CODESECURE

7.

SEI CERT-JAVA RULES AND RECOMMENDATIONS

MAPPED TO CODESONAR® 8.0 WARNING CLASSES

TRUSTED LEADERS OF SOFTWARE ASSURANCE AND ADVANCED CYBER-SECURITY SOLUTIONS

WWW.CODESECURE.COM

INTRODUCTION

The SEI CERT Oracle Coding Standard for Java (CERT-Java) provides rules and recommendations for secure coding in the Java programming language. The goal of these rules and recommendations is to develop safe, reliable, and secure systems, for example by eliminating undefined behaviors that can lead to undefined program behaviors and exploitable vulnerabilities. Conformance to the coding rules defined in this standard is necessary (but not sufficient) to ensure the safety, reliability, and security of software systems developed in the Java programming language.

CodeSonar 8.0 includes a large number of warning classes that support checking for the CERT-Java rules and recommendations. Every CodeSonar warning report includes the identifiers of any CERT-Java rules and recommendations that are closely mapped to the warning's class. (The close mapping for a warning class is the set of categories—including CERT-Java rules and recommendations—that most closely match the class, if any).

You can configure CodeSonar to enable and disable warning classes mapped to specific CERT-Java rules and recommendations, or use build presets to enable all warning classes that are closely mapped to any CERT-Java rules and recommendations. In addition, you can use the CodeSonar search function to find warnings related to specific CERT-Java rules or recommendations, or to any CERT-Java rule or recommendation.

For more information on the SEI CERT-Java Coding Standard:

https://wiki.sei.cmu.edu/confluence/display/java/SEI+CERT+Oracle+Coding+Standard+for+Java

The remainder of this document comprises two tables:

- A table showing the close mapping between CodeSonar warning classes and the SEI CERT Oracle Coding Standard for Java.

- A table showing the broad mapping between CodeSonar warning classes and the SEI CERTOracle Coding Standard for Java. The broad CERT-Java mapping for a CodeSonar warning class includes the close CERT-Java mapping for the class, plus any other CERT-Java rules and recommendations that are related to the class in a meaningful way, but not eligible for theclose mapping.

CodeSecure, Inc. is a leading developer of software-assurance tools and advanced cyber-security solutions. CodeSecure helps organizations develop and release high quality software, free of harmful defects that cause system failures, enable data breaches, and increase corporate liabilities in today's connected world. CodeSecure's CodeSonar is used by embedded developers worldwide.

SEI CERT ORACLE CODING STANDARD FOR JAVA CLOSE MAPPING (CODESONAR V8.0)

The following table contains CodeSonar warning classes that are closely mapped to CERT-Java rules and recommendations.

Rule	Rule Name	Category	Supported
CERT-Java:CON50-J	Do not assume that declaring a reference volatile guarantees safe publication of the members of the referenced object	Recommendation	No
CERT-Java:CON51-J	Do not assume that the sleep(), yield(), or getState() methods provide synchronization semantics	Recommendation	No
CERT-Java:CON52-J	Document thread-safety and use annotations where applicable	Recommendation	No
CERT-Java:DCL00-J	Prevent class initialization cycles	Rule	Yes
CERT-Java:DCL01-J	Do not reuse public identifiers from the Java Standard Library	Rule	No
CERT-Java:DCL02-J	Do not modify the collection's elements during an enhanced for statement	Rule	No
CERT-Java:DCL50-J	Use visually distinct identifiers	Recommendation	No
CERT-Java:DCL51-J	Do not shadow or obscure identifiers in subscopes	Recommendation	No
CERT-Java:DCL52-J	Do not declare more than one variable per declaration	Recommendation	No
CERT-Java:DCL53-J	Minimize the scope of variables	Recommendation	No
CERT-Java:DCL54-J	Use meaningful symbolic constants to represent literal values in program logic	Recommendation	No
CERT-Java:DCL55-J	Properly encode relationships in constant definitions	Recommendation	No
CERT-Java:DCL56-J	Do not attach significance to the ordinal associated with an enum	Recommendation	No
CERT-Java:DCL57-J	Avoid ambiguous overloading of variable arity methods	Recommendation	No
CERT-Java:DCL58-J	Enable compile-time type checking of variable arity parameter types	Recommendation	No
CERT-Java:DCL59-J	Do not apply public final to constants whose value might change in later releases	Recommendation	No
CERT-Java:DCL60-J	Avoid cyclic dependencies between packages	Recommendation	No
CERT-Java:DCL61-J	Do not use raw types	Recommendation	No
CERT-Java:DRD00	Do not store sensitive information on external storage (SD card) unless encrypted first	Rule	Yes
CERT-Java:DRD01-X	Limit the accessibility of an app's sensitive content provider	Rule	No
CERT-Java:DRD02-J	Do not allow WebView to access sensitive local resource through file scheme	Rule	No
CERT-Java:DRD03-J	Do not broadcast sensitive information using an implicit intent	Rule	No
CERT-Java:DRD04-J	Do not log sensitive information	Rule	No
CERT-Java:DRD05-J	Do not grant URI permissions on implicit intents	Rule	No
CERT-Java:DRD06	Do not act on malicious intents	Rule	No
CERT-Java:DRD07-X	Protect exported services with strong permissions	Rule	No
CERT-Java:DRD08-J	Always canonicalize a URL received by a content provider	Rule	No
CERT-Java:DRD09	Restrict access to sensitive activities	Rule	No
CERT-Java:DRD10-X	Do not release apps that are debuggable	Rule	No
CERT-Java:DRD11	Ensure that sensitive data is kept secure	Rule	No
CERT-Java:DRD12	Do not trust data that is world writable	Rule	No
CERT-Java:DRD13	Do not provide addJavascriptInterface method access in a WebView which could contain untrusted content. (API level JELLY_BEAN or below)	Rule	Yes
CERT-Java:DRD14-J	Check that a calling app has appropriate permissions before responding	Rule	No
CERT-Java:DRD15-J	Consider privacy concerns when using Geolocation API	Rule	No
CERT-Java:DRD16-X	Explicitly define the exported attribute for private components	Rule	No
CERT-Java:DRD17-J	Do not use the Android cryptographic security provider encryption default for AES	Rule	Yes

