Data Types & Type Casting in Java

**Introduction**

A data type is a classification of data. It tells the compiler or interpreter how the programmer aims to use the data. Data types are a crucial factor in all computer [programming](https://www.naukri.com/learning/what-is-programming-st619) languages. While developing computer software, a programmer ensures that data types are assigned correctly to get an error-free and workable program. Data types represent the type, nature, and set of operations for the value which they stores.

**What are the Data types in Languages?**

**Data types** are different sizes and values that can be stored in the variable that is made as per convenience and circumstances to cover up all test cases. Also, let us cover up other important ailments that there are majorly two types of languages that are as follows:

1. **Statically typed language:** where each variable and expression type is already known at compile time. Once a variable is declared to be of a certain data type, it cannot hold values of other data types. Mostly these can be compiler based Languages and Programmer **must specify** it like C, C++, Java, Scala and C#.
2. **Dynamically typed language:** These languages can receive different data types over time. Mostly these can be interpreter based Languages and Programmer **not need** not specify type like in the case of BASIC, Ruby, Python Erlang, Perl, VB, and PHP.

**What are the Types in Java?**

The Java programming language is **statically-typed**, which means that all ***variables*** must first be ***declared*** before they can be used. This involves stating the variable's type and name, like

float height=5.10F;

ofcourse Java programming language is also **strongly typed language**, which means that every ***variable*** and every ***expression*** has a type that is known at compile time.

**Type**s

(i) limit the values that a ***variable*** can hold or that an ***expression***can produce,

(ii) limit the ***operations*** supported on those values, and

(iii) determine the ***meaning*** of the operations.

NOTE: Strong typing helps detect errors at ***compile time***.

## What are the Kinds of Types and Values?

There are **two** kinds of Types in the Java programming language

(I) primitive types and (II) reference types.

(I) The **8** primitive types are categorized as

(A) **numeric types** and the (B) **boolean type**

The (A) **numeric types** are again categorized as

(i) **integral types** and (ii) **floating-point types**

**integral types** are again into (a) **integer** and (b) **character Types.**

(a) **integer types** : byte, short, int, long

(b) **character type :**  char,

(ii) **floating-point types** : float and double.

The (B) **boolean type** : boolean

(II)The **3** reference types are only…

class types,interface types, and array types (as per James Gosling)

NOTE: There is also a special **null type in Java**.

## 

## I) Primitive Types

## Primitive data types in Java are built-in data types that are predefined, always have a value. There are eight built-in primitive data types in Java like int, float etc.

## Example: int age=49;

## 

## Primitive data types in java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primitive**  **Type** | **Size bytes bits** | | **min and max values(inclusive)** | **Range** |
| byte | 1 | 8 | –128 to 127 | -2**7** **to** 2**7** – 1 |
| short | 2 | 16 | –32768 to 32767 | -2**15** **to** 2**15** – 1 |
| int | 4 | 32 | –2147483648 to 2147483647 | -2**31** **to** 2**31**– 1 |
| long | 8 | 64 | –9223372036854775808 to 9223372036854775807 | -2**63** **to** 2**63**– 1 |
| char | 2 | 16 | ('\u0000') 0 to 65535 ('\uffff') | -2**15** **to** 2**15** – 1 |
| float | 4 | 32 | **1.40129846432481707e-45 to 3.40282346638528860e+38** | **3.4e-038 to 3.4e+038** |
| double | 8 | 64 | **4.94065645841246544e-324d to 1.79769313486231570e+308d** | **1.7e-308 to 1.7e+308** |
| boolean | - | 1 | *true* and *false* (only) |  |

**Points to Remember:**

**1.** Size of the boolean type is dependent upon the Java Virtual Machine. Therefore, it

fluctuates on different platforms. Most of the Operating Systems uses 1 bit to store

false(0) or true(1) .So internal representation in windows JVM can be 1 bit.

**2.** Java supports Unicode characters and provides provision to multiple languages like

English, French, German, etc. So char type enhanced to 2 bytes to occupy16 bits.

**3.** All numeric values in java are signed only (no unsigned like C/C++).

**4.** ‘L’ or ‘l’ must be appended to long type literal.

**5.** ‘F’ or ‘f’ must be appended to float type literal.

**6.** A double literal may optionally end with‘d’ or ‘D’.

**7.** Java’s default types for integers is int and double for floating-points

**8.** Local variables are slightly different; the compiler never assigns a default value to

an uninitialized local variable. If you cannot initialize your local variable where it is

declared, make sure to assign it a value before you attempt to use it. Accessing an

uninitialized local variable will result in a compile-time error.

Ex: if u declare in main ()/any method as:

int a;

System.out.println (a); // ERROR

Hint: Unlike C/C++ Java does not use **auto** storage class, so **no garbage** values.

**9.**Both **float** and **double** data types were designed especially for scientific calculations, where approximation errors are acceptable. If accuracy is the most prior concern then, it is recommended not to use these data types and use **BigDecimal** class instead.

**Java Version Updates**

**Java 8 and Later :**

You can use an unsigned 32-bit integer. This will have a minimum value of 0 and a maximum value of 2**32**-1.Use the **Integer** wrapper class to use the i**nt** data type as an unsigned integer.

You can use an unsigned 64-bit integer with a minimum value of 0 and and a maximum value of 2**64**-1.Use the **Long** wrapper class to use the **long** data type as an unsigned long integer, also contains methods like comparing Unsigned, divide Unsigned, etc to support arithmetic operations for unsigned long

## II) Reference Types

## Non-primitive data types in Java are not predefined, created by the programmer. Known as ‘reference variables’ or ‘object references’ as they reference a memory location where data is stored. There are three reference type available in Java. They are class type, interface type and array type.

### 1. Class

A class is a user-defined data type from which objects are created. It describes the set of properties or methods that are common to all objects of the same type. It contains fields and methods that represent the behaviour of an object. A class gets invoked by the creation of the respective object.

### 2. Interface

An interface is declared like a class. The key difference is that the interface contains methods that are abstract by default; they have nobody.

### 3. Array

An array is used to hold elements of the same type. It is an object in java, and the array name (used for declaration) is a reference value that carries the base address of the continuous location of elements of an array.