1.What are default class loader used in java

a.Bootstrap

b.Extension

c.System or Application class loader

2.Bootstrap ClassLoader is responsible for loading standard JDK class files from rt.jar and it is parent of all class loaders in Java

3.Bootstrap class loader is also known as Primordial ClassLoader in Java.

4.Extension ClassLoader delegates class loading request to its parent, Bootstrap and if unsuccessful, loads class form jre/lib/ext directory or any other directory pointed by java.ext.dirs system property. Extension ClassLoader in JVM is implemented by sun.misc.Launcher$ExtClassLoader.

5.Third default class loader used by JVM to load Java classes is called System or Application class loader and it is responsible for loading application specific classes from CLASSPATH environment variable, -classpath or -cp command line option, Class-Path attribute of Manifest file inside JAR. Application class loader is a child of Extension ClassLoader and its implemented by sun.misc.Launcher$AppClassLoader class. Also, except Bootstrap class loader, which is implemented in native language mostly in C, all Java class loaders are implemented using java.lang.ClassLoader.

6.Java ClassLoader works in three principles : delegation, visibility and uniqueness

7.How to load class explicitly in Java

Class.forName(classname) and Class.forName(classname, initialized, classloader)

java.lang.ClassLoader : loadClass() -> findClass() -> defineClass() to convert bytecodes into a .class instance which is returned to the caller.

<https://javarevisited.blogspot.com/2012/12/how-classloader-works-in-java.html>