CERT-Java:DRD18	Do not use the default behavior in a cryptographic library if it does not use recommended practices	Rule	Yes
CERT-Java:DRD19	Properly verify server certificate on SSL/TLS	Rule	No
CERT-Java:DRD20-C	Specify permissions when creating files via the NDK	Rule	No
CERT-Java:DRD21-J	Always pass explicit intents to a PendingIntent	Rule	No
CERT-Java:DRD22	Do not cache sensitive information	Rule	Yes
CERT-Java:DRD23	Do not use world readable or writeable to share files between apps	Rule	No
CERT-Java:DRD23-J	Do not use loopback when handling sensitive data	Rule	No
CERT-Java:DRD24	Do not bundle OAuth security-related protocol logic or sensitive data into a relying party's app	Rule	No
CERT-Java:DRD25	To request user permission for OAuth, identify relying party and its permissions scope	Rule	No
CERT-Java:DRD26-J	For OAuth, use a secure Android method to deliver access tokens	Rule	No
CERT-Java:DRD27-J	For OAuth, use an explicit intent method to deliver access tokens	Rule	No
CERT-Java:ENV00-J	Do not sign code that performs only unprivileged operations	Rule	No
CERT-Java:ENV01-J	Place all security-sensitive code in a single JAR and sign and seal it	Rule	Yes
CERT-Java:ENV02-J	Do not trust the values of environment variables	Rule	No
CERT-Java:ENV03-J	Do not grant dangerous combinations of permissions	Rule	Yes
CERT-Java:ENV04-J	Do not disable bytecode verification	Rule	No
CERT-Java:ENV05-J	Do not deploy an application that can be remotely monitored	Rule	No
CERT-Java:ENV06-J	Production code must not contain debugging entry points	Rule	Yes
CERT-Java:ERR00-J	Do not suppress or ignore checked exceptions	Rule	Yes
CERT-Java:ERR01-J	Do not allow exceptions to expose sensitive information	Rule	No
CERT-Java:ERR02-J	Prevent exceptions while logging data	Rule	Yes
CERT-Java:ERR03-J	Restore prior object state on method failure	Rule	No
CERT-Java:ERR04-J	Do not complete abruptly from a finally block	Rule	No
CERT-Java:ERR05-J	Do not let checked exceptions escape from a finally block	Rule	No
CERT-Java:ERR06-J	Do not throw undeclared checked exceptions	Rule	No
CERT-Java:ERR07-J	Do not throw RuntimeException, Exception, or Throwable	Rule	Yes
CERT-Java:ERR08-J	Do not catch NullPointerException or any of its ancestors	Rule	Yes
CERT-Java:ERR09-J	Do not allow untrusted code to terminate the JVM	Rule	Yes
CERT-Java:ERR50-J	Use exceptions only for exceptional conditions	Recommendation	No
CERT-Java:ERR51-J	Prefer user-defined exceptions over more general exception types	Recommendation	No
CERT-Java:ERR52-J	Avoid in-band error indicators	Recommendation	No
CERT-Java:ERR53-J	Try to gracefully recover from system errors	Recommendation	No
CERT-Java:ERR54-J	Use a try-with-resources statement to safely handle closeable resources	Recommendation	No
CERT-Java:EXP00-J	Do not ignore values returned by methods	Rule	Yes
CERT-Java:EXP01-J	Do not use a null in a case where an object is required	Rule	Yes
CERT-Java:EXP02-J	Do not use the Object.equals() method to compare two arrays	Rule	Yes
CERT-Java:EXP03-J	Do not use the equality operators when comparing values of boxed primitives	Rule	Yes
CERT-Java:EXP04-J	Do not pass arguments to certain Java Collections Framework methods that are a different type than the collection parameter type	Rule	No
CERT-Java:EXP05-J	Do not follow a write by a subsequent write or read of the same object within an expression	Rule	No
CERT-Java:EXP06-J	Expressions used in assertions must not produce side effects	Rule	Yes
CERT-Java:EXP07-J	Prevent loss of useful data due to weak references	Rule	No
CERT-Java:EXP50-J	Do not confuse abstract object equality with reference equality	Recommendation	No
CERT-Java:EXP51-J	Do not perform assignments in conditional expressions	Recommendation	No

CERT-Java:EXP52-J	Use braces for the body of an if, for, or while statement	Recommendation	No
CERT-Java:EXP53-J	Use parentheses for precedence of operation	Recommendation	No
CERT-Java:EXP54-J	Understand the differences between bitwise and logical operators	Recommendation	No
CERT-Java:EXP55-J	Use the same type for the second and third operands in conditional expressions	Recommendation	No
CERT-Java:FIO00-J	Do not operate on files in shared directories	Rule	No
CERT-Java:FIO01-J	Create files with appropriate access permissions	Rule	Yes
CERT-Java:FIO02-J	Detect and handle file-related errors	Rule	Yes
CERT-Java:FIO03-J	Remove temporary files before termination	Rule	No
CERT-Java:FIO04-J	Release resources when they are no longer needed	Rule	Yes
CERT-Java:FIO05-J	Do not expose buffers or their backing arrays methods to untrusted code	Rule	No
CERT-Java:FIO06-J	Do not create multiple buffered wrappers on a single byte or character stream	Rule	No
CERT-Java:FIO07-J	Do not let external processes block on IO buffers	Rule	No
CERT-Java:FIO08-J	Distinguish between characters or bytes read from a stream and -1	Rule	No
CERT-Java:FIO09-J	Do not rely on the write() method to output integers outside the range 0 to 255	Rule	Yes
CERT-Java:FIO10-J	Ensure the array is filled when using read() to fill an array	Rule	No
CERT-Java:FIO11-J	Do not convert between strings and bytes without specifying a valid character encoding	Rule	No
CERT-Java:FIO12-J	Provide methods to read and write little-endian data	Rule	No
CERT-Java:FIO13-J	Do not log sensitive information outside a trust boundary	Rule	No
CERT-Java:FIO14-J	Perform proper cleanup at program termination	Rule	No
CERT-Java:FIO15-J	Do not reset a servlet's output stream after committing it	Rule	No
CERT-Java:FIO16-J	Canonicalize path names before validating them	Rule	No
CERT-Java:FIO50-J	Do not make assumptions about file creation	Recommendation	No
CERT-Java:FIO51-J	Identify files using multiple file attributes	Recommendation	No
CERT-Java:FIO52-J	Do not store unencrypted sensitive information on the client side	Recommendation	No
CERT-Java:FIO53-J	Use the serialization methods writeUnshared() and readUnshared() with care	Recommendation	No
CERT-Java:IDS00-J	Prevent SQL injection	Rule	Yes
CERT-Java:IDS01-J	Normalize strings before validating them	Rule	No
CERT-Java:IDS02-J	Canonicalize path names before validating them	Rule	No
CERT-Java:IDS03-J	Do not log unsanitized user input	Rule	Yes
CERT-Java:IDS04-J	Safely extract files from ZipInputStream	Rule	No
CERT-Java:IDS05-J	Use a safe subset of ASCII for file and path names	Rule	No
CERT-Java:IDS06-J	Exclude unsanitized user input from format strings	Rule	No
CERT-Java:IDS07-J	Sanitize untrusted data passed to the Runtime.exec() method	Rule	Yes
CERT-Java:IDS08-J	Sanitize untrusted data included in a regular expression	Rule	Yes
CERT-Java:IDS09-J	Specify an appropriate locale when comparing locale-dependent data	Rule	No
CERT-Java:IDS10-J	Don't form strings containing partial characters	Rule	No
CERT-Java:IDS11-J	Perform any string modifications before validation	Rule	No
CERT-Java:IDS13-J	Use compatible character encodings on both sides of file or network IO	Rule	No
CERT-Java:IDS14-J	Do not trust the contents of hidden form fields	Rule	Yes
CERT-Java:IDS15-J	Do not allow sensitive information to leak outside a trust boundary	Rule	No
CERT-Java:IDS16-J	Prevent XML Injection	Rule	No
CERT-Java:IDS17-J	Prevent XML External Entity Attacks	Rule	No
CERT-Java:IDS50-J	Use conservative file naming conventions	Recommendation	No
CERT-Java:IDS51-J	Properly encode or escape output	Recommendation	No
CERT-Java:IDS52-J	Prevent code injection	Recommendation	No

