

Java Primitive and Reference Types

Table of contents

Two categories of types	1
Primitive types	1
Reference types	2

Introduced in [2026-02-26-course-overview-and-java-basics¹](#) (Lecture 1, Week 1).

Two categories of types

Java distinguishes two sorts of types:

1. **Primitive types** — built-in, fixed representations.
2. **Reference types** — everything else (i.e. classes), including `String`, arrays, and any user-defined class.

Primitive types

Type	Size	Stores
<code>boolean</code>	not specified	<code>true</code> or <code>false</code>
<code>byte</code>	1 byte / 8 bits	whole numbers from -128 to 127
<code>short</code>	2 bytes / 16 bits	whole numbers from -32,768 to 32,767
<code>int</code>	4 bytes / 32 bits	whole numbers from -2,147,483,648 to 2,147,483,647

¹[courses/csse2002/2026-02-26-course-overview-and-java-basics.pdf](#)

Type	Size	Stores
<code>long</code>	8 bytes / 64 bits	whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
<code>float</code>	4 bytes / 32 bits	fractional numbers, accurate to 6-7 decimal places
<code>double</code>	8 bytes / 64 bits	fractional numbers, accurate to 15 decimal places
<code>char</code>	2 bytes / 16 bits	a single character/letter

Reference types

Reference types are everything that isn't primitive — classes, including `String` and arrays. A variable of a reference type stores a reference to an object, not the object's data directly (see [2026-02-26-java-basics-part-02-collections-and-strings²](#) for the stack/heap consequences of this, and how it changes the meaning of `=`, `==`, and `!=`).

Collections (`Stack`, `List`, `Set`, `Map`) can only store objects/reference types — see [java-collections-framework³](#) for the wrapper classes (`Boolean`, `Byte`, `Character`, `Double`, `Float`, `Integer`, `Long`, `Short`) Java provides so primitives can be stored in them.

²[courses/csse2002/2026-02-26-java-basics-part-02-collections-and-strings.pdf](#)

³[courses/csse2002/java-collections-framework.pdf](#)