
Normalized Notes for JavaTutorialExamples

1. Java Syntax

- Java programs start with a class and a main method.
- `System.out.println()` → prints output to the console with a newline.
- Syntax rules:
 - Statements end with ;
 - Code blocks are enclosed in { }

2. Java Output

- `System.out.println("Text")` → prints text.
- `System.out.println(123)` → prints numbers.
- Output can handle strings, numbers, booleans, and expressions.

3. Java Comments

- Used to explain code; ignored by compiler.
- Types:
 - `//` → single-line comment.
 - `/* ... */` → multi-line comment.

4. Java Variables

- Variables store data values.
- Syntax: `type variableName = value;`
- Example: `int age = 20; String name = "Sridhar";`
- Types: integer, string, float, boolean, etc.

5. Java Data Types

- **Primitive Types:**
 - `int` → whole numbers.

- float → decimal numbers (requires f suffix).
- char → single character.
- boolean → true/false.

- **Non-Primitive Types:**

- String → sequence of characters.
-

6. Java Type Casting

- **Implicit Casting:** automatic conversion from smaller to larger type.
Example: int → double
 - **Explicit Casting:** manual conversion from larger to smaller type.
Example: double → int using (int)
-

7. Java Operators

- **Arithmetic:** + - * / %
 - **Assignment:** = += -= *= /= %=
 - **Comparison:** == != > < >= <=
 - **Logical:** && || !
 - **Bitwise:** & | ^ ~ << >>
-

8. Java Strings

- Strings are objects used to store text.
- Common methods:
 - .length() → number of characters.
 - .toUpperCase() / .toLowerCase() → case conversion.
 - .indexOf("x") → position of substring.
 - .charAt(i) → character at index.
 - .equals(str) / .equalsIgnoreCase(str) → compare strings.
 - .trim() → remove spaces.
 - .compareTo(str) / .compareToIgnoreCase(str) → lexicographic comparison.

- `.concat(str)` → join strings.
 - `.contains(str)` → check substring.
 - `String.copyValueOf(charArray, start, length)` → convert char array to string.
 - `String.join(delimiter, str1, str2, ...)` → join strings with delimiter.
 - `String.format("pattern", values)` → formatted string.
 - `.replace(old, new)` → replace substring.
-

9. Java Math

- Utility class for mathematical operations.
 - Common methods:
 - `Math.max(a, b)` → larger value.
 - `Math.min(a, b)` → smaller value.
 - `Math.sqrt(x)` → square root.
 - `Math.abs(x)` → absolute value.
 - `Math.pow(a, b)` → power (a^b).
 - `Math.round(x)` → nearest integer.
 - `Math.ceil(x)` → round up.
 - `Math.floor(x)` → round down.
 - `Math.random()` → random double $[0,1)$.
 - `(int)(Math.random()*n)` → random integer $[0,n]$.
-

10. Java Booleans

- Boolean type stores true or false.
 - Used in conditions, loops, and logical operations.
 - Example:
`boolean isJavaFun = true; boolean isFishTasty = false;`
-

Quick Summary (Cheat-Sheet)

Section	Concept	Key Notes
Syntax	Structure	Class, main method, println
Output	Display	Text, numbers, expressions
Comments	Notes	//, /* */
Variables	Storage	type name = value
Data Types	Primitive & Non-Primitive	int, float, char, boolean, String
Type Casting	Conversion	Implicit & Explicit
Operators	Arithmetic, Assignment, Comparison, Logical, Bitwise	Perform operations
Strings	Methods	length, trim, equals, concat, replace, etc.
Math	Functions	max, min, sqrt, pow, round, random
Booleans	True/False	Used in logic and conditions
