Difference Between Primitive Data Types & Reference Data Types in Java

Helpful Links:

* Java Programming: https://java-programming.mooc.fi/part-5/3-primitive-and-reference-variables
* Geeks for Geeks: https://www.geeksforgeeks.org/primitive-data-type-vs-object-data-type-in-java-with-examples/
* Neso Academy: https://www.youtube.com/watch?v=OmcFVHpb0v0

# Categories of data types

Java has 2 categories of variable data types: primitive and reference.

Examples of primitive data types: int, double, char, boolean

Examples of reference data types: Integer, Double, Character, Boolean, String

Reference data types are also called classes and objects.

How are these similar?

Similar: the syntax of declaring and assigning variables

datatype identifier = value;

examples:

int num3 = 3; // primitive variable

String helloString = "Hello"; // reference variable: String object

Scanner input = new Scanner (System.in);// reference variable: Scanner object

How are these different?

By the value that is stored.

Primitive data type stored values:

It is the literal value that is stored in that method's memory.

Reference data type stored values:

It is the reference of where the object's value is in that method's memory that is stored in memory.

Memory Values:

(Very simplified, very abstract representation.)

Method Memory:

|  |  |  |  |
| --- | --- | --- | --- |
| **Memory Address** | **Data Type** | **Identifier** | **Value** |
| 100 | int | num3 | 3 |
| 101 | String | helloString | 200 |
| 102 | Scanner | input | 201 |
| 103 |  |  |  |

…

|  |  |  |  |
| --- | --- | --- | --- |
| **Memory Address** |  |  |  |
| 200 | Hello |  |  |
| 201 | (lots of info for Scanner object) |  |  |
| 203 |  |  |  |

Why is this significant?

Because Java is pass-by-value! This means that it is the variable's value that is stored in the method's memory that is being passed to a method in the invoking statement.

Specifically, in the invoking statement:

* when it is a primitive data type variable that is the argument being passed to a method, then the literal value is passed
* when is a reference data type variable that is the argument being passed to a method, then the memory address of where that variable's information is passed

Example: