**Difference between NoClassDefFoundError and ClassNotFoundException?**

**---------------------------------------------------------------------------------------------------**

**Test t1 = new Test();**

**For the our hard coded class name, at run time if the corresponding .class file is not present then JVM will throw RunTime Exception saying NoClassDefFoundError.**

**This is unchecked type of exception.**

**public class Test**

**{**

**public static void main(String[] args)**

**{**

**A a = new A();**

**a.test();**

**System.out.println("done");**

**}**

**}**

**class A{**

**public void test(){**

**System.out.println("Inside the test method!!");**

**}**

**}**

* **Compile it – after compilation we will get two dot class file – A.class and Test.class**
* **Delete the A.class and try to run it – will throw NoClassDefFoundError.**

**ClassNotFoundException**

**---------------------------------**

**Object obj = Class.forName(args[0]).newInstance();**

**At RunTime Dynamically provided class name, if the corresponding dot class in not found or available then Jvm will throw run time exception which saying ClassNotFoundException.**

**public class Test{**

**public static void main(String[] args) {**

**try {**

**Class.forName("GeeksForGeeks");**

**}**

**catch (ClassNotFoundException ex)**

**{**

**ex.printStackTrace();**

**}**

**}**

**}**

**Note: above program will compile successfully and while running it will throw ClassNodFoundException**

**ClassNotFoundException Vs NoClassDefFoundError**

* As the name suggests, ClassNotFoundException is an exception while NoClassDefFoundError is an error.
* ClassNotFoundException occurs when classpath is does not get updated with required JAR files while error occurs when required class definition is not present at runtime.

|  |  |
| --- | --- |
| **ClassNotFoundException** | **NoClassDefFoundError** |
| It is an exception. It is of type java.lang.Exception. | It is an error. It is of type java.lang.Error. |
| It occurs when an application tries to load a class at run time which is not updated in the classpath. | It occurs when java runtime system doesn’t find a class definition, which is present at compile time, but missing at run time. |
| It is thrown by the application itself. It is thrown by the methods like Class.forName(), loadClass() and findSystemClass(). | It is thrown by the Java Runtime System. |
| It occurs when classpath is not updated with required JAR files. | It occurs when required class definition is missing at runtime. |