Java Basics: Understanding Literals, Type Conversion, and Constants

This guide is meant for Java beginners. It explains some of the basic but important concepts like **Literals**, **Type Conversion**, and **Constants** in Java, using simple examples to help you understand these ideas.

Table of Contents

Literals

What Are Literals?

In Java, **literals** are fixed values that you directly assign to variables. For example, in the line:

```java
int number = 10;
```

The number `10` is a **literal**. It's a fixed value that doesn’t change.

Types of Literals

Java has different types of literals based on the data type you are using. Let's explore the most common ones:

1. **Integer Literals**:

   - These are numbers without decimals.

   Examples:

   ```java
   int x = 10;  // Decimal literal
   int y = 0xA;  // Hexadecimal literal (0x represents a hex number, 0xA means 10)
   ```

2. **Floating-Point Literals**:

   - These are numbers with decimals (for example, `3.14`).
   - Java uses `double` for these by default. If you want to use `float`, you need to add `f` or `F` at the end.

   Examples:

   ```java
   double pi = 3.14;  // Double literal
   float temperature = 36.6f;  // Float literal (must add 'f')
   ```

3. **Boolean Literals**:

   - These can only be `true` or `false`.

   Example:

   ```java
   boolean isJavaFun = true;  // Boolean literal
   ```

4. **Character Literals**:

   - These represent a single character enclosed in single quotes (`'`).

   Example:

   ```java
   char grade = 'A';  // Character literal
   ```

5. **String Literals**:

   - These are sequences of characters enclosed in double quotes (`"`).

   Example:

```java
String greeting = "Hello, World!";  // String literal
```

Type Conversion

What Is Type Conversion?

Type conversion in Java is when we convert one data type to another. This can happen **automatically** or **manually**.

Implicit Type Conversion

This type of conversion happens **automatically** when you assign a smaller data type to a larger data type. For example, assigning an integer value to a variable of type `double`.

Example:

```java
int x = 5;
double y = x;  // Implicit conversion from int to double
```

Here, `x` is an `int` (integer), but when it's assigned to `y` (which is a `double`), Java automatically converts it to a `double`.

Explicit Type Conversion (Casting)

Sometimes, you need to manually convert a larger data type to a smaller one. This is called **casting**.

Example:

```java
double x = 5.75;
int y = (int) x;  // Explicit conversion from double to int
```

In this example, `x` is a `double` with a decimal. When we assign it to `y` (an `int`), we have to **explicitly cast** it with `(int)`. This will **remove** the decimal part of the number.

Constants

What Are Constants?

A **constant** is a value that cannot be changed once it is set. In Java, you use the `final` keyword to create a constant. Once a constant is assigned a value, it **cannot** be changed.

How to Declare Constants

To declare a constant in Java, use the `final` keyword, followed by the data type and the constant name.

Example:

```
final int MAX_SPEED = 120;  // Constant
```

In this example, `MAX_SPEED` is a constant. Once you assign a value to it, it cannot be changed anywhere else in the program. If you try to do something like `MAX_SPEED = 150;`, Java will give you an error.

Summary

This guide has introduced you to the basics of Java:

1. **Literals**: Fixed values like numbers, characters, and strings.
2. **Type Conversion**: The process of changing one data type to another. It can happen automatically (implicit) or manually (explicit casting).
3. **Constants**: Values that cannot be changed once defined, created using the `final` keyword.

These are foundational concepts that every Java programmer should understand. With this knowledge, you'll be able to work with variables, constants, and data types more effectively in your programs.