CERT-Java:IDS53-J	Prevent XPath Injection	Recommendation	No
CERT-Java:IDS54-J	Prevent LDAP injection	Recommendation	No
CERT-Java:IDS55-J	Understand how escape characters are interpreted when strings are loaded	Recommendation	No
CERT-Java:IDS56-J	Prevent arbitrary file upload	Recommendation	No
CERT-Java:JNI00-J	Define wrappers around native methods	Rule	No
CERT-Java:JNI01-J	Safely invoke standard APIs that perform tasks using the immediate caller's class loader instance (loadLibrary)	Rule	No
CERT-Java:JNI02-J	Do not assume object references are constant or unique	Rule	No
CERT-Java:JNI03-J	Do not use direct pointers to Java objects in JNI code	Rule	No
CERT-Java:JNI04-J	Do not assume that Java strings are null-terminated	Rule	No
CERT-Java:LCK00-J	Use private final lock objects to synchronize classes that may interact with untrusted code	Rule	Yes
CERT-Java:LCK01-J	Do not synchronize on objects that may be reused	Rule	No
CERT-Java:LCK02-J	Do not synchronize on the class object returned by getClass()	Rule	No
CERT-Java:LCK03-J	Do not synchronize on the intrinsic locks of high-level concurrency objects	Rule	No
CERT-Java:LCK04-J	Do not synchronize on a collection view if the backing collection is accessible	Rule	No
CERT-Java:LCK05-J	Synchronize access to static fields that can be modified by untrusted code	Rule	Yes
CERT-Java:LCK06-J	Do not use an instance lock to protect shared static data	Rule	No
CERT-Java:LCK07-J	Avoid deadlock by requesting and releasing locks in the same order	Rule	No
CERT-Java:LCK08-J	Ensure actively held locks are released on exceptional conditions	Rule	No
CERT-Java:LCK09-J	Do not perform operations that can block while holding a lock	Rule	Yes
CERT-Java:LCK10-J	Use a correct form of the double-checked locking idiom	Rule	Yes
CERT-Java:LCK11-J	Avoid client-side locking when using classes that do not commit to their locking strategy	Rule	No
CERT-Java:MET00-J	Validate method arguments	Rule	No
CERT-Java:MET01-J	Never use assertions to validate method arguments	Rule	No
CERT-Java:MET02-J	Do not use deprecated or obsolete classes or methods	Rule	No
CERT-Java:MET03-J	Methods that perform a security check must be declared private or final	Rule	No
CERT-Java:MET04-J	Do not increase the accessibility of overridden or hidden methods	Rule	No
CERT-Java:MET05-J	Ensure that constructors do not call overridable methods	Rule	No
CERT-Java:MET06-J	Do not invoke overridable methods in clone()	Rule	No
CERT-Java:MET07-J	Never declare a class method that hides a method declared in a superclass or superinterface	Rule	No
CERT-Java:MET08-J	Preserve the equality contract when overriding the equals() method	Rule	Yes
CERT-Java:MET09-J	Classes that define an equals() method must also define a hashCode() method	Rule	Yes
CERT-Java:MET10-J	Follow the general contract when implementing the compareTo() method	Rule	Yes
CERT-Java:MET11-J	Ensure that keys used in comparison operations are immutable	Rule	No
CERT-Java:MET12-J	Do not use finalizers	Rule	No
CERT-Java:MET13-J	Do not assume that reassigning method arguments modifies the calling environment	Rule	No
CERT-Java:MET50-J	Avoid ambiguous or confusing uses of overloading	Recommendation	No
CERT-Java:MET51-J	Do not use overloaded methods to differentiate between runtime types	Recommendation	No
CERT-Java:MET52-J	Do not use the clone() method to copy untrusted method parameters	Recommendation	No
CERT-Java:MET53-J	Ensure that the clone() method calls super.clone()	Recommendation	No
CERT-Java:MET54-J	Always provide feedback about the resulting value of a method	Recommendation	No
CERT-Java:MET55-J	Return an empty array or collection instead of a null value for methods that return an array or collection	Recommendation	No
CERT-Java:MET56-J	Do not use Object.equals() to compare cryptographic keys	Recommendation	No

CERT-Java:MSC00-J	Use SSLSocket rather than Socket for secure data exchange	Rule	No
CERT-Java:MSC01-J	Do not use an empty infinite loop	Rule	No
CERT-Java:MSC02-J	Generate strong random numbers	Rule	Yes
CERT-Java:MSC03-J	Never hard code sensitive information	Rule	Yes
CERT-Java:MSC04-J	Do not leak memory	Rule	No
CERT-Java:MSC05-J	Do not exhaust heap space	Rule	Yes
CERT-Java:MSC06-J	Do not modify the underlying collection when an iteration is in progress	Rule	No
CERT-Java:MSC07-J	Prevent multiple instantiations of singleton objects	Rule	No
CERT-Java:MSC08-J	Do not store nonserializable objects as attributes in an HTTP session	Rule	No
CERT-Java:MSC09-J	For OAuth, ensure (a) [relying party receiving user's ID in last step] is same as (b) [relying party the access token was granted to].	Rule	No
CERT-Java:MSC10-J	Do not use OAuth 2.0 implicit grant (unmodified) for authentication	Rule	No
CERT-Java:MSC11-J	Do not let session information leak within a servlet	Rule	No
CERT-Java:MSC50-J	Minimize the scope of the @SuppressWarnings annotation	Recommendation	No
CERT-Java:MSC51-J	Do not place a semicolon immediately following an if, for, or while condition	Recommendation	No
CERT-Java:MSC52-J	Finish every set of statements associated with a case label with a break statement	Recommendation	No
CERT-Java:MSC53-J	Carefully design interfaces before releasing them	Recommendation	No
CERT-Java:MSC54-J	Avoid inadvertent wrapping of loop counters	Recommendation	No
CERT-Java:MSC55-J	Use comments consistently and in a readable fashion	Recommendation	No
CERT-Java:MSC56-J	Detect and remove superfluous code and values	Recommendation	No
CERT-Java:MSC57-J	Strive for logical completeness	Recommendation	No
CERT-Java:MSC58-J	Prefer using iterators over enumerations	Recommendation	No
CERT-Java:MSC59-J	Limit the lifetime of sensitive data	Recommendation	No
CERT-Java:MSC60-J	Do not use assertions to verify the absence of runtime errors	Recommendation	No
CERT-Java:MSC61-J	Do not use insecure or weak cryptographic algorithms	Recommendation	No
CERT-Java:MSC62-J	Store passwords using a hash function	Recommendation	No
CERT-Java:MSC63-J	Ensure that SecureRandom is properly seeded	Recommendation	No
CERT-Java:NUM00-J	Detect or prevent integer overflow	Rule	Yes
CERT-Java:NUM01-J	Do not perform bitwise and arithmetic operations on the same data	Rule	No
CERT-Java:NUM02-J	Ensure that division and remainder operations do not result in divide-by-zero errors	Rule	No
CERT-Java:NUM03-J	Use integer types that can fully represent the possible range of unsigned data	Rule	No
CERT-Java:NUM04-J	Do not use floating-point numbers if precise computation is required	Rule	No
CERT-Java:NUM07-J	Do not attempt comparisons with NaN	Rule	No
CERT-Java:NUM08-J	Check floating-point inputs for exceptional values	Rule	No
CERT-Java:NUM09-J	Do not use floating-point variables as loop counters	Rule	No
CERT-Java:NUM10-J	Do not construct BigDecimal objects from floating-point literals	Rule	No
CERT-Java:NUM11-J	Do not compare or inspect the string representation of floating-point values	Rule	No
CERT-Java:NUM12-J	Ensure conversions of numeric types to narrower types do not result in lost or misinterpreted data	Rule	Yes
CERT-Java:NUM13-J	Avoid loss of precision when converting primitive integers to floating-point	Rule	Yes
CERT-Java:NUM14-J	Use shift operators correctly	Rule	No
CERT-Java:NUM50-J	Convert integers to floating point for floating-point operations	Recommendation	No
CERT-Java:NUM51-J	Do not assume that the remainder operator always returns a nonnegative result for integral operands	Recommendation	No
CERT-Java:NUM52-J	Be aware of numeric promotion behavior	Recommendation	No
CERT-Java:NUM53-J	Use the strictfp modifier for floating-point calculation consistency across platforms	Recommendation	No

