

## Lesson 3: Values, Data Types, and Concatenation

1. **Constant**- a value that cannot be changed while a program is running.
2. **Variable**- a data item that might change its value; used to store temporary data to be used in a program's runtime.

3. **Data types**- type of data inside a variable.
4. The eight primitive data types are called "**primitive**" because they are simple and uncomplicated. Primitive types also serve as the building blocks for more complex data types, called reference types.

Keyword	Description
byte	Byte-length integer
short	Short integer
int	Integer
long	Long integer
float	Single-precision floating point
double	Double-precision floating point
char	A single character
boolean	A Boolean value (true or false)

5. **Identifiers**- the name of the variable that the programmer indicated; used to read and write the variable
  - You cannot use any special characters other than \_ (underscore)
  - You cannot use white spaces
  - You cannot use numbers alone
  - You can use numbers but with letters
6. Variable names:
  - Basically, *variable names must start with a letter and cannot be any reserved keyword*. You must declare them before you can use them.\
  - They conventionally begin with *lowercase letters* to distinguish them from class names, but *variable names can begin with either an uppercase or a lowercase letter*.
  - Declaring variables:
    - datatype identifier;
    - datatype identifier = value;

7. The **equal sign** (=) is the assignment operator.

8. In Java, you use variables of type **int** to store (or hold) integers; an integer is a whole number without decimal places.

Type	Minimum Value	Maximum Value	Size in Bytes
byte	-128	127	1
short	-32,768	32,767	2
int	-2,147,483,648	2,147,483,647	4
long	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	8

9. The operators / and % deserve special consideration. When you perform integer division, whether the two operators used in the arithmetic expression are integer constants or integer variables, the result is an integer.

Operator	Description	Example
+	Addition	45 + 2, the result is 47
-	Subtraction	45 - 2, the result is 43
*	Multiplication	45 * 2, the result is 90
/	Division	45/2, the result is 22 (not 22.5)
%	Remainder (modulus)	45 % 2, the result is 1 (that is, 45/2 = 22 with a remainder of 1)

Operator	Description	True example	False example
<	Less than	3 < 8	8 < 3
>	Greater than	4 > 2	2 > 4
==	Equal to	7 == 7	3 == 9
<=	Less than or equal to	5 <= 5	8 <= 6
>=	Greater than or equal to	7 >= 3	1 >= 2
!=	Not equal to	5 != 6	3 != 3

10. Operator precedence refers to the rules for the order in which parts of a mathematical expression are evaluated.
11. **Boolean logic** is based on true-or-false comparisons; a Boolean variable can hold only one of two values—true or false.
12. A **floating-point number** contains decimal positions. Java supports two floating-point data types: float and double. A float data type can hold floating-point values of up to six or seven significant digits of accuracy. A double data type requires more memory than a float, and can hold 14 or 15 significant digits of accuracy.

Type	Minimum	Maximum	Size in Bytes
float	$-3.4 * 10^{38}$	$3.4 * 10^{38}$	4
double	$-1.7 * 10^{308}$	$1.7 * 10^{308}$	8

13. **Concatenation**- process of joining strings together with the plus operator (ex: "Hello" + "World"). You can concatenate inside a variable or on the Print statement

```
char section = 'A';  
String name = "David";  
boolean isTall = false;  
int num3 = 5;  
float grade = 95.55f;  
double grade1 = 93.25;
```

```
System.out.println(name);
```

```
name = "Alenere";  
isTall = true;  
num3 = 10;
```