public class **AclNotFoundException** extends Exception	This is an exception that is thrown whenever a reference is made to a non-existent ACL (Access Control List)
public class **ActivationException** extends Exception	this exception has been retrofitted to conform to the general purpose exception-chaining mecha The "detail exception" that may be provided at construction time and accessed via the public detail field is now known as the *cause*, and may be accessed via the Throwable.getCause() method, as well as the aforementioned "legacy field."
public class **AlreadyBoundException** extends Exception	An AlreadyBoundException is thrown if an attempt is made to bind an object in the registry to a n that already has an associated binding.
public class **ApplicationException** extends Exception	This class is used for reporting application level exceptions between ORBs and stubs.
public class **AWTException** extends Exception	Signals that an Absract Window Toolkit exception has occurred.
public class **BackingStoreException** extends Exception	Thrown to indicate that a preferences operation could not complete because of a failure in the backing store, or a failure to contact the backing store.
public class **BadAttributeValueExpException** extends Exception	Thrown when an invalid MBean attribute is passed to a query constructing method. This exception is used internally by JMX during the evaluation of a query. User code does not usually see it.
public class **BadBinaryOpValueExpException** extends Exception	Thrown when an invalid expression is passed to a method for constructing a query. This exception is used internally by JMX during the evaluation of a query. User code does not usu
public class **BadLocationException** extends Exception	This exception is to report bad locations within a document model (that is, attempts to reference a location that doesn't exist)
public class **BadStringOperationException**	Thrown when an invalid string operation is passed to a method for constructing a query.

extends Exception	
public class **BrokenBarrierException** extends Exception	Exception thrown when a thread tries to wait upon a barrier that is in a broken state, or which enters the broken state while the thread is waiting.
public class **CertificateException** extends Exception	This exception indicates one of a variety of certificate problems. CertificateEncodingException, CertificateExpiredException, CertificateNotYetValidException, CertificateParsingException
public class **ClassNotFoundException** extends Exception	Thrown when an application tries to load in a class through its string name using: • The forName method in class Class. • The findSystemClass method in class ClassLoader . • The loadClass method in class ClassLoader. but no definition for the class with the specified name could be found. As of release 1.4, this exception has been retrofitted to conform to the general purpose exception-chaining mechanism. The "optional exception that was raised while loading the class" that may be provided at construction time and accessed via the getException() method is now known as the *cause*, and may be accessed via the Throwable.getCause() method, as well as the aforementioned "legacy method."
public class **CloneNotSupportedException** extends Exception	Thrown to indicate that the clone method in class Object has been called to clone an object, but tl t the object's class does not implement the Cloneable interface. Applications that override the clone method can also throw this exception to indicate that an obje could not or should not be cloned.ServerCloneException
public class **DataFormatException** extends Exception	Signals that a data format error has occurred.
public class **DatatypeConfigurationException**	Indicates a serious configuration error.

extends Exception	
public class **DestroyFailedException** extends Exception	Signals that a destroy operation failed. This exception is thrown by credentials implementing the Destroyable interface when the destro method fails.
public class **ExecutionException** extends Exception	Exception thrown when attempting to retrieve the result of a task that aborted by throwing an exception. This exception can be inspected using the Throwable.getCause() method.
public class **ExpandVetoException** extends Exception	Exception used to stop and expand/collapse from happening. See How to Write a Tree-Will-Expand Listener in *The Java Tutorial* for further information and examples.
public class **FontFormatException** extends Exception	Thrown by method createFont in the Font class to indicate that the specified font is bad.
public class **GeneralSecurityException** extends Exception	The GeneralSecurityException class is a generic security exception class that provides type safe for all the security-related exception classes that extend from it. BadPaddingException, CertificateException, CertPathBuilderException, CertPathValidatorExcepti CertStoreException, CRLException, DigestException, ExemptionMechanismException, IllegalBlockSizeException, InvalidAlgorithmParameterException, InvalidKeySpecException, InvalidParameterSpecException, KeyException, KeyStoreException, LoginException, NoSuchAlgorithmException, NoSuchPaddingException, NoSuchProviderException, ShortBufferException, SignatureException, UnrecoverableEntryException
public class **GSSException** extends Exception	This exception is thrown whenever a GSS-API error occurs, including any mechanism specific er It may contain both the major and the minor GSS-API status codes. Major error codes are those defined at the GSS-API level in this class. Minor error codes are mechanism specific error code

