VIPS
Technical Campus
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

SCHOOL OF ENGINEERING AND TECHNOLOGY

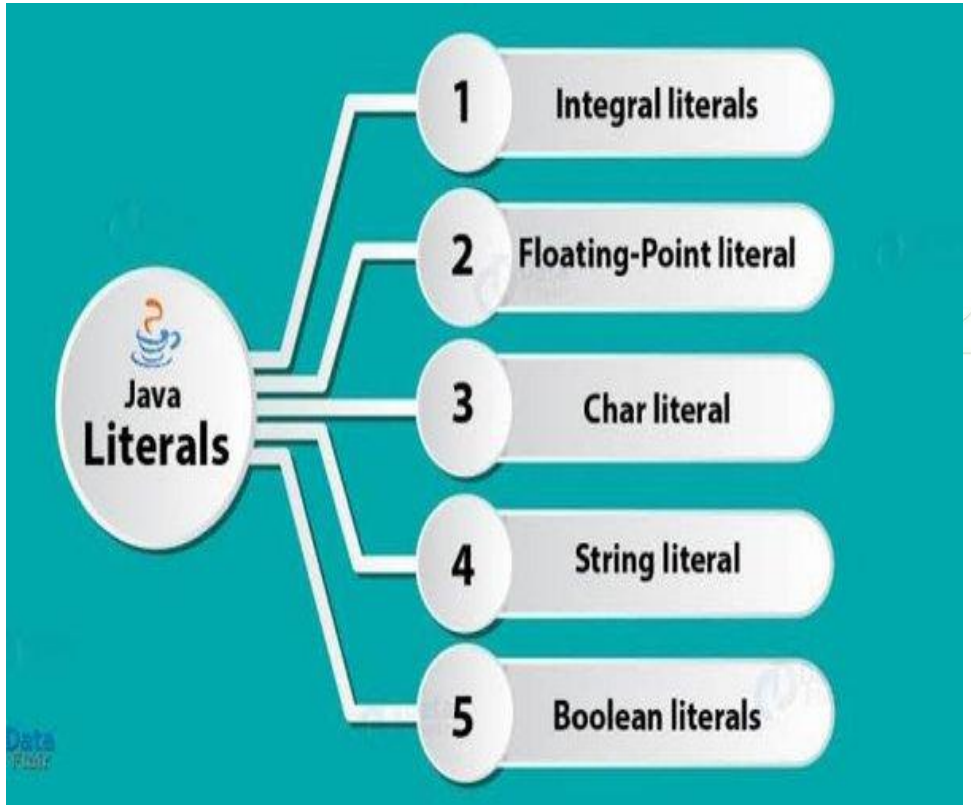
Course : OBJECT ORIENTED PROGRAMMING

Paper Code: IIOT-202,AIML-202

Faculty : Dr. Shivanka
Assistant Professor
VIPS

Literals in Java

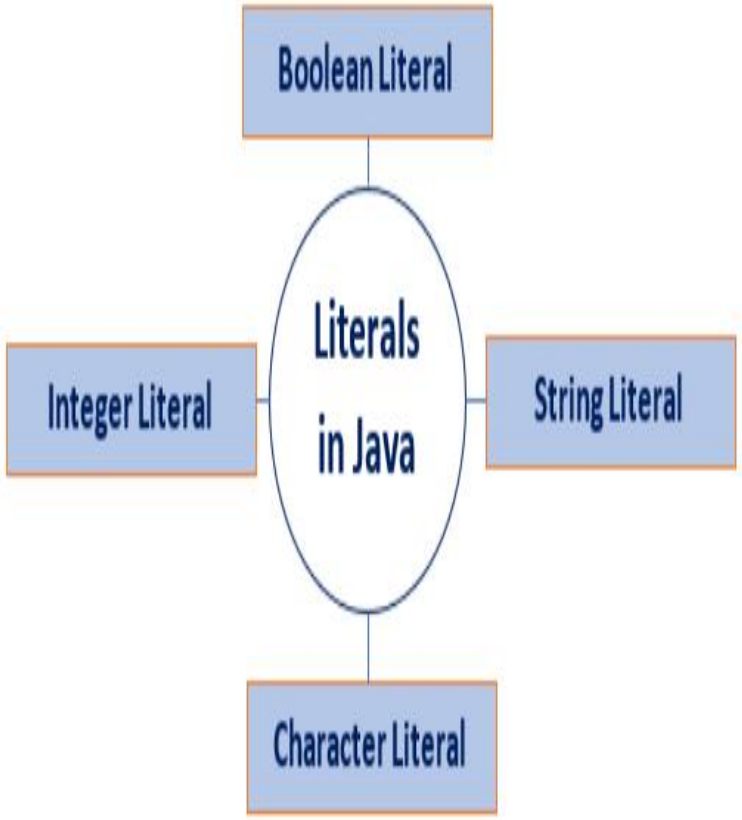
- In Java, literal is a notation that represents a fixed value in the source code.
- In Java, literals are the constant values that appear directly in the program. It can be assigned directly to a variable. Java has various types of literals.



