

Introduction

The **java.lang.Errors** provides for different errors thrown under java lang package.

Error Summary

S.N.	Error & Description
1	AbstractMethodError This is Thrown when an application tries to call an abstract method.
2	AssertionError This is Thrown to indicate that an assertion has failed.
3	ClassCircularityError This is Thrown when a circularity has been detected while initializing a class.
4	ClassFormatError This is Thrown when the Java Virtual Machine attempts to read a class file and determines that the file is malformed or otherwise cannot be interpreted as a class file.
5	Error This is an Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch.
6	ExceptionInInitializerError These are the Signals that an unexpected exception has occurred in a static initializer.
7	IllegalAccessError This is Thrown if an application attempts to access or modify a field, or to call a method that it does not have access to
8	IncompatibleClassChangeError This is Thrown when an incompatible class change has occurred to some class definition.
9	InstantiationError This is Thrown when an application tries to use the Java new construct to instantiate an abstract class or an interface.
10	InternalError This is Thrown to indicate some unexpected internal error has occurred in the Java Virtual Machine.

- 11 **LinkageError**

The Subclasses of LinkageError indicate that a class has some dependency on another class.
- 12 **NoClassDefFoundError**

This is Thrown if the Java Virtual Machine or a ClassLoader instance tries to load in the definition of a class and no definition of the class could be found.
- 13 **NoSuchFieldError**

This is Thrown if an application tries to access or modify a specified field of an object, and that object no longer has that field.
- 14 **NoSuchMethodError**

This is Thrown if an application tries to call a specified method of a class *eitherstaticorinstance*, and that class no longer has a definition of that method.
- 15 **OutOfMemoryError**

This is Thrown when the Java Virtual Machine cannot allocate an object because it is out of memory, and no more memory could be made available by the garbage collector.
- 16 **StackOverflowError**

This is Thrown when a stack overflow occurs because an application recurses too deeply.
- 17 **ThreadDeath**

This is an instance of ThreadDeath is thrown in the victim thread when the stop method with zero arguments in class Thread is called.
- 18 **UnknownError**

This is Thrown when an unknown but serious exception has occurred in the Java Virtual Machine.
- 19 **UnsatisfiedLinkError**

This is Thrown if the Java Virtual Machine cannot find an appropriate native-language definition of a method declared native.
- 20 **UnsupportedClassVersionError**

This is Thrown when the Java Virtual Machine attempts to read a class file and determines that the major and minor version numbers in the file are not supported.
- 21 **VerifyError**

This is Thrown when the "verifier" detects that a class file, though well formed, contains some sort of internal inconsistency or security problem.
- 22 **VirtualMachineError**

This is Thrown to indicate that the Java Virtual Machine is broken or has run out of resources necessary for it to continue operating.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js