs/technotes/guides/collections/overview.html) provides a well designed set of interfaces and classes that support operations on a collections of objects. The most basic interfaces that reside in the Java Collections Framework are:

* [Collection](http://docs.oracle.com/javase/7/docs/api/java/util/Collection.html), which represents a group of objects known as its elements.
* [Set](http://docs.oracle.com/javase/7/docs/api/java/util/Set.html), which is a collection that cannot contain duplicate elements.
* [List](http://docs.oracle.com/javase/7/docs/api/java/util/List.html), which is an ordered collection and can contain duplicate elements.
* [Map](http://docs.oracle.com/javase/7/docs/api/java/util/Map.html), which is an object that maps keys to values and cannot contain duplicate keys.

1. **Explain the available thread states in a high-level.**

During its execution, a thread can reside in one of the following [states](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.State.html):

* [NEW](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.State.html#NEW): The thread becomes ready to run, but does not necessarily start running immediately.
* [RUNNABLE](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.State.html#RUNNABLE): The Java Virtual Machine (JVM) is actively executing the thread’s code.
* [BLOCKED](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.State.html#BLOCKED): The thread is in a blocked state while waiting for a monitor lock.
* [WAITING](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.State.html#WAITING): The thread waits for another thread to perform a particular action.
* [TIMED\_WAITING](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.State.html#TIMED_WAITING): The thread waits for another thread to perform a particular action up to a specified waiting time.
* [TERMINATED](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.State.html#TERMINATED): The thread has finished its execution.

1. What is difference between fail-fast and fail-safe ?

The [Iterator's](http://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html) fail-safe property works with the clone of the underlying collection and thus, it is not affected by any modification in the collection. All the collection classes in java.util package are fail-fast, while the collection classes in java.util.concurrent are fail-safe. Fail-fast iterators throw a [ConcurrentModificationException](http://examples.javacodegeeks.com/java-basics/exceptions/java-util-concurrentmodificationexception-how-to-handle-concurrent-modification-exception/), while fail-safe iterator never throws such an exception.

1. **When is the finalize() called ? What is the purpose of finalization ?**

The finalize method is called by the garbage collector, just before releasing the object’s memory. It is normally advised to release resources held by the object inside the finalize method.

1. **What are the two types of Exceptions in Java ? Which are the differences between them ?**

Java has two types of exceptions: checked exceptions and unchecked exceptions. Unchecked exceptions do not need to be declared in a method or a constructor’s throws clause, if they can be thrown by the execution of the method or the constructor, and propagate outside the method or constructor boundary. On the other hand, checked exceptions must be declared in a method or a constructor’s throws clause. See here for tips on [Java exception handling](http://www.javacodegeeks.com/2013/07/java-exception-handling-tutorial-with-examples-and-best-practices.html).

1. **When does an Object becomes eligible for Garbage collection in Java ?**

A Java object is subject to garbage collection when it becomes unreachable to the program in which it is currently used.

1. **If an object reference is set to null, will the Garbage Collector immediately free the memory held by that object ?**

No, the object will be available for garbage collection in the next cycle of the garbage collector.

1. **What is the importance of hashCode() and equals() methods ?**

In Java, a [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) uses the [hashCode](http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#hashCode%28%29) and [equals](http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#equals%28java.lang.Object%29) methods to determine the index of the key-value pair and to detect duplicates. More specifically, the [hashCode](http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#hashCode%28%29) method is used in order to determine where the specified key will be stored. Since different keys may produce the same hash value, the [equals](http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#equals%28java.lang.Object%29) method is used, in order to determine whether the specified key actually exists in the collection or not. Therefore, the implementation of both methods is crucial to the accuracy and efficiency of the [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html).

1. **What does the “static” keyword mean ? Can you override private or static method in Java ?**

The static keyword denotes that a member variable or method can be accessed, without requiring an instantiation of the class to which it belongs. A user cannot override static methods in Java, because method overriding is based upon dynamic binding at runtime and static methods are statically binded at compile time. A static method is not associated with any instance of a class so the concept is not applicable.

1. **Explain different ways of creating a thread?**

There are three ways that can be used in order for a [Thread](http://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html) to be created:

* A class may extend the [Thread](http://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html) class.
* A class may implement the [Runnable](http://docs.oracle.com/javase/7/docs/api/java/lang/Runnable.html) interface.
* An application can use the [Executor](http://docs.oracle.com/javase/7/docs/api/java/util/concurrent/Executor.html) framework, in order to create a thread pool.

1. **What differences exist between HashMap and Hashtable ?**

Both the [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) and [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) classes implement the Map interface and thus, have very similar characteristics. However, they differ in the following features:

* A [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) allows the existence of null keys and values, while a [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) doesn’t allow neither null keys, nor null values.
* A [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) is synchronized, while a [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) is not. Thus, [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) is preferred in single-threaded environments, while a [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) is suitable for multi-threaded environments.
* A [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) provides its set of keys and a Java application can iterate over them. Thus, a [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) is fail-fast. On the other hand, a [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) provides an [Enumeration](http://docs.oracle.com/javase/7/docs/api/java/util/Enumeration.html) of its keys.
* The [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) class is considered to be a legacy class.

1. **What is the difference between Exception and Error in java ?**

[Exception](http://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html) and [Error](http://docs.oracle.com/javase/7/docs/api/java/lang/Error.html) classes are both subclasses of the [Throwable](http://docs.oracle.com/javase/7/docs/api/java/lang/Throwable.html) class. The [Exception](http://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html) class is used for exceptional conditions that a user’s program should catch. The [Error](http://docs.oracle.com/javase/7/docs/api/java/lang/Error.html) class defines exceptions that are not excepted to be caught by the user program.