CERT-Java:NUM54-J	Do not use denormalized numbers	Recommendation	No
CERT-Java:OBJ01-J	Limit accessibility of fields	Rule	No
CERT-Java:OBJ02-J	Preserve dependencies in subclasses when changing superclasses	Rule	No
CERT-Java:OBJ03-J	Prevent heap pollution	Rule	No
CERT-Java:OBJ04-J	Provide mutable classes with copy functionality to safely allow passing instances to untrusted code	Rule	No
CERT-Java:OBJ05-J	Do not return references to private mutable class members	Rule	No
CERT-Java:OBJ06-J	Defensively copy mutable inputs and mutable internal components	Rule	No
CERT-Java:OBJ07-J	Sensitive classes must not let themselves be copied	Rule	Yes
CERT-Java:OBJ08-J	Do not expose private members of an outer class from within a nested class	Rule	Yes
CERT-Java:OBJ09-J	Compare classes and not class names	Rule	No
CERT-Java:OBJ10-J	Do not use public static nonfinal fields	Rule	No
CERT-Java:OBJ11-J	Be wary of letting constructors throw exceptions	Rule	No
CERT-Java:OBJ12-J	Respect object-based annotations	Rule	No
CERT-Java:OBJ13-J	Ensure that references to mutable objects are not exposed	Rule	No
CERT-Java:OBJ14-J	Do not use an object that has been freed.	Rule	No
CERT-Java:OBJ50-J	Never confuse the immutability of a reference with that of the referenced object	Recommendation	No
CERT-Java:OBJ51-J	Minimize the accessibility of classes and their members	Recommendation	No
CERT-Java:OBJ52-J	Write garbage-collection-friendly code	Recommendation	No
CERT-Java:OBJ53-J	Do not use direct buffers for short-lived, infrequently used objects	Recommendation	No
CERT-Java:OBJ54-J	Do not attempt to help the garbage collector by setting local reference variables to null	Recommendation	No
CERT-Java:OBJ55-J	Remove short-lived objects from long-lived container objects	Recommendation	No
CERT-Java:OBJ56-J	Provide sensitive mutable classes with unmodifiable wrappers	Recommendation	No
CERT-Java:OBJ57-J	Do not rely on methods that can be overridden by untrusted code	Recommendation	No
CERT-Java:OBJ58-J	Limit the extensibility of classes and methods with invariants	Recommendation	No
CERT-Java:SEC00-J	Do not allow privileged blocks to leak sensitive information across a trust boundary	Rule	No
CERT-Java:SEC01-J	Do not allow tainted variables in privileged blocks	Rule	Yes
CERT-Java:SEC02-J	Do not base security checks on untrusted sources	Rule	No
CERT-Java:SEC03-J	Do not load trusted classes after allowing untrusted code to load arbitrary classes	Rule	No
CERT-Java:SEC04-J	Protect sensitive operations with security manager checks	Rule	No
CERT-Java:SEC05-J	Do not use reflection to increase accessibility of classes, methods, or fields	Rule	Yes
CERT-Java:SEC06-J	Do not rely on the default automatic signature verification provided by URLClassLoader and java.util.jar	Rule	Yes
CERT-Java:SEC07-J	Call the superclass's getPermissions() method when writing a custom class loader	Rule	No
CERT-Java:SEC08-J	Trusted code must discard or clean any arguments provided by untrusted code	Rule	No
CERT-Java:SEC09-J	Never leak the results of certain standard API methods from trusted code to untrusted code	Rule	No
CERT-Java:SEC10-J	Never permit untrusted code to invoke any API that may (possibly transitively) invoke the reflection APIs	Rule	No
CERT-Java:SEC50-J	Avoid granting excess privileges	Recommendation	No
CERT-Java:SEC51-J	Minimize privileged code	Recommendation	No
CERT-Java:SEC52-J	Do not expose methods that use reduced-security checks to untrusted code	Recommendation	No
CERT-Java:SEC53-J	Define custom security permissions for fine-grained security	Recommendation	No
CERT-Java:SEC54-J	Create a secure sandbox using a security manager	Recommendation	No
CERT-Java:SEC55-J	Ensure that security-sensitive methods are called with validated arguments	Recommendation	No
CERT-Java:SEC56-J	Do not serialize direct handles to system resources	Recommendation	No

CERT-Java:SEC57-J	Do not let untrusted code misuse privileges of callback methods	Recommendation	No
CERT-Java:SEC58-J	Deserialization methods should not perform potentially dangerous operations	Recommendation	No
CERT-Java:SER00-J	Enable serialization compatibility during class evolution	Rule	Yes
CERT-Java:SER01-J	Do not deviate from the proper signatures of serialization methods	Rule	Yes
CERT-Java:SER02-J	Sign then seal objects before sending them outside a trust boundary	Rule	Yes
CERT-Java:SER03-J	Do not serialize unencrypted sensitive data	Rule	Yes
CERT-Java:SER04-J	Do not allow serialization and deserialization to bypass the security manager	Rule	No
CERT-Java:SER05-J	Do not serialize instances of inner classes	Rule	No
CERT-Java:SER06-J	Make defensive copies of private mutable components during deserialization	Rule	Yes
CERT-Java:SER07-J	Do not use the default serialized form for classes with implementation-defined invariants	Rule	Yes
CERT-Java:SER08-J	Minimize privileges before deserializing from a privileged context	Rule	No
CERT-Java:SER09-J	Do not invoke overridable methods from the readObject() method	Rule	No
CERT-Java:SER10-J	Avoid memory and resource leaks during serialization	Rule	Yes
CERT-Java:SER11-J	Prevent overwriting of externalizable objects	Rule	No
CERT-Java:SER12-J	Prevent deserialization of untrusted data	Rule	Yes
CERT-Java:SER13-J	Deserialization methods should not perform potentially dangerous operations	Rule	No
CERT-Java:STR00-J	Don't form strings containing partial characters from variable-width encodings	Rule	No
CERT-Java:STR01-J	Do not assume that a Java char fully represents a Unicode code point	Rule	No
CERT-Java:STR02-J	Specify an appropriate locale when comparing locale-dependent data	Rule	No
CERT-Java:STR03-J	Do not encode noncharacter data as a string	Rule	No
CERT-Java:STR04-J	Use compatible character encodings when communicating string data between JVMs	Rule	No
CERT-Java:STR50-J	Use the appropriate method for counting characters in a string	Recommendation	No
CERT-Java:STR51-J	Use the charset encoder and decoder classes when more control over the encoding process is required	Recommendation	No
CERT-Java:THI00-J	Do not invoke Thread.run()	Rule	Yes
CERT-Java:THI01-J	Do not invoke ThreadGroup methods	Rule	No
CERT-Java:THI02-J	Notify all waiting threads rather than a single thread	Rule	No
CERT-Java:THI03-J	Always invoke wait() and await() methods inside a loop	Rule	No
CERT-Java:THI04-J	Ensure that threads performing blocking operations can be terminated	Rule	No
CERT-Java:THI05-J	Do not use Thread.stop() to terminate threads	Rule	No
CERT-Java:TPS00-J	Use thread pools to enable graceful degradation of service during traffic bursts	Rule	No
CERT-Java:TPS01-J	Do not execute interdependent tasks in a bounded thread pool	Rule	No
CERT-Java:TPS02-J	Ensure that tasks submitted to a thread pool are interruptible	Rule	No
CERT-Java:TPS03-J	Ensure that tasks executing in a thread pool do not fail silently	Rule	No
CERT-Java:TPS04-J	Ensure ThreadLocal variables are reinitialized when using thread pools	Rule	No
CERT-Java:TSM00-J	Do not override thread-safe methods with methods that are not thread-safe	Rule	No
CERT-Java:TSM01-J	Do not let the this reference escape during object construction	Rule	No
CERT-Java:TSM02-J	Do not use background threads during class initialization	Rule	No
CERT-Java:TSM03-J	Do not publish partially initialized objects	Rule	No
CERT-Java:VNA00-J	Ensure visibility when accessing shared primitive variables	Rule	Yes
CERT-Java:VNA01-J	Ensure visibility of shared references to immutable objects	Rule	No
CERT-Java:VNA02-J	Ensure that compound operations on shared variables are atomic	Rule	No
CERT-Java:VNA03-J	Do not assume that a group of calls to independently atomic methods is atomic	Rule	Yes
CERT-Java:VNA04-J	Ensure that calls to chained methods are atomic	Rule	No
CERT-Java:VNA05-J	Ensure atomicity when reading and writing 64-bit values	Rule	No

