

LOADING :  
-----

- \* Here we have different class loaders are available to load the required .class file into JVM memory.
- \* In order to load the required .class file, It uses "Delegation Hierarchy Algorithm".
- \* We have following class loaders :

1) Bootstrap OR Primordial class loader :  
-----

It is responsible for loading all the predefined .class files that means all API(Application Programming Interface) level predefined classes are loaded by Bootstrap class loader.

It has the highest priority becuase Bootstrap class loader is the super class for Platform class loader.

It loads the classes from the following path  
C -> Program files -> Java -> JDK -> lib -> jrt-fs.jar

2) Platform OR Extension class loader :  
-----

It is responsible to load the required .class file which is given by some 3rd party in the form of jar file. [JDBC]

It is the sub class of Bootstrap class loader and super class of Application class loader so it has more priority than Application class loader.

It loads the required .class file from the following path.  
C -> Program files -> Java -> JDK -> lib -> ext -> ThirdParty.jar

Command to create the jar file :  
jar cf FileName.jar FileName.class            [\* .class]

[If we want to compile more than one java file at a time then the command is : javac \*.java]

Note : Before java 9, It was known as Extension class loader, but from java 9v It is platform class loader because It uses module system

Application OR System class loader :  
-----

- \* It is responsible to load all the user defined .class file into JVM Memory.
- \* It loads the user defined .class file from **classpath level OR Environemnt variable.**
- \* It has the lowest priority because It is the sub class of Platform class loader.

Note : If all the class loaders are failed to load the .class file into JVM memory then JVM will generate a Runtime Exception i.e **java.lang.ClassNotFoundException**

**How Delegation Hierarchy algorithm works :**  
-----

Whenever JVM makes a request to class loader sub system to load the required .class file into JVM memory, first of all, class loader sub system makes a request to Application class loader, Application class loader will delegate(by pass) the request to the Platform class loader, Platform class loader will also delegate the request to Bootstrap class loader.

Bootstrap class loader will load the .class file from lib folder(jrt-fs.jar) and then by pass the request back to Platform class loader, Platform class loader will load the .class file from ext folder(\*.jar [3rd party jar file]) and by pass the request back to Application class loader, It will load the .class file from environment variable into JVM memory.

Note : java.lang.Object is the first class to be loaded into JVM Memory.