	that can provide additional information. The underlying mechanism implementation is responsibl for setting appropriate minor status codes when throwing this exception. Aside from delivering numeric error codes to the caller, this class performs the mapping from their numeric values to textual representations.
public class **IllegalAccessException** extends Exception	An IllegalAccessException is thrown when an application tries to reflectively create an instance (other than an array), set or get a field, or invoke a method, but the currently executing method does not have access to the definition of the specified class, field, method or constructor.
public class **IllegalClassFormatException** extends Exception	Thrown by an implementation of ClassFileTransformer.transform when its input parameters are invalid. This may occur either because the initial class file bytes were invalid or a previously app transform corrupted the bytes.
public class **InstantiationException** extends Exception	Thrown when an application tries to create an instance of a class using the newInstance method class Class, but the specified class object cannot be instantiated. The instantiation can fail for a variety of reasons including but not limited to: 1.the class object represents an abstract class, an interface, an array class, a primitive type, or 2.the class has no nullary constructor

public class **InterruptedException** extends Exception	Thrown when a thread is waiting, sleeping, or otherwise occupied, and the thread is interrupted, either before or during the activity. Occasionally a method may wish to test whether the current has been interrupted, and if so, to immediately throw this exception. The following code can be us
	`if (Thread.interrupted())  // Clears interrupted status!`
	`    throw new InterruptedException();`

public class **InvalidApplicationException** extends Exception	Thrown when an attempt is made to apply either of the following: A subquery expression to an MB qualified attribute expression to an MBean of the wrong class. This exception is used internally b the evaluation of a query. User code does not usually see it.
public class **InvalidMidiDataException** extends Exception	An InvalidMidiDataException indicates that inappropriate MIDI data was encountered. This often means that the data is invalid in and of itself, from the perspective of the MIDI specific An example would be an undefined status byte. However, the exception might simply mean that th in the context it was used, or that the object to which the data was given was unable to parse or For example, a file reader might not be able to parse a Type 2 MIDI file, even though that format
public class **InvalidPreferencesFormatExceptio n** extends Exception	Thrown to indicate that an operation could not complete because the input did not conform to the appropriate XML document type for a collection of preferences, as per the Preferences s
public class **InvalidTargetObjectTypeException** extends Exception	Exception thrown when an invalid target object type is specified. The serialVersionUID of this class is 1190536278266811217L.
public class **InvocationTargetException** extends Exception	InvocationTargetException is a checked exception that wraps an exception thrown by an invoked As of release 1.4, this exception has been retrofitted to conform to the general purpose exceptio . The "target exception" that is provided at construction time and accessed via the**getTargetExce** method is now known as the *cause*, and may be accessed via the **Throwable.getCause()** method, as the aforementioned "legacy method."
public class **IOException** extends Exception	Signals that an I/O exception of some sort has occurred. This class is the general class of excep

	by failed or interrupted I/O operations. ChangedCharSetException, CharacterCodingException, CharConversionException, ClosedChannel EOFException, FileLockInterruptionException, FileNotFoundException, FilerException,HttpRetryExc IIOException, InterruptedIOException, InvalidPropertiesFormatException, JMXProviderException, JMXServerErrorException, MalformedURLException,ObjectStreamException, ProtocolException, SaslException, SocketException, SSLException, SyncFailedException, UnknownHostException, Unk UnsupportedDataTypeException, UnsupportedEncodingException, UTFDataFormatException, ZipEx
public class **JAXBException** extends Exception	This is the root exception class for all JAXB exceptions. MarshalException, PropertyException, UnmarshalException, ValidationException