SEI CERT ORACLE CODING STANDARD FOR JAVA BROAD MAPPING (CODESONAR V8.0)

The following table contains CodeSonar warning classes that are broadly mapped to CERT-Java rules and recommendations.

Rule	Rule Name	Category	Supported
CERT-Java:CON50-J	Do not assume that declaring a reference volatile guarantees safe publication of the members of the referenced object	Recommendation	No
CERT-Java:CON51-J	Do not assume that the sleep(), yield(), or getState() methods provide synchronization semantics	Recommendation	No
CERT-Java:CON52-J	Document thread-safety and use annotations where applicable	Recommendation	No
CERT-Java:DCL00-J	Prevent class initialization cycles	Rule	Yes
CERT-Java:DCL01-J	Do not reuse public identifiers from the Java Standard Library	Rule	No
CERT-Java:DCL02-J	Do not modify the collection's elements during an enhanced for statement	Rule	No
CERT-Java:DCL50-J	Use visually distinct identifiers	Recommendation	No
CERT-Java:DCL51-J	Do not shadow or obscure identifiers in subscopes	Recommendation	No
CERT-Java:DCL52-J	Do not declare more than one variable per declaration	Recommendation	No
CERT-Java:DCL53-J	Minimize the scope of variables	Recommendation	No
CERT-Java:DCL54-J	Use meaningful symbolic constants to represent literal values in program logic	Recommendation	No
CERT-Java:DCL55-J	Properly encode relationships in constant definitions	Recommendation	No
CERT-Java:DCL56-J	Do not attach significance to the ordinal associated with an enum	Recommendation	No
CERT-Java:DCL57-J	Avoid ambiguous overloading of variable arity methods	Recommendation	No
CERT-Java:DCL58-J	Enable compile-time type checking of variable arity parameter types	Recommendation	No
CERT-Java:DCL59-J	Do not apply public final to constants whose value might change in later releases	Recommendation	No
CERT-Java:DCL60-J	Avoid cyclic dependencies between packages	Recommendation	No
CERT-Java:DCL61-J	Do not use raw types	Recommendation	No
CERT-Java:DRD00	Do not store sensitive information on external storage (SD card) unless encrypted first	Rule	Yes
CERT-Java:DRD01-X	Limit the accessibility of an app's sensitive content provider	Rule	No
CERT-Java:DRD02-J	Do not allow WebView to access sensitive local resource through file scheme	Rule	No
CERT-Java:DRD03-J	Do not broadcast sensitive information using an implicit intent	Rule	No
CERT-Java:DRD04-J	Do not log sensitive information	Rule	No
CERT-Java:DRD05-J	Do not grant URI permissions on implicit intents	Rule	No
CERT-Java:DRD06	Do not act on malicious intents	Rule	No
CERT-Java:DRD07-X	Protect exported services with strong permissions	Rule	No
CERT-Java:DRD08-J	Always canonicalize a URL received by a content provider	Rule	No
CERT-Java:DRD09	Restrict access to sensitive activities	Rule	No
CERT-Java:DRD10-X	Do not release apps that are debuggable	Rule	No
CERT-Java:DRD11	Ensure that sensitive data is kept secure	Rule	No
CERT-Java:DRD12	Do not trust data that is world writable	Rule	No
CERT-Java:DRD13	Do not provide addJavascriptInterface method access in a WebView which could contain untrusted content. (API level JELLY_BEAN or below)	Rule	Yes
CERT-Java:DRD14-J	Check that a calling app has appropriate permissions before responding	Rule	No
CERT-Java:DRD15-J	Consider privacy concerns when using Geolocation API	Rule	No
CERT-Java:DRD16-X	Explicitly define the exported attribute for private components	Rule	No
CERT-Java:DRD17-J	Do not use the Android cryptographic security provider encryption default for AES	Rule	Yes

CERT-Java:DRD18	Do not use the default behavior in a cryptographic library if it does not use recommended practices	Rule	Yes
CERT-Java:DRD19	Properly verify server certificate on SSL/TLS	Rule	No
CERT-Java:DRD20-C	Specify permissions when creating files via the NDK	Rule	No
CERT-Java:DRD21-J	Always pass explicit intents to a PendingIntent	Rule	No
CERT-Java:DRD22	Do not cache sensitive information	Rule	Yes
CERT-Java:DRD23	Do not use world readable or writeable to share files between apps	Rule	No
CERT-Java:DRD23-J	Do not use loopback when handling sensitive data	Rule	No
CERT-Java:DRD24	Do not bundle OAuth security-related protocol logic or sensitive data into a relying party's app	Rule	No
CERT-Java:DRD25	To request user permission for OAuth, identify relying party and its permissions scope	Rule	No
CERT-Java:DRD26-J	For OAuth, use a secure Android method to deliver access tokens	Rule	No
CERT-Java:DRD27-J	For OAuth, use an explicit intent method to deliver access tokens	Rule	No
CERT-Java:ENV00-J	Do not sign code that performs only unprivileged operations	Rule	No
CERT-Java:ENV01-J	Place all security-sensitive code in a single JAR and sign and seal it	Rule	Yes
CERT-Java:ENV02-J	Do not trust the values of environment variables	Rule	No
CERT-Java:ENV03-J	Do not grant dangerous combinations of permissions	Rule	Yes
CERT-Java:ENV04-J	Do not disable bytecode verification	Rule	No
CERT-Java:ENV05-J	Do not deploy an application that can be remotely monitored	Rule	No
CERT-Java:ENV06-J	Production code must not contain debugging entry points	Rule	Yes
CERT-Java:ERR00-J	Do not suppress or ignore checked exceptions	Rule	Yes
CERT-Java:ERR01-J	Do not allow exceptions to expose sensitive information	Rule	No
CERT-Java:ERR02-J	Prevent exceptions while logging data	Rule	Yes
CERT-Java:ERR03-J	Restore prior object state on method failure	Rule	No
CERT-Java:ERR04-J	Do not complete abruptly from a finally block	Rule	No
CERT-Java:ERR05-J	Do not let checked exceptions escape from a finally block	Rule	No
CERT-Java:ERR06-J	Do not throw undeclared checked exceptions	Rule	No
CERT-Java:ERR07-J	Do not throw RuntimeException, Exception, or Throwable	Rule	Yes
CERT-Java:ERR08-J	Do not catch NullPointerException or any of its ancestors	Rule	Yes
CERT-Java:ERR09-J	Do not allow untrusted code to terminate the JVM	Rule	Yes
CERT-Java:ERR50-J	Use exceptions only for exceptional conditions	Recommendation	No
CERT-Java:ERR51-J	Prefer user-defined exceptions over more general exception types	Recommendation	No
CERT-Java:ERR52-J	Avoid in-band error indicators	Recommendation	No
CERT-Java:ERR53-J	Try to gracefully recover from system errors	Recommendation	No
CERT-Java:ERR54-J	Use a try-with-resources statement to safely handle closeable resources	Recommendation	No
CERT-Java:EXP00-J	Do not ignore values returned by methods	Rule	Yes
CERT-Java:EXP01-J	Do not use a null in a case where an object is required	Rule	Yes
CERT-Java:EXP02-J	Do not use the Object.equals() method to compare two arrays	Rule	Yes
CERT-Java:EXP03-J	Do not use the equality operators when comparing values of boxed primitives	Rule	Yes
CERT-Java:EXP04-J	Do not pass arguments to certain Java Collections Framework methods that are a different type than the collection parameter type	Rule	No
CERT-Java:EXP05-J	Do not follow a write by a subsequent write or read of the same object within an expression	Rule	No
CERT-Java:EXP06-J	Expressions used in assertions must not produce side effects	Rule	Yes
CERT-Java:EXP07-J	Prevent loss of useful data due to weak references	Rule	No
CERT-Java:EXP50-J	Do not confuse abstract object equality with reference equality	Recommendation	No
CERT-Java:EXP51-J	Do not perform assignments in conditional expressions	Recommendation	No

