60 days of Java from 0 to Hero series

Day 7
Literals

Hariprasath V

Literals

- Literal is a fixed value written in the code (like 10,'A', "Hello",3.14f)
- A constant value which can be assigned to the variable is called literals.

```
// Literal used without assignment
System.out.println(100);  // 100 is an integer literal used directly here
```



```
int x = 100;
int num = 10;          // 10 is an integer literal
float pi = 3.14f;      // 3.14f is a float literal
char letter = 'A';     // 'A' is a char literal
String name = "John"; // "John" is a String literal
```

1) Integral literals

- For integral data types [byte,short,int,long]

we can specify litreal value in the following ways

1)Decimal literals(base 10)

- Allowed digits are 0 to 9

2)Octal form (base 8)

- Allowed digits are 0 to 7 .
- Literal value will be prefixed with zero

```
int value = 010;
```

3)Hexa decimal form (base 16)

- Allowed digits are 0 to 9,a to f

By deafult java case sensitive

- For extra digits (a to f) we can use both lower case and upper case characters.
- This is one of very few areas where java is not case sensitive Literal value should be prefixed with 0x or 0X

```
int value = 0X10
```

These are only possible way to define literal value for integral data types


```
int decimalValue = 10;          //10
int octalValue = 010;           // 9
int hexDecimalValue = 0xFace; //64206
```

- Program having chance to specific value decimal,octal,hexadecimal But JVM will always provide value in the form of decimal form
- By default every integral literal is of int type but we can specify explictly as long type by suffixed with l or Label

```
int value = 10;
long value = 10L;
int value  = 10L;
long value  = 10;
```

line 3 -Compile time Error: incompatible types: possible lossy conversion from long to int

Hariprasath V

- There is no direct way to specify byte and short literal explictly but indirectly we can specify when never we are assigning integral literal to the byte variable and its the value with in the range of byte then compiler treats it automatically as byte literal similarly short literal also

```
byte value = 127;
```

```
byte value = 128;
short value   = 32767;
short value = 32768;
```


2) Floating point literal

- By deafult every floating point literal is of double type and hence we can't assign directly to the float variable but we can specify floating point literal as float type by suffixed with small f or F

```
float value = 34f;          //34.0
double value = 12345678.901300;
System.out.println(value); //1.23456789013E7
System.out.println(String.format("%.5f",value));
```


12345678.90130

- we can specify explictly floating point literal as doubled type by suffixed with small d or D offcourse this conversion is not required

we can't assign floating point literal to integral types

```
double value = 10;
int value = 10.0;
double value = 1.2e3;
float value = 1.2e3;
float value = 1.2e3f;
```


Line no 2: error: incompatible types: possible lossy conversion from double to int

Line 4: error: incompatible types: possible lossy conversion from double to float

3) Boolean Literals

- The only allowed values for boolean data type of true or false

```
boolean isValid = true;
boolean isVerified = 4;
```

Line no 2 compile time error: incompatible types: int cannot be converted to boolean
boolean isVerified = 4;

4)char literal

- we can specify char literal as single character with in single quotes
- we can specify char literal has integral literal which represent unique code value of the character and that integral can be specified either in decimal or octal or hexadecimal forms but allowed range is 0 to 65535

Boolean

- The actual size depends on the JVM and may vary, but conceptually, it is 1 bit because there are only two values: true or false.

```
boolean isValid= true;
boolean isVerified = false;
```

Char

- It is used to store a single character.
- The character must be surrounded by single quotes, like 'A' or 'c':
- We can represent char using ASCII code also.

```
char initial = 'v;'
char initial = 98; //a
char initial = 0111; //I
char initial = 0X45; //E
```

- We can represent char literal in unicode representation which is nothing but single quotes backslash

```
char initial = '\\u0045'; \\E
char initial = '\\u007a';  \\z
```


Non-Primitive Data Types

String

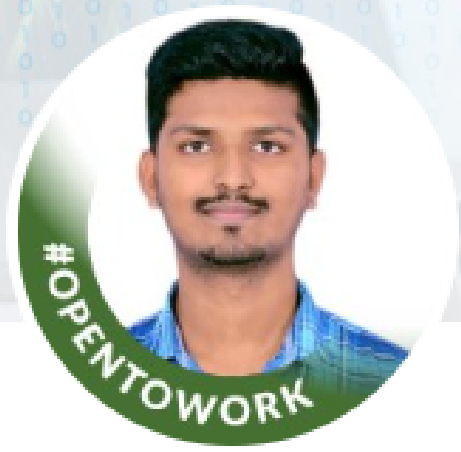
String name = "hari";

String literal we will we see in seperate session becaue it have more content

HARIPRASATH V

Java | SpringBoot | Microservices | Kafka | React.js | Python

hariprasathv6@gmail.com +91 9080555464



+ Follow

Hariprasath V

Add verification badge

Anna University Chennai

Java Full Stack Developer | Spring Boot | Microservices | Kafka | React Js | AWS | Data Structures and Algorithms | Javascript | Python

Chennai, Tamil Nadu, India · Contact info

Give Like content is helpful

Click below link to follow on linkedIn :

https://www.linkedin.com/in/hariprasathv6/

+ Follow **Follow me on linked in get daily tech contents**

REPOST **if you want others also learn and grow together**

Hariprasath V