

# Oracle: The Java® Language Specification SE 8 Edition

[return to Oracle de facto Standards](#)

Table 1: Data sheet for The Java® Language Specification

Title	The Java® Language Specification
Acronym	Java
Version	SE 8 Edition
Document Number	
Release Date	13 February 2015
Reference	<a href="https://docs.oracle.com/javase/specs/jls/se8/html/">https://docs.oracle.com/javase/specs/jls/se8/html/</a>

**Note:** The following is an excerpt from the official site. It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

## Introduction

*The Java® [programming language](#) is a general-purpose, concurrent, [class](#)-based, object-oriented language. It is designed to be simple enough that many programmers can achieve fluency in the language. The Java programming language is related to C and C++ but is organized rather differently, with a number of aspects of C and C++ omitted and a few ideas from other languages included. It is intended to be a production language, not a research language, and so, as C. A. R. Hoare suggested in his classic paper on language design, the design has avoided including new and untested features.*

*The Java programming language is strongly and statically typed. This specification clearly distinguishes between the compile-time errors that can and must be detected at compile time, and those that occur at run time. Compile time normally consists of translating programs into a machine-independent byte code representation. Run-time activities include loading and linking of the classes needed to execute a program, optional machine code generation and dynamic optimization of the program, and actual program execution.*

*The Java programming language is a relatively high-level language, in that details of the machine representation are not available through the language. It includes automatic storage management, typically using a garbage collector, to avoid the safety problems of explicit deallocation (as in C's `free` or C++'s `delete`). High-[performance](#) garbage-collected implementations can have bounded pauses to support systems programming and real-time applications. The language does not include any unsafe constructs, such as array accesses without index checking, since such unsafe constructs would cause a program to behave in an unspecified way.*

*The Java programming language is normally compiled to the bytecode instruction set and binary format defined in The Java [Virtual Machine](#) Specification, Java SE 8 Edition.*

Last update: 2021/11/09 15:40 dido:public:ra:xapend:xapend.b\_stds:defact:orcle:java [https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b\\_stds:defact:orcle:java](https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:orcle:java)

---

From: <https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link: [https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b\\_stds:defact:orcle:java](https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:orcle:java)

Last update: **2021/11/09 15:40**