CERT-Java:EXP52-J	Use braces for the body of an if, for, or while statement	Recommendation	No
CERT-Java:EXP53-J	Use parentheses for precedence of operation	Recommendation	No
CERT-Java:EXP54-J	Understand the differences between bitwise and logical operators	Recommendation	No
CERT-Java:EXP55-J	Use the same type for the second and third operands in conditional expressions	Recommendation	No
CERT-Java:FIO00-J	Do not operate on files in shared directories	Rule	No
CERT-Java:FIO01-J	Create files with appropriate access permissions	Rule	Yes
CERT-Java:FIO02-J	Detect and handle file-related errors	Rule	Yes
CERT-Java:FIO03-J	Remove temporary files before termination	Rule	No
CERT-Java:FIO04-J	Release resources when they are no longer needed	Rule	Yes
CERT-Java:FIO05-J	Do not expose buffers or their backing arrays methods to untrusted code	Rule	No
CERT-Java:FIO06-J	Do not create multiple buffered wrappers on a single byte or character stream	Rule	No
CERT-Java:FIO07-J	Do not let external processes block on IO buffers	Rule	No
CERT-Java:FIO08-J	Distinguish between characters or bytes read from a stream and -1	Rule	No
CERT-Java:FIO09-J	Do not rely on the write() method to output integers outside the range 0 to 255	Rule	Yes
CERT-Java:FIO10-J	Ensure the array is filled when using read() to fill an array	Rule	No
CERT-Java:FIO11-J	Do not convert between strings and bytes without specifying a valid character encoding	Rule	No
CERT-Java:FIO12-J	Provide methods to read and write little-endian data	Rule	No
CERT-Java:FIO13-J	Do not log sensitive information outside a trust boundary	Rule	No
CERT-Java:FIO14-J	Perform proper cleanup at program termination	Rule	No
CERT-Java:FIO15-J	Do not reset a servlet's output stream after committing it	Rule	No
CERT-Java:FIO16-J	Canonicalize path names before validating them	Rule	No
CERT-Java:FIO50-J	Do not make assumptions about file creation	Recommendation	No
CERT-Java:FIO51-J	Identify files using multiple file attributes	Recommendation	No
CERT-Java:FIO52-J	Do not store unencrypted sensitive information on the client side	Recommendation	No
CERT-Java:FIO53-J	Use the serialization methods writeUnshared() and readUnshared() with care	Recommendation	No
CERT-Java:IDS00-J	Prevent SQL injection	Rule	Yes
CERT-Java:IDS01-J	Normalize strings before validating them	Rule	No
CERT-Java:IDS02-J	Canonicalize path names before validating them	Rule	No
CERT-Java:IDS03-J	Do not log unsanitized user input	Rule	Yes
CERT-Java:IDS04-J	Safely extract files from ZipInputStream	Rule	No
CERT-Java:IDS05-J	Use a safe subset of ASCII for file and path names	Rule	No
CERT-Java:IDS06-J	Exclude unsanitized user input from format strings	Rule	No
CERT-Java:IDS07-J	Sanitize untrusted data passed to the Runtime.exec() method	Rule	Yes
CERT-Java:IDS08-J	Sanitize untrusted data included in a regular expression	Rule	Yes
CERT-Java:IDS09-J	Specify an appropriate locale when comparing locale-dependent data	Rule	No
CERT-Java:IDS10-J	Don't form strings containing partial characters	Rule	No
CERT-Java:IDS11-J	Perform any string modifications before validation	Rule	No
CERT-Java:IDS13-J	Use compatible character encodings on both sides of file or network IO	Rule	No
CERT-Java:IDS14-J	Do not trust the contents of hidden form fields	Rule	Yes
CERT-Java:IDS15-J	Do not allow sensitive information to leak outside a trust boundary	Rule	No
CERT-Java:IDS16-J	Prevent XML Injection	Rule	No
CERT-Java:IDS17-J	Prevent XML External Entity Attacks	Rule	No
CERT-Java:IDS50-J	Use conservative file naming conventions	Recommendation	No
CERT-Java:IDS51-J	Properly encode or escape output	Recommendation	No
CERT-Java:IDS52-J	Prevent code injection	Recommendation	No

