**Q. What is the difference b/w throw and throws?**

Ans.

* We can declare multiple exceptions thrown by a method in throws separating them in comma while we can throw only one instance of exception in throw.
* We cannot use throws inside static and switch statement while throw can use.

Q. **What is the checked and unchecked exception?**

Ans :

* Checked exceptions are the direct subclass of Exception. While unchecked exceptions are the direct subclass of Runtime Exception.
* Checked exception is required handle at compile time while unchecked exception not required.

**Q. Difference between System.exit(-1), System.exit(0) and System.exit(1) in java**

* System.exit(-1)>
* If exit method returns -1 or some other negative number that means some error occurred, The [error](http://www.javamadesoeasy.com/2015/05/javalangerror-in-exception-handling-in.html) occurred which user could NOT have expected. Its Abnormal termination.
* System.exit(0) >
* If exit method returns 0 that means method execution was successful. Its Normal termination.
* System.exit(1) >
* If exit method returns 1 or some other positive number that means some exception occurred, The [exception](http://www.javamadesoeasy.com/2015/05/exception-handling-exception-hierarchy.html) occurred which user could have expected.
* These may be user defined codes to indicate the exception occurred which user could have expected.
* Its Abnormal termination.

**Q. What is the exception?**

Ans : A exception is a unexpected result.

**Q. What is the runtime or unchecked and check exception?**

Ans :

**Unchecked exception:**

* This type of exception occurs due to programming problem.
* These types of exceptions are the direct subclass of runtime Exception.

**Check Exception**

* Checked exception are the exceptions which forces the programmer to catch them explicitly in try-catch block.
* It is a subclass of Exception

**Q. What is the difference b/w exception and error?**

Ans :

* An error is an irrecoverable condition occurring at runtime.
* While exceptions are conditions that occur because of bad input etc.

**Q. What is difference in final, finalize and finally keyword in Java?**  
Ans :

* Final keyword used to create immutable method or class. By making a class final we cannot extend the class, same we cannot override a final method.
* Finalize method called by garbage collector to clean object before it is collected.
* Finally, keyword is used in error or exception handling.

**Q. Finally block is not executed in following scenarios in java ?**

* After System.exit(0);
* for(;;){}



**What is InvalidClassException?**

* No argument constructor
* During searlization and deserialization no serial id found.

**Difference between ClassNotFoundException and NoClassDefFoundError in java ?**

* ClassNotFoundException is Checked (compile time) Exception in java.
* NoClassDefFoundError is a Error in java. Error and its subclasses are regarded as unchecked exceptions in java.
* Here is the hierarchy of java.lang.ClassNotFoundException -
* -java.lang.Object
* java.lang.Throwable
* java.lang.Exception
* java.lang.ReflectiveOperationException
* java.lang.ClassNotFoundException
* Here is the hierarchy of java.lang.NoClassDefFoundError
* -java.lang.Object
* java.lang.Throwable
* java.lang.Error
* java.lang.LinkageError
* java.lang.NoClassDefFoundError
* ClassNotFoundException is thrown when JVM tries to class from classpath but it does not find that class.
* NoClassDefFoundError is thrown when JVM tries to load class which >
* was NOT available at runtime but
* was available at compile time.
* ClassNotFoundException is thrown whenever an java application tries to load a class by passing class name as String in following methods -
* forName(String className) method of java.lang.Class class.
* findSystemClass(String name) method of java.lang.ClassLoader class.
* loadClass(String name) method of java.lang.ClassLoader class.
* These methods will never throw NoClassDefFoundError.