Lec 10 Data Types

This lesson explains Java's primitive data types, how they store different kinds of values, and why understanding their memory size and range is important for writing efficient programs. It covers how numbers are stored in memory, how to choose the right data type, and the logic behind data type ranges.

Understanding Data Types and Conventions

Programming uses conventions like the Camel case for naming, and basic building blocks such as variables, constants, and keywords. Data in programs is constructed from alphabets, digits, and special symbols, forming statements and complete programs.

The Importance of Data Types in Software

All software processes data, and every piece of data—like names, addresses, numbers—has a specific type. Choosing the correct data type ensures the program can handle and process information accurately.

Types of Data Types in Java

Java has two main data type categories: primitive and non-primitive. Primitive types store simple values directly in memory, while non-primitive types (like objects) are more complex and discussed separately.

Categories of Primitive Data Types

Primitive types are divided into three groups: integers (byte, short, int, long), decimals (float, double), and others (char, boolean). Each group is meant for storing a specific kind of value.

Integer Data Types: Memory and Range

Integer types differ in how much memory they use: byte (1 byte), short (2 bytes), int (4 bytes), long (8 bytes). Each type can store values within a specific range, which is determined by its memory size.

Calculating Data Type Ranges

The range of an integer data type is calculated using powers of two, based on the number of bits it uses. For example, a byte (8 bits) can represent values from -128 to 127.

Choosing the Right Integer Data Type

Selecting the smallest data type that fits your value saves memory. For instance, using a byte for age (which rarely exceeds 127) is more efficient than using an int.

Decimal Data Types: Float and Double

For decimal numbers, use float (4 bytes, about 5 digits after decimal) or double (8 bytes, about 15 digits after decimal). Float is for less precise values, double for higher precision needs.

Other Primitive Types: Char and Boolean

The char type stores a single character using 2 bytes, while boolean stores true or false using the smallest memory (1 bit). Char uses single quotes, and boolean can only be true or false.

Primitive Data Types vs. Objects

Primitive types are stored directly in memory and are not objects, making them faster and simpler to use for basic data storage. They are also called basic data types.

Why Data Type Ranges Are Fixed

The fixed range of each data type comes from its bit size. For example, a byte uses 8 bits, so its range is determined by how many combinations 8 bits can represent, using binary numbers. - 127 → gets represents int 7 bits 1111111 and 127 → 01111111. 0 for +ve and 1 for -ve. Binary of 128 → 010000000 (contains 9 bit which can't be stored).

-128 → 10000000

Binary of -ve number: suppose -128

1) one's complement : 0 → 1 and 1 → 0

2) two's complement : add 1

128 → 10000000

1's comp → 0111111111

2's comp → 0111111111 + 1 = 10000000 → -128

Binary Representation and Data Storage

Numbers are stored in memory as binary (0s and 1s). The largest positive value fits exactly into the available bits, while the leftmost bit indicates if a number is positive or negative.

Handling Negative Numbers in Binary

Negative numbers are stored using a method called two's complement, which allows both positive and negative values to fit in the same number of bits. This explains why the negative range matches the positive range in most integer types.

Recap and Practical Significance

Understanding data types, memory size, and range helps avoid errors and optimize program efficiency. While deep technical questions are rare in most interviews, knowing these details can be useful for advanced roles.