CERT-Java:IDS53-J	Prevent XPath Injection	Recommendation	No
CERT-Java:IDS54-J	Prevent LDAP injection	Recommendation	No
CERT-Java:IDS55-J	Understand how escape characters are interpreted when strings are loaded	Recommendation	No
CERT-Java:IDS56-J	Prevent arbitrary file upload	Recommendation	No
CERT-Java:JNI00-J	Define wrappers around native methods	Rule	No
CERT-Java:JNI01-J	Safely invoke standard APIs that perform tasks using the immediate caller's class loader instance (loadLibrary)	Rule	No
CERT-Java:JNI02-J	Do not assume object references are constant or unique	Rule	No
CERT-Java:JNI03-J	Do not use direct pointers to Java objects in JNI code	Rule	No
CERT-Java:JNI04-J	Do not assume that Java strings are null-terminated	Rule	No
CERT-Java:LCK00-J	Use private final lock objects to synchronize classes that may interact with untrusted code	Rule	Yes
CERT-Java:LCK01-J	Do not synchronize on objects that may be reused	Rule	No
CERT-Java:LCK02-J	Do not synchronize on the class object returned by getClass()	Rule	No
CERT-Java:LCK03-J	Do not synchronize on the intrinsic locks of high-level concurrency objects	Rule	No
CERT-Java:LCK04-J	Do not synchronize on a collection view if the backing collection is accessible	Rule	No
CERT-Java:LCK05-J	Synchronize access to static fields that can be modified by untrusted code	Rule	Yes
CERT-Java:LCK06-J	Do not use an instance lock to protect shared static data	Rule	No
CERT-Java:LCK07-J	Avoid deadlock by requesting and releasing locks in the same order	Rule	No
CERT-Java:LCK08-J	Ensure actively held locks are released on exceptional conditions	Rule	No
CERT-Java:LCK09-J	Do not perform operations that can block while holding a lock	Rule	Yes
CERT-Java:LCK10-J	Use a correct form of the double-checked locking idiom	Rule	Yes
CERT-Java:LCK11-J	Avoid client-side locking when using classes that do not commit to their locking strategy	Rule	No
CERT-Java:MET00-J	Validate method arguments	Rule	No
CERT-Java:MET01-J	Never use assertions to validate method arguments	Rule	No
CERT-Java:MET02-J	Do not use deprecated or obsolete classes or methods	Rule	No
CERT-Java:MET03-J	Methods that perform a security check must be declared private or final	Rule	No
CERT-Java:MET04-J	Do not increase the accessibility of overridden or hidden methods	Rule	No
CERT-Java:MET05-J	Ensure that constructors do not call overridable methods	Rule	No
CERT-Java:MET06-J	Do not invoke overridable methods in clone()	Rule	No
CERT-Java:MET07-J	Never declare a class method that hides a method declared in a superclass or superinterface	Rule	No
CERT-Java:MET08-J	Preserve the equality contract when overriding the equals() method	Rule	Yes
CERT-Java:MET09-J	Classes that define an equals() method must also define a hashCode() method	Rule	Yes
CERT-Java:MET10-J	Follow the general contract when implementing the compareTo() method	Rule	Yes
CERT-Java:MET11-J	Ensure that keys used in comparison operations are immutable	Rule	No
CERT-Java:MET12-J	Do not use finalizers	Rule	No
CERT-Java:MET13-J	Do not assume that reassigning method arguments modifies the calling environment	Rule	No
CERT-Java:MET50-J	Avoid ambiguous or confusing uses of overloading	Recommendation	No
CERT-Java:MET51-J	Do not use overloaded methods to differentiate between runtime types	Recommendation	No
CERT-Java:MET52-J	Do not use the clone() method to copy untrusted method parameters	Recommendation	No
CERT-Java:MET53-J	Ensure that the clone() method calls super.clone()	Recommendation	No
CERT-Java:MET54-J	Always provide feedback about the resulting value of a method	Recommendation	No
CERT-Java:MET55-J	Return an empty array or collection instead of a null value for methods that return an array or collection	Recommendation	No
CERT-Java:MET56-J	Do not use Object.equals() to compare cryptographic keys	Recommendation	No

CERT-Java:MSC00-J	Use SSLSocket rather than Socket for secure data exchange	Rule	No
CERT-Java:MSC01-J	Do not use an empty infinite loop	Rule	No
CERT-Java:MSC02-J	Generate strong random numbers	Rule	Yes
CERT-Java:MSC03-J	Never hard code sensitive information	Rule	Yes
CERT-Java:MSC04-J	Do not leak memory	Rule	No
CERT-Java:MSC05-J	Do not exhaust heap space	Rule	Yes
CERT-Java:MSC06-J	Do not modify the underlying collection when an iteration is in progress	Rule	No
CERT-Java:MSC07-J	Prevent multiple instantiations of singleton objects	Rule	No
CERT-Java:MSC08-J	Do not store nonserializable objects as attributes in an HTTP session	Rule	No
CERT-Java:MSC09-J	For OAuth, ensure (a) [relying party receiving user's ID in last step] is same as (b) [relying party the access token was granted to].	Rule	No
CERT-Java:MSC10-J	Do not use OAuth 2.0 implicit grant (unmodified) for authentication	Rule	No
CERT-Java:MSC11-J	Do not let session information leak within a servlet	Rule	No
CERT-Java:MSC50-J	Minimize the scope of the @SuppressWarnings annotation	Recommendation	No
CERT-Java:MSC51-J	Do not place a semicolon immediately following an if, for, or while condition	Recommendation	No
CERT-Java:MSC52-J	Finish every set of statements associated with a case label with a break statement	Recommendation	No
CERT-Java:MSC53-J	Carefully design interfaces before releasing them	Recommendation	No
CERT-Java:MSC54-J	Avoid inadvertent wrapping of loop counters	Recommendation	No
CERT-Java:MSC55-J	Use comments consistently and in a readable fashion	Recommendation	No
CERT-Java:MSC56-J	Detect and remove superfluous code and values	Recommendation	No
CERT-Java:MSC57-J	Strive for logical completeness	Recommendation	No
CERT-Java:MSC58-J	Prefer using iterators over enumerations	Recommendation	No
CERT-Java:MSC59-J	Limit the lifetime of sensitive data	Recommendation	No
CERT-Java:MSC60-J	Do not use assertions to verify the absence of runtime errors	Recommendation	No
CERT-Java:MSC61-J	Do not use insecure or weak cryptographic algorithms	Recommendation	No
CERT-Java:MSC62-J	Store passwords using a hash function	Recommendation	No
CERT-Java:MSC63-J	Ensure that SecureRandom is properly seeded	Recommendation	No
CERT-Java:NUM00-J	Detect or prevent integer overflow	Rule	Yes
CERT-Java:NUM01-J	Do not perform bitwise and arithmetic operations on the same data	Rule	No
CERT-Java:NUM02-J	Ensure that division and remainder operations do not result in divide-by-zero errors	Rule	No
CERT-Java:NUM03-J	Use integer types that can fully represent the possible range of unsigned data	Rule	No
CERT-Java:NUM04-J	Do not use floating-point numbers if precise computation is required	Rule	No
CERT-Java:NUM07-J	Do not attempt comparisons with NaN	Rule	No
CERT-Java:NUM08-J	Check floating-point inputs for exceptional values	Rule	No
CERT-Java:NUM09-J	Do not use floating-point variables as loop counters	Rule	No
CERT-Java:NUM10-J	Do not construct BigDecimal objects from floating-point literals	Rule	No
CERT-Java:NUM11-J	Do not compare or inspect the string representation of floating-point values	Rule	No
CERT-Java:NUM12-J	Ensure conversions of numeric types to narrower types do not result in lost or misinterpreted data	Rule	Yes
CERT-Java:NUM13-J	Avoid loss of precision when converting primitive integers to floating-point	Rule	Yes
CERT-Java:NUM14-J	Use shift operators correctly	Rule	No
CERT-Java:NUM50-J	Convert integers to floating point for floating-point operations	Recommendation	No
CERT-Java:NUM51-J	Do not assume that the remainder operator always returns a nonnegative result for integral operands	Recommendation	No
CERT-Java:NUM52-J	Be aware of numeric promotion behavior	Recommendation	No

