Java Programming Language

By
Sunil Kumar(Master of Sc.)
Bangalore, India

Lesson 3

Variables, Data Types & Math Operator

- Variables
- Primitive Data Types
- Arithmetic Operators
- Type Conversion

Variable

- Named Data Storage
- Java is a Strongly Typed
  - It means that we have to store the similar type of variable of which it is declared. Declaration of data type should be defined during the variable declaration.

int p; //Which means ,in the variable p we can only store data type which are compatible with integer only

p = 100;

or

int p = 100; //instead of writing in 2 line we can write in single line i.e. declaring and assigning

Value can be modified or we can say overrided.

Variable-Naming Convention

- Variable naming is based on a combination of rules and conventions.
  - Rules allow the use of letters, numbers, $ and _
  - By convention only letters and numbers are used
  - Rules require the first character is not a number
  - By convention it is always a letter
  - By convention, follow the style often referred to as a camel case
  - First letter is lower case
  - Start of each word after the first is upper case
  - All other letters are lower case
    - String myFirstJavaProject;
    - Int year2001;

Primitive Data Type

- Primitive meaning present early stage or at very basic stage
- Foundation of all other types
- Four categories of primitive data types in Java
  - Integer
  - Floating Point
  - Character
  - Boolean

Primitive Data Type Contd..

- Integer Type
  - 4 different types of integer type is there
  - Vary in size only but nature is always same
  - Please note here when we use long then we need to put L in the end

Type	Size (bits)	Min Value	Max Value	Literal Format
byte	8	-128	127	0
short	16	-32768	32767	0
int	32	-2147483648	2147483647	0
long	64	-9223372036854775808	9223372036854775807	0L

Primitive Data Type Contd..

- Floating Point
  - Implementation of IEEE 754 floating point standard
  - Stores values containing a fractional portion
  - Supports positive negating and zero values

Type	Size (bits)	Smallest Positive Value	Largest Positive Value	Literal Format
float	32	1.4×10^{-45}	3.4×10^{38}	0.0f
double	64	4.9×10^{-324}	1.7×10^{308}	0.0 or 0.0d

Primitive Data Type Contd..

- Character Type
  - Char type store a single Unicode character
  - Literal values placed between single quotes
    - Char accountStatus = ‘S’;
  - Remember this is different than string which we will talk later session.
  - Since char support unicode, for unicode code points, use \u followed by 4-digit hex value
    - Char codeStnd = ‘\u00DA’; //i.e. a U ‘ with upper

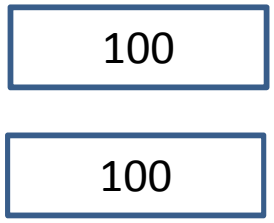
Primitive Data Type Contd..

- Boolean Type
  - Boolean type stores true or false value
    - TURE -> 1 and FALSE -> 0
  - This is very fast during comparison and wherever required to store only 2 output then we use such kind of data type.
  - Conditional operators operate only on *boolean* values. Java has three, **conditional-and**, **conditional-or**, and **conditional-not** like we are having in Digital Circuit.

Primitive Data Type Contd..

- Primitive data types are stored by value.
- We can discuss it with the example we already had.
- When we initialize a variable as below

```
Int firstValue = 100;
Int secondValue = firstValue;
```



- Other language like C, they have pointer to the 1st value i.e. copy by reference but java doesn't

Arithmetic Operator

- Java provides 3 kind of arithmetic operator
  - Basic Operator (+ - * /)
  - Prefix and Postfix Operator (++ --)
  - Compound assignment operator (+= -= /= *= %=)

Arithmetic Operator

	Operator	Floating Point Example	Integer Example
Add	+	1.0 + 2.0 = 3.0	1+2 = 3
Subtract	-	1.0 – 2.0 = -1.0	1-2= -1
Multiply	*	1.0* 2.0 = 2.0	1*2 = 2
Divide	/	13.0/5.0 = 2.6	13/5 = 2
Modulus	%	13.0%5.0 = 3	13%5 = 3

Prefix and Postfix Operators

- ++ increment value by 1
- -- decrement value by 1
- Prefix applies operation before returning value
- Postfix applies operation after returning value
- We can look into one example.

Compound Assignment Operator

- We have Compound assignment operator as
  - +=
  - -=
  - /=
  - *=
  - %=
- How does it work. We will see it through an example.
- Combine an operator and assignment.

Operator Precedence

- We have similar kind of preferences followed here as we are in Mathematics.
- So along with all operator here in the precedence
  - Postfix
  - Prefix
  - Multiplicative / % *
  - Additive + -
- Operator of equal precedence are evaluated left to right.
- We can override precedence with parenthesis.
- Nested parenthesis evaluated from the inside out.

Type Conversion

- In Java we have we have 2 type of conversion
- Implicit Conversion
- Explicit Conversion

Summary on Variables

- Variables are strongly typed in Java
- Primitive types
  - Integer Type
  - Floating Type
  - Char type
  - Boolean Type
- Math Operator
  - Basic Operator, postfix/prefix operator, compound assignment operator
- Math Operator follows a well defined order of preference
- Type Conversion
  - Compiler can automatically apply widening type conversion
  - We use type casting to explicitly perform type conversion.