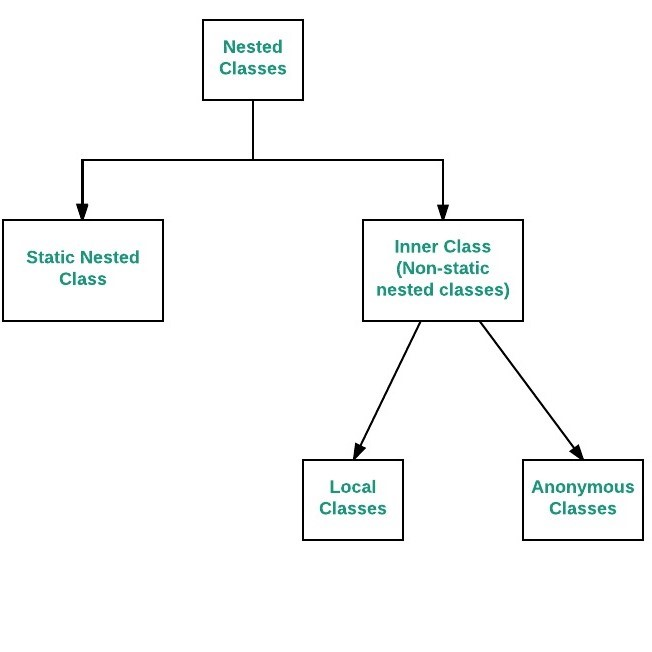
1. Exception Handling

* An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program’s instructions.
* An Error indicates serious problem that a reasonable application should not try to catch *while* Exception indicates conditions that a reasonable application might try to catch.
* Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, FileNotFoundException, NoSuchFieldException, NoSuchMethodException
* The **Throwable** class is the superclass of all errors and exceptions in the **Java** language.

1. **Local class**: Local Inner Classes are the inner classes that are defined inside a block. Generally, this block is a method body  
   **Nested Classes:** In Java, it is possible to define a class within another class, such classes are known as nested classes. They enable you to logically group classes that are only used in one place, thus this increases the use of [encapsulation](https://www.geeksforgeeks.org/encapsulation-in-java/), and creates more readable and maintainable code.  
   types: static nested classes, inner class  
     
     
   It is an inner class without a name and for which only a single object is created. An anonymous inner class can be useful when making an instance of an object with certain “extras” such as overloading methods of a class or interface, without having to actually subclass a class.
2. **Java Reflection** makes it possible to inspect classes, interfaces, fields and methods at runtime, without knowing the names of the classes, methods etc. at compile time. It is also possible to instantiate new objects, invoke methods and get/set field values using **reflection**.  
   **Reflection** is an API which is used to examine or modify the behavior of **methods**, classes, interfaces at runtime. The required classes for **reflection** are provided under **java**. lang. **reflect** package. ... Through **reflection** we can invoke **methods** at runtime irrespective of the access specifier used with them.  
     
     
   Class is a class in java.lang package. Instances of the class represent classes aand interfaces in a running Java application. It has no public constructor. Class objects are constructed automatically by the JVM  
   **Class.forName  
   Myclass.class  
   obj.getClass()**