What is a variable in Java?

What is a data type and why is it needed?

What are the three approaches to declare variables in Java?

What is the difference between declaration and assignment?

What are the two main categories of data types in Java?

What primitive data types are used for numbers WITHOUT decimals?

What primitive data types are used for decimal numbers?

What is the difference between char and String?

A data type represents the type of data stored in a variable. In Java, you MUST specify a data type before creating a variable. Without a data type, you cannot create a variable.

A variable is a container that holds data. It represents data and the value can change over time. Example: int x = 100; (x is the variable, 100 is the value)

Declaration: Creating a variable without assigning data (int a;) Assignment: Storing data into the variable (a = 100;) Both together: int a = 100; (declaration + assignment)

Approach 1: int a=100; int b=200; int c=300; Approach 2: int a,b,c; then a=100; b=200; c=300; Approach 3: int a=100, b=200, c=300; (Approaches 2 & 3 work only for same data types)

byte, short, int, long - byte: -128 to 127 (1 byte) - short: -32,768 to 32,767 (2 bytes) - int: -2,147,483,648 to 2,147,483,647 (4 bytes) - long: ±9,223,372,036,854,775,808 (8 bytes)

1. Primitive Data Types: byte, short, int, long, float, double, char, boolean 2. Non-Primitive (Derived/Collections): String, ArrayList, HashMap, HashSet, Arrays

char: Single character, uses SINGLE quotes, primitive type Example: char grade = 'A'; String: Multiple characters, uses DOUBLE quotes, non-primitive type Example: String name = "John";

float and double - float: 4 bytes, stores up to 7 decimal digits (requires 'f' or 'F' literal) - double: 8 bytes, stores up to 15 decimal digits (most commonly used)

What does the boolean data type store?

When do you need to use literals in Java?

How do you choose the right numeric data type?

What is the difference between primitive and non-primitive types?

How do you print variable values in Java?

What is concatenation in Java?

What is a constant variable and how do you create it?

What is the difference between statically typed and dynamically typed languages?

Literals are required for: 1. long: Use 'L' or 'l' at the end (long l = 12345678789L;) 2. float: Use 'F' or 'f' at the end (float price = 15.5F;) Double and int don't require literals.

Boolean stores only true or false (without quotes). It uses 1 bit of memory. Example: boolean bl = true; NOT: boolean bl = "true"; (this is invalid)

Primitive: Stores SINGLE value at a time Example: int x = 100; (only one value) Non-Primitive: Stores MULTIPLE values Example: String s = "Welcome"; (multiple characters) Array: int[] a = new int[100]; (100 values)

Based on the SIZE of your number: - Small numbers (-128 to 127): byte - Medium numbers (-32K to 32K): short - Most common range: int (most moderate) - Very large numbers: long Storing small values in large types wastes memory!

Concatenation means joining multiple values using the + operator. Example: System.out.println("Value: " + a); IMPORTANT: To print a+b correctly with text: System.out.println("Sum: " + (a+b)); Without parentheses, it concatenates instead of adding!

System.out.println(variableName); With message: System.out.println("The value is: " + variableName); Multiple variables: System.out.println(a + " " + b + " " + c); (Use spaces to prevent addition)

Statically Typed (Java): MUST specify data type, CANNOT change type int x = 100; x = "welcome"; // ERROR Dynamically Typed (Python): NO explicit type needed, CAN change type x = 100 (integer) x = "welcome" (now string) ✓

A constant is a variable whose value CANNOT be changed after initialization. Use the 'final' keyword: final int x = 100; x = 200; // ERROR - cannot change Constants ensure values remain fixed throughout the program.

Which are valid/invalid declarations?

What happens with boolean in quotes vs without?

What are the three ways to declare and initialize int variables a, b, c with values 100, 200, 300?

Why do we need variables in programming?

What are the naming rules demonstrated in the examples?

What is the output of: System.out.println(a + b); where a=100, b=200?

When should you use byte vs short vs int vs long?

What is the correct syntax for float and long with literals?

VALID: ✓ boolean bl = true; (correct) ✓ boolean bl = false; (correct) INVALID: ✗ boolean bl = "true"; (this is a String, not boolean) ✗ boolean bl = "false"; (this is a String) Valid with String: ✓ String bl = "true"; (storing as text)

VALID: ✓ String ch = "A"; (string with double quotes) ✓ char grade = 'A'; (char with single quote) INVALID: ✗ char ch = 'ABC'; (multiple chars in single quotes) ✗ String ch = 'A'; (single quotes for string) ✗ char ch = "A"; (double quotes for char)

Variables allow us to: 1. Store data for later use 2. Represent data in memory 3. Change values during program execution 4. Perform operations on data 5. Make programs dynamic and flexible Without variables, we couldn't store or manipulate data!

// Approach 1: Separate declarations int a=100; int b=200; int c=300; // Approach 2: Declare together, assign separately int a,b,c; a=100; b=200; c=300; // Approach 3: Declare and assign together int a=100, b=200, c=300;

Output: 300 (performs addition) BUT: System.out.println("Sum: " + a + b); Output: Sum: 100200 (concatenation!) Correct way: System.out.println("Sum: " + (a+b)); Output: Sum: 300

✓ Use meaningful names: itemPrice, studentId, age ✓ Camel case: itemPrice (first word lowercase, rest capitalized) ✓ No spaces: use underscore or camelCase ✓ Can't reuse same variable name in same scope ✓ Cannot start with numbers Class names: PascalCase (DataTypesDemo, VariablesDemo)

FLOAT: ✓ float price = 15.5f; or 15.5F; ✗ float price = 15.5; (ERROR) LONG: ✓ long l = 12345678789L; or 12345678789l; ✗ long l = 12345678789; (ERROR) Remember: Only float and long require literals!

Use based on data size to avoid memory waste: - byte: Very small numbers (-128 to 127), rare use - short: Small numbers (-32K to 32K), uncommon - int: MOST COMMON, moderate range (±2 billion) - long: Very large numbers, when int range exceeded Rule: Match data type to data size, but int is the default choice.

Where would you use a boolean variable in real programming?

Booleans are used for STATUS and FLAGS: 1. Checking if element exists in array boolean found = false; if(element matches) found = true; 2. Process status boolean isComplete = false; // after process isComplete = true; 3. Validation, conditions, flow control