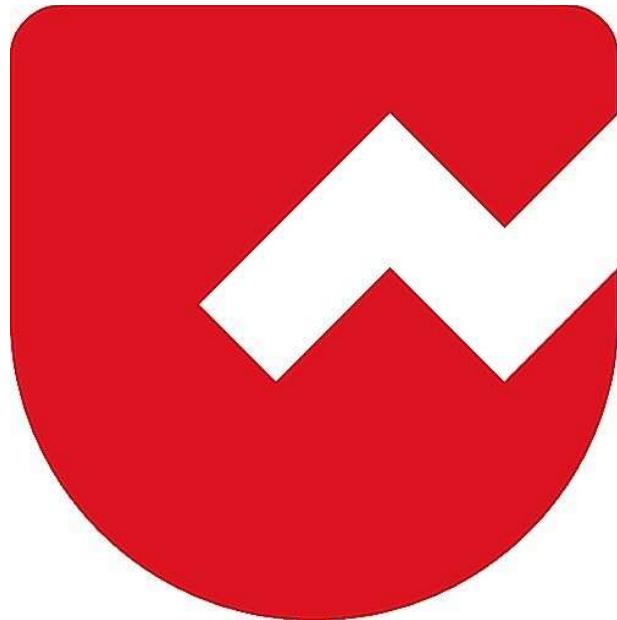




Zoho Schools for Graduate Studies



Notes

JAVA LANGUAGE BASICS PART -1

IDENTIFIERS

Identifiers are names given to identify various programming elements(components) such as variables, constants, array name, string name, method name, class name, etc

RULES FOR IDENTIFIERS IN JAVA

- **Can contain → letters (A–Z, a–z), digits (0–9), underscore (_), and dollar sign (\$).**
Example: myVar, _value, \$amount, student1
- **Cannot start with a digit.**
1student (invalid)
student1 (valid)
- **Cannot use Java keywords (reserved words).**
class, int, public
- **Case-sensitive** → Age and age are different.
- **No spaces or special characters** (like @, -, #).
total marks
student-name
- **Should be meaningful** (for readability).
studentName instead of sn
- **No length limit**, but keep it reasonable.

Valid: name, age1, _rollNo, \$salary

Invalid: 123name, class, student-name, total marks

- **Total Special Symbols – 32**
- **In Java Identifiers we use only 2 special character (_, \$)**

RULES FOR NAMING IDENTIFIERS IN JAVA

- If several words are linked together to form a name for an identifier, then the first letter of the inner words should be capitalized (upper case)- Camel Case Notation
- Class names and interface names must begin with uppercase (Pascal Case Notation)
 - a) class names should be typically be nouns
 - b) interface names should be typically be adjectives
- Methods and variables must begin with a lower case
 - a) method names should be typically verb-noun pairs
 - b) variable names should be short and meaningful
- Constants must be fully capitalized with underscore connecting multiple words

ADVANTAGES OF NAMING IDENTIFIERS IN JAVA

- ✓ Improved Code Readability
- ✓ Enhanced Maintainability

SOURCE FILE

- A source file is a file that contains Java code, usually defining one or more classes.

- It must have the file extension .java.
- The source file name should exactly match the public class name inside the file.

SOURCE FILE DECLARATION RULES

- Only one public class is allowed per Java source file
- The source file name must exactly match the public class name
- Multiple non-public classes, interfaces, or enums can be included in a single file.
- The package statement (if present) must be the first line in the file.
- Statements should follow the package declaration and appear before the class declaration.
- If there is no package statement, import statements (if any) must come first before class declarations.

- In source file we can create “n” no.of classes
- Every class should contain main method?
Ans: Every class can have main method.

Class contains,

1.PUBLIC

How many Public class can a source file have ?
Ans: Atmost one (0 or 1)

2.NON PUBLIC

How many Non Public class can a source file have?

Ans: "n" no.of Non-public classes

If our source file don't have public class,

We can give any name to source file or we can give any one of the non-public class name.

Can we create "n" number of public class in Notepad?

Ans: No. In Notepad, you can have only one public class per file.

DATATYPES

Java as a Strongly Typed Language

Strong Typing: Java enforces strict type rules. Every variable and expression has a clearly defined type, and all assignments are checked for compatibility. Type mismatches result in compile-time errors.

Primitive Types: Java includes eight primitive (or simple) types:

Integers: byte, short, int, long

Floating-point: float, double

Character: char

Boolean: boolean

JAVA INTEGER TYPES

- **Types Defined:** Java provides four signed integer types:

- byte
- short
- int
- long

- Signed Only: Java does not support unsigned integers (positive-only values).

