1. Arithmetic Operators: Used for basic mathematical operations (+, -, \*, /, %).
2. Relational (Comparison) Operators: Compare values (==, !=, >, <, >=, <=).
3. Logical Operators: Perform logical operations (&&, ||, !).
4. Bitwise Operators: Operate at the bit level (&, |, ^, ~, <<, >>).
5. Assignment Operators: Assign values to variables (=, +=, -=, \*=, /=, %=).

Summary in English:

Java provides various operators that help in performing operations on variables and values. Arithmetic operators like +, -, \*, /, and % are used for mathematical calculations. Relational operators such as ==,!=, >, <, >=, and <= help in comparing two values and return a Boolean result. Logical operators (&&, ||,!) are used to perform logical operations, mainly in conditions and loops.

Java also has bitwise operators (& |, ^, ~, <<, >>) that perform operations at the binary level, commonly used in low-level programming and optimization tasks. Assignment operators like =, +=, -=, \*=, /= are used to assign and update values efficiently.

Apart from these, Java supports increment (++), decrement (--), ternary (? :), and instance of (instance of) operators, each serving specific purposes. Operator precedence determines the order in which operators are evaluated in an expression, ensuring correct execution. Understanding and using these operators efficiently is crucial for writing optimized Java programs.

తెలుగులో సారాంశం:

జావాలో ఆపరేటర్లు అనేవి ప్రోగ్రామింగ్‌లో మార్పులను మరియు విలువలను ప్రాసెస్ చేయడానికి ఉపయోగిస్తారు. Arithmetic operators (+, -, \*, /, %) గణిత సంబంధిత లెక్కలు చేయడానికి ఉపయోగిస్తారు. Relational operators (==, !=, >, <, >=, <=) విలువలను పోల్చి నిజం (true) లేదా అబద్ధం (false) ను తిరిగి ఇస్తాయి. Logical operators (&&, ||, !) శరతాలను పరీక్షించడానికి ఉపయోగిస్తారు.

Bitwise operators (&, |, ^, ~, <<, >>) బిట్ స్థాయిలో ఆపరేషన్లు నిర్వహిస్తాయి, ఇవి లో-లెవెల్ ప్రోగ్రామింగ్ కోసం ఉపయోగపడతాయి. Assignment operators (=, +=, -=, \*=, /=) మార్పులను మంజూరు చేసి అప్‌డేట్ చేయడానికి ఉపయోగిస్తారు.

ఈ వాటితో పాటు, Increment (++), Decrement (--), Ternary (?:), Instance of (instance of) వంటి ప్రత్యేక ఆపరేటర్లు కూడా అందుబాటులో ఉన్నాయి. ఆపరేటర్ ప్రాధాన్యత (precedence) ప్రకారం, ఏ ఆపరేటర్ ముందుగా అమలవాలో నిర్ణయించబడుతుంది. జావాలో ఆపరేటర్లను సమర్థవంతంగా ఉపయోగించడం మెరుగైన ప్రోగ్రామింగ్‌కు అవసరం.