**1. Variables and Constants**

**Variables**:

* Containers to store data.
* **Declaration**: Declare before use.

int age; // Declare

age = 25; // Assign

int age = 25; // Declare and initialize

**Constants**:

* Variables whose values cannot change after initialization. Declared with final.

final int MAX\_AGE = 100; // Constant

**2. Data Types in Java**

**Primitive Data Types**: Basic types predefined by Java.

| **Data Type** | **Size** | **Description** |
| --- | --- | --- |
| byte | 1 byte | 8-bit signed integer |
| short | 2 bytes | 16-bit signed integer |
| int | 4 bytes | 32-bit signed integer |
| long | 8 bytes | 64-bit signed integer |
| float | 4 bytes | 32-bit floating-point number |
| double | 8 bytes | 64-bit floating-point number |
| char | 2 bytes | Single 16-bit Unicode character |
| boolean | 1 byte | True or false |

Examples:

int x = 10;

char c = 'A';

boolean flag = true;

**Non-Primitive Data Types**: More complex types like String, arrays, and user-defined classes.

String name = "John";

int[] numbers = {1, 2, 3};

**3. Type Casting**

**Implicit Casting**: Smaller type to larger type, handled automatically.

int x = 10;

double y = x; // int to double

**Explicit Casting**: Larger type to smaller type, done manually.

double x = 9.78;

int y = (int) x; // double to int (fractional part lost)

**4. Operators in Java**

**Arithmetic Operators**: Perform mathematical operations.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| + | Addition | a + b |
| - | Subtraction | a - b |
| \* | Multiplication | a \* b |
| / | Division | a / b |
| % | Modulus | a % b |

Example:

int a = 10, b = 5;

System.out.println(a + b); // 15

System.out.println(a % b); // 0

**Relational Operators**: Compare values, return boolean results.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| == | Equal to | a == b |
| != | Not equal to | a != b |
| > | Greater than | a > b |

Example:

int a = 10, b = 5;

System.out.println(a > b); // true

System.out.println(a == b); // false

**Logical Operators**: Combine boolean expressions.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| && | Logical AND | a && b |
| ` |  | ` |
| ! | Logical NOT | !a |

Example:

boolean a = true, b = false;

System.out.println(a && b); // false

System.out.println(a || b); // true

**Assignment Operators**: Assign values to variables.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| = | Assign | a = b |
| += | Add and assign | a += b |

Example:

int a = 10;

a += 5; // a = 15

**Bitwise Operators**: Work at the bit level.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| & | Bitwise AND | a & b |
| ` | ` | Bitwise OR |

**Unary Operators**: Operate on a single operand.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| + | Unary plus | +a |
| - | Unary minus | -a |
| ++ | Increment | a++ |

**Ternary Operator**: Shorthand for if-else.

int a = 10, b = 5;

int result = (a > b) ? a : b; // result = 10