Type	Size	Range	Common Uses & Notes
byte	8-bit	-128 to 127	Efficient for raw binary data, file/network streams.
short	16-bit	-32,768 to 32,767	Rarely used; sometimes for memory-constrained systems.
int	32-bit	-2,147,483,648 to 2,147,483,647	Most commonly used; ideal for loops, array indexing. Promotes performance due to type promotion.
long	64-bit	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Used for large whole numbers (e.g., astronomical calculations).

- Type promotion: byte and short are automatically promoted to int in expressions, which often makes int more efficient.
- Use long when int isn't sufficient for storing large values.

Type	Precision	Size	Use Case & Notes
<code>float</code>	Single	32-bit	Good for saving memory when precision isn't critical (e.g., dollars and cents). May lose accuracy with very large/small values.
<code>double</code>	Double	64-bit	Preferred for high-precision calculations, especially in scientific or iterative computations. All math functions like <code>sin()</code> , <code>cos()</code> , <code>sqrt()</code> return <code>double</code> .

JAVA CHAR TYPE

- Type Definition: `char` is a 16-bit data type used to store characters.
- Unicode-Based: Java uses Unicode, a universal character set that supports characters from all human.
- Range: `char` values range from 0 to 65,535. There are no negative values.
- ASCII Compatibility:
ASCII characters (0-127) are part of Unicode.
ISO-Latin-1 (0-255) is also supported.

JAVA BOOLEAN TYPE

- Boolean is a primitive type used for logical values.
- Possible Values: Only two – `true` and `false`.
- Usage: Returned by relational operators like `<`, `>`, `==`, etc.
- Required in control statements such as `if`, `while`, and `for`.

How range for data types is calculated?

$$-2^{(n-1)} \text{ to } 2^{(n-1)} - 1$$

- **n**
Total number of bits used to store the value.
- **Sign bit**
The highest-order bit (bit at position n-1) determines the sign:
 - 0 → non-negative
 - 1 → negative
- $-2^{(n-1)}$
This is the **minimum** value.
 - All magnitude bits zero except the sign bit → most negative number.
 - Example for $n = 8$: $-2^7 = -128$.
- $2^{(n-1)} - 1$
This is the **maximum** value.
 - Sign bit zero, all other bits one → largest positive number.
 - Example for $n = 8$: $2^7 - 1 = 127$.

Calculating range for short, int, long, char

- **byte** → 8 bits → range = -128 to 127 (-2^7 to $2^7 - 1$)
- **short** → 16 bits → range = -32,768 to 32,767 (-2^{15} to $2^{15} - 1$)
- **int** → 32 bits → range = -2,147,483,648 to 2,147,483,647 (-2^{31} to $2^{31} - 1$)
- **char** → 16 bits (unsigned) → range = 0 to 65,535 (0 to $2^{16} - 1$)

DERIVED DATA TYPES

In Java, data types are classified into:

1. **Primitive Data Types** → byte, short, int, long, float, double, char, boolean
2. **Derived Data Types** → These are built from primitive types.

Derived Data Types in Java are:

- Arrays
- Classes
- Interfaces
- Strings
- Objects
- Enumerations (enum)

These are called derived because they are created using primitive data types or by combining existing data types.

ESCAPE SEQUENCE

Escape sequences are special character combinations used to represent characters that are difficult or impossible to type directly. They begin with a backslash (\) and are commonly used in character and string literals.

```
System.out.println("She said \"Java is fun!\" ");
```

KEYWORDS IN JAVA

Keywords are special words in the Java language that have predefined meanings and cannot be used as identifiers (like variable names, class names, or method names). They form the core syntax of Java.

As per the official Oracle Java documentation,
There are 51 keywords (in Java SE 17+).

Complete list as per Oracle's official documentation:

abstract, assert, boolean, break, byte, case, catch, char, class, const*, continue, default, do, double, else, enum*, extends, false*, final, finally, float, for, goto*, if, implements, import, instanceof, int, interface, long, native, new, null*, package, private, protected, public, return, short, static, strictfp**, super, switch, synchronized, this, throw, throws, transient, true*, try, void, volatile, while

RESERVED WORDS IN JAVA

Reserved words are words that are set aside by Java for special meaning.

They cannot be used as identifiers (variable names, class names, etc.).

Contextual (Reserved) Keywords

exports, module, non-sealed, open, opens, permits,
provides, record, requires, sealed, to, transitive,
uses, var, when, with, yield

Reserved but Unused Keywords

const, goto, strictfp

Reserved Literals

true, false, null

Are they the same?

- All keywords are reserved words
- But not all reserved words are keywords.

How many keywords and reserved words?