

CHARACTERISTIC	AWT	SWING	JavaFX
DEPENDENCY	Platform Dependent	Platform Independent	Platform Independent
WEIGHT OF COMPONENT	Heavyweight	Lightweight	Lightweight
CURRENTLY IN USE	Discarded	Use It In A Few Places.	Currently In Use
PLUGGABLE	Does Not Support A Pluggable Look And Feel	Support Pluggable Look And Feel. Components Look The Same On All Systems	Support Pluggable Look And Feel. Components Look The Same On All Systems.
MVC	Does Not Follow MVC	Follow MVC	Follow MVC.
NO. OF COMPONENTS	Less	More Than AWT	More Than AWT But Less Than SWING.
PACKAGE	Java.Awt.Package	Javax.Swing	Javafx.Util
RELEASED	1995	1997	2008

Sr. no.	Key	final	finally	finalize
1.	Definition	final is the keyword and access modifier which is used to apply restrictions on a class, method or variable.	finally is the block in Java Exception Handling to execute the important code whether the exception occurs or not.	finalize is the method in Java which is used to perform clean up processing just before object is garbage collected.
2.	Applicable to	Final keyword is used with the classes, methods and variables.	Finally block is always related to the try and catch block in exception handling.	finalize() method is used with the objects.
3.	Functionality	(1) Once declared, final variable becomes constant and cannot be modified. (2) final method cannot be overridden by sub class. (3) final class cannot be inherited.	(1) finally block runs the important code even if exception occurs or not. (2) finally block cleans up all the resources used in try block	finalize method performs the cleaning activities with respect to the object before its destruction.
4.	Execution	Final method is executed only when we call it.	Finally block is executed as soon as the try-catch block is executed. It's execution is not dependant on the exception.	finalize method is executed just before the object is destroyed.

Checked Exception	Unchecked Exception
Checked exceptions occur at compile time.	Unchecked exceptions occur at runtime.
The compiler checks a checked exception.	The compiler does not check these types of exceptions.
These types of exceptions can be handled at the time of compilation.	These types of exceptions cannot be a catch or handle at the time of compilation, because they get generated by the mistakes in the program.
They are the sub-class of the exception class.	They are runtime exceptions and hence are not a part of the Exception class.
Here, the JVM needs the exception to catch and handle.	Here, the JVM does not require the exception to catch and handle.
Examples of Checked exceptions: <ul style="list-style-type: none"> • File Not Found Exception • No Such Field Exception • Interrupted Exception • No Such Method Exception • Class Not Found Exception 	Examples of Unchecked Exceptions: <ul style="list-style-type: none"> • No Such Element Exception • Undeclared Throwable Exception • Empty Stack Exception • Arithmetic Exception • Null Pointer Exception • Array Index Out of Bounds Exception • Security Exception

Aspect	Character Streams	Byte Streams
Data Handling	Handle character-based data	Handle raw binary data
Representation	Classes end with "Reader" or "Writer"	Classes end with "InputStream" or "OutputStream"
Suitable for	Textual data, strings, human-readable info	Non-textual data, binary files, multimedia
Character Encoding	Automatic encoding and decoding	No encoding or decoding
Text vs non-Text data	Text-based data, strings	Binary data, images, audio, video
Performance	Additional conversion may impact performance	Efficient for handling large binary data
Handle Large Text Files	May impact performance due to encoding	Efficient, no encoding overhead
String Operations	Convenient methods for string operations	Not specifically designed for string operations
Convenience Methods	Higher-level abstractions for text data	Low-level interface for byte data
Reading Line by Line	Convenient methods for reading lines	Byte-oriented, no built-in line-reading methods
File Handling	Read/write text files	Read/write binary files
Network Communication	Sending/receiving text data	Sending/receiving binary data
Handling Images/Audio/Video	Not designed for handling binary data directly	Suitable for handling binary multimedia data
Text Encoding	Supports various character encodings	No specific text encoding support

Sr. no.	Basis of Differences	throw	throws
1.	Definition	Java throw keyword is used to throw an exception explicitly in the code, inside the function or the block of code.	Java throws keyword is used in the method signature to declare an exception which might be thrown by the function while the execution of the code.
2.	Type of exception Using throw keyword, we can only propagate unchecked exception i.e., the checked exception cannot be propagated using throw only.	Using throws keyword, we can declare both checked and unchecked exceptions. However, the throws keyword can be used to propagate checked exceptions only.	
3.	Syntax	The throw keyword is followed by an instance of Exception to be thrown.	The throws keyword is followed by class names of Exceptions to be thrown.
4.	Declaration	throw is used within the method.	throws is used with the method signature.
5.	Internal implementation	We are allowed to throw only one exception at a time i.e. we cannot throw multiple exceptions.	We can declare multiple exceptions using throws keyword that can be thrown by the method. For example, main() throws IOException, SQLException.