Types of Literals in Java

- There are the majorly four types of literals in Java:

- Integer Literal
- Character Literal
- Boolean Literal
- String Literal



Integer Literal

- ***Integer literals*** are sequences of digits. There are three types of integer literals:
- ***Decimal Integer***: These are the set of numbers that consist of digits from 0 to 9. It may have a positive (+) or negative (-) Note that between numbers commas and non-digit characters are not permitted. For example, 5678, +657, -89, etc.

  int decVal = 26;
- ***Octal Integer:*** It is a combination of number have digits from 0 to 7 with a leading 0. For example, 045, 026, int octVal = 067;
- ***Hexa-Decimal:*** The sequence of digits preceded by 0x or 0X is considered as hexadecimal integers. It may also include a character from a to f or A to F that represents numbers from 10 to 15, respectively. For example, 0xd, 0xf,

  int hexVal = 0x1a;
- ***Binary Integer:*** Base 2, whose digits consists of the numbers 0 and 1 (you can create binary literals in Java SE 7 and later). Prefix 0b represents the Binary system. For example, 0b11010.
- int binVal = 0b11010;

Real Literals and Backslash Literals

- **Real Literals:** The numbers that contain fractional parts are known as real literals. We can also represent real literals in exponent form. For example, 879.90, 99E-3, etc.
- **Backslash Literals**
- Java supports some special backslash character literals known as backslash literals. They are used in formatted output. For example:

1)\n: It is used for a new line

2) \a: It is used for a small beep

3)\r: It is used for carriage return (It moves cursor to current position to first position of the current line and it does not read any character in between)

4) \b: It is used for blank space

5) \': It is used for a single quote

6) \t: It is used for horizontal tab

7) \v: It is used for vertical tab

8) \”: It is used for a double quote

Example of Backslash Literals


```
public class backslashLiterals {
    public static void main(String  args[]){
System.out.println("Hello world");
System.out.println("\"IIOT in Java Class\"");
System.out.println(" welcome \rIIoT");
System.out.println("\n Hello world");
System.out.println("\t Hello world");
}}
```

Output :
Hello world
"IIOT in Java Class"
IIoT

Hello world
Hello world

Character Literals and String Literals

- **Character Literals**

A character literal is expressed as a character or an escape sequence, enclosed in a single quote (') mark. It is always a type of char. For example, 'a', '%', '\u000d', etc.

- **String Literals**

String literal is a sequence of characters that is enclosed between double quotes ("") marks. It may be alphabet, numbers, special characters, blank space, etc. For example, "Jack", "12345", "\n", etc.

Character Literals & String Literals

- ***Character Literals***
- A character literal is expressed as a character or an escape sequence, enclosed in a single quote (') mark. It is always a type of char. For example, 'a', '%', '\u000d', etc.

- ***String Literals***
- String literal is a sequence of characters that is enclosed between double quotes ("") marks. It may be alphabet, numbers, special characters, blank space, etc. For example, "Jack", "12345", "\n", etc.

Floating Point Literals

- ***Floating Point Literals***
- The values that contain decimal are floating literals. In Java, float and double primitive types fall into floating-point literals. Keep in mind while dealing with floating-point literals.
- Floating-point literals for float type end with F or f. For example, 6f, 8.354F, etc. It is a 32-bit float literal.
- Floating-point literals for double type end with D or d. It is optional to write D or d. For example, 6d, 8.354D, etc. It is a 64-bit double literal.
- It can also be represented in the form of the exponent.
- Floating:

- float length = 155.4f;

Decimal & Boolean Literals

Decimal:

double interest = 99658.445;

Decimal in Exponent form:

double val= 1.234e2;

Boolean Literals

Boolean literals are the value that is either true or false. It may also have values 0 and 1. For example, true, 0, etc.

boolean isEven = true;

Null Literals

Null literal is often used in programs as a marker to indicate that reference type object is unavailable. The value null may be assigned to any variable, except variables of primitive types.

String stuName = null;

Student age = null;

Class Literals & Invalid Literals

Class Literals

Class literal formed by taking a type name and appending .class extension. For example, Scanner.class. It refers to the object (of type Class) that represents the type itself.

```
class classType = Scanner.class;
```

Invalid Literals

There is some invalid declaration of literals.

```
float g = 6_.674f;
float g = 6._674F;
long phoneNumber = 99_00_99_00_99_L;
int x = 77_;
int y = 0_x76;
int z = 0X_12;
int z = 0X12_;
```

Restrictions to Use Underscore (_)

It can be used at the beginning, at the end, and in-between of a number.

It can be adjacent to a decimal point in a floating-point literal.

Also, can be used prior to an F or L suffix.

In positions where a string of digits is expected.

Why use literals?

To avoid defining the constant somewhere and making up a label for it. Instead, to write the value of a constant operand as a part of the instruction.

How to use literals?

A literal in Java can be identified with the prefix =, followed by a specific value.

Example : using Literals

LiteralsExample.java

```
public class LiteralsExample
{
public static void main(String args[])
{
int count = 987;
float floatVal = 4534.99f;
double cost = 19765.567;
int hexaVal = 0x7e4;
int binary = 0b11010;
char alpha = 'p';
String str = "Java";  boolean boolVal = true;
int octalVal = 067;
String stuName = null;
```

- char ch1 = '\u0021';
- char ch2 = 1456;
- System.out.println(count);
- System.out.println(floatVal);
- System.out.println(cost);
- System.out.println(hexaVal);
- System.out.println(binary);
- System.out.println(alpha);
- System.out.println(str);
- System.out.println(boolVal);
- System.out.println(octalVal);
- System.out.println(stuName);
- System.out.println(ch1);
- System.out.println("\t" +"backslash literal");
- System.out.println(ch2);
- } }

Output : 987
4534.99
19765.567
2020
26
p
Java
true
55
null
!
	backslash literal
?

Thank You