CERT-Java:NUM53-J	Use the strictfp modifier for floating-point calculation consistency across platforms	Recommendation	No
CERT-Java:NUM54-J	Do not use denormalized numbers	Recommendation	No
CERT-Java:OBJ01-J	Limit accessibility of fields	Rule	No
CERT-Java:OBJ02-J	Preserve dependencies in subclasses when changing superclasses	Rule	No
CERT-Java:OBJ03-J	Prevent heap pollution	Rule	No
CERT-Java:OBJ04-J	Provide mutable classes with copy functionality to safely allow passing instances to untrusted code	Rule	No
CERT-Java:OBJ05-J	Do not return references to private mutable class members	Rule	No
CERT-Java:OBJ06-J	Defensively copy mutable inputs and mutable internal components	Rule	No
CERT-Java:OBJ07-J	Sensitive classes must not let themselves be copied	Rule	Yes
CERT-Java:OBJ08-J	Do not expose private members of an outer class from within a nested class	Rule	Yes
CERT-Java:OBJ09-J	Compare classes and not class names	Rule	No
CERT-Java:OBJ10-J	Do not use public static nonfinal fields	Rule	No
CERT-Java:OBJ11-J	Be wary of letting constructors throw exceptions	Rule	No
CERT-Java:OBJ12-J	Respect object-based annotations	Rule	No
CERT-Java:OBJ13-J	Ensure that references to mutable objects are not exposed	Rule	No
CERT-Java:OBJ14-J	Do not use an object that has been freed.	Rule	No
CERT-Java:OBJ50-J	Never confuse the immutability of a reference with that of the referenced object	Recommendation	No
CERT-Java:OBJ51-J	Minimize the accessibility of classes and their members	Recommendation	No
CERT-Java:OBJ52-J	Write garbage-collection-friendly code	Recommendation	No
CERT-Java:OBJ53-J	Do not use direct buffers for short-lived, infrequently used objects	Recommendation	No
CERT-Java:OBJ54-J	Do not attempt to help the garbage collector by setting local reference variables to null	Recommendation	No
CERT-Java:OBJ55-J	Remove short-lived objects from long-lived container objects	Recommendation	No
CERT-Java:OBJ56-J	Provide sensitive mutable classes with unmodifiable wrappers	Recommendation	No
CERT-Java:OBJ57-J	Do not rely on methods that can be overridden by untrusted code	Recommendation	No
CERT-Java:OBJ58-J	Limit the extensibility of classes and methods with invariants	Recommendation	No
CERT-Java:SEC00-J	Do not allow privileged blocks to leak sensitive information across a trust boundary	Rule	No
CERT-Java:SEC01-J	Do not allow tainted variables in privileged blocks	Rule	Yes
CERT-Java:SEC02-J	Do not base security checks on untrusted sources	Rule	No
CERT-Java:SEC03-J	Do not load trusted classes after allowing untrusted code to load arbitrary classes	Rule	No
CERT-Java:SEC04-J	Protect sensitive operations with security manager checks	Rule	No
CERT-Java:SEC05-J	Do not use reflection to increase accessibility of classes, methods, or fields	Rule	Yes
CERT-Java:SEC06-J	Do not rely on the default automatic signature verification provided by URLClassLoader and java.util.jar	Rule	Yes
CERT-Java:SEC07-J	Call the superclass's getPermissions() method when writing a custom class loader	Rule	No
CERT-Java:SEC08-J	Trusted code must discard or clean any arguments provided by untrusted code	Rule	No
CERT-Java:SEC09-J	Never leak the results of certain standard API methods from trusted code to untrusted code	Rule	No
CERT-Java:SEC10-J	Never permit untrusted code to invoke any API that may (possibly transitively) invoke the reflection APIs	Rule	No
CERT-Java:SEC50-J	Avoid granting excess privileges	Recommendation	No
CERT-Java:SEC51-J	Minimize privileged code	Recommendation	No
CERT-Java:SEC52-J	Do not expose methods that use reduced-security checks to untrusted code	Recommendation	No

CERT-Java:SEC53-J	Define custom security permissions for fine-grained security	Recommendation	No
CERT-Java:SEC54-J	Create a secure sandbox using a security manager	Recommendation	No
CERT-Java:SEC55-J	Ensure that security-sensitive methods are called with validated arguments	Recommendation	No
CERT-Java:SEC56-J	Do not serialize direct handles to system resources	Recommendation	No
CERT-Java:SEC57-J	Do not let untrusted code misuse privileges of callback methods	Recommendation	No
CERT-Java:SEC58-J	Deserialization methods should not perform potentially dangerous operations	Recommendation	No
CERT-Java:SER00-J	Enable serialization compatibility during class evolution	Rule	Yes
CERT-Java:SER01-J	Do not deviate from the proper signatures of serialization methods	Rule	Yes
CERT-Java:SER02-J	Sign then seal objects before sending them outside a trust boundary	Rule	Yes
CERT-Java:SER03-J	Do not serialize unencrypted sensitive data	Rule	Yes
CERT-Java:SER04-J	Do not allow serialization and deserialization to bypass the security manager	Rule	No
CERT-Java:SER05-J	Do not serialize instances of inner classes	Rule	No
CERT-Java:SER06-J	Make defensive copies of private mutable components during deserialization	Rule	Yes
CERT-Java:SER07-J	Do not use the default serialized form for classes with implementation-defined invariants	Rule	Yes
CERT-Java:SER08-J	Minimize privileges before deserializing from a privileged context	Rule	No
CERT-Java:SER09-J	Do not invoke overridable methods from the readObject() method	Rule	No
CERT-Java:SER10-J	Avoid memory and resource leaks during serialization	Rule	Yes
CERT-Java:SER11-J	Prevent overwriting of externalizable objects	Rule	No
CERT-Java:SER12-J	Prevent deserialization of untrusted data	Rule	Yes
CERT-Java:SER13-J	Deserialization methods should not perform potentially dangerous operations	Rule	No
CERT-Java:STR00-J	Don't form strings containing partial characters from variable-width encodings	Rule	No
CERT-Java:STR01-J	Do not assume that a Java char fully represents a Unicode code point	Rule	No
CERT-Java:STR02-J	Specify an appropriate locale when comparing locale-dependent data	Rule	No
CERT-Java:STR03-J	Do not encode noncharacter data as a string	Rule	No
CERT-Java:STR04-J	Use compatible character encodings when communicating string data between JVMs	Rule	No
CERT-Java:STR50-J	Use the appropriate method for counting characters in a string	Recommendation	No
CERT-Java:STR51-J	Use the charset encoder and decoder classes when more control over the encoding process is required	Recommendation	No
CERT-Java:THI00-J	Do not invoke Thread.run()	Rule	Yes
CERT-Java:THI01-J	Do not invoke ThreadGroup methods	Rule	No
CERT-Java:THI02-J	Notify all waiting threads rather than a single thread	Rule	No
CERT-Java:THI03-J	Always invoke wait() and await() methods inside a loop	Rule	No
CERT-Java:THI04-J	Ensure that threads performing blocking operations can be terminated	Rule	No
CERT-Java:THI05-J	Do not use Thread.stop() to terminate threads	Rule	No
CERT-Java:TPS00-J	Use thread pools to enable graceful degradation of service during traffic bursts	Rule	No
CERT-Java:TPS01-J	Do not execute interdependent tasks in a bounded thread pool	Rule	No
CERT-Java:TPS02-J	Ensure that tasks submitted to a thread pool are interruptible	Rule	No
CERT-Java:TPS03-J	Ensure that tasks executing in a thread pool do not fail silently	Rule	No
CERT-Java:TPS04-J	Ensure ThreadLocal variables are reinitialized when using thread pools	Rule	No
CERT-Java:TSM00-J	Do not override thread-safe methods with methods that are not thread-safe	Rule	No
CERT-Java:TSM01-J	Do not let the this reference escape during object construction	Rule	No
CERT-Java:TSM02-J	Do not use background threads during class initialization	Rule	No
CERT-Java:TSM03-J	Do not publish partially initialized objects	Rule	No
CERT-Java:VNA00-J	Ensure visibility when accessing shared primitive variables	Rule	Yes
CERT-Java:VNA01-J	Ensure visibility of shared references to immutable objects	Rule	No

CERT-Java:VNA02-J	Ensure that compound operations on shared variables are atomic	Rule	No
CERT-Java:VNA03-J	Do not assume that a group of calls to independently atomic methods is atomic	Rule	Yes
CERT-Java:VNA04-J	Ensure that calls to chained methods are atomic	Rule	No
CERT-Java:VNA05-J	Ensure atomicity when reading and writing 64-bit values	Rule	No