BecomeCoder – 2026 – CSE & IT – Java Topics Covered so Far (as of 10-02-2024)

1. **Basics of Java Programming**
    a. Basic Structure of a Java Program
    b. Using **System.out.print()** or **System.out.println()** for printing something onto the output screen.
    c. Primitive Data types and Variables
    d. Rules to create identifiers (names given to variables, methods, classes etc in Java)
    e. Commands to
        i. Compile a java program (javac filename.java)
        ii. Run a java program (java classname)
    f. Using Scanner class in java.util package to read various types of input
        i. nextInt() -> for int
        ii. nextDouble() -> for double values
        iii. next() -> for strings without spaces
        iv. nextLine() -> for strings with spaces too
        v. etc...
    g. Formatted output using **System.out.printf()**
        i. format specifiers to be used in printf() for various primitive data types
        ii. %[flags][width][.precision]conversion_character
    h. Operators
        i. Arithmetic Operators
        ii. Relational Operators
        iii. Logical Operators
        iv. Assignment Operators
        v. Bitwise Operators
        vi. Instance of Operator
        vii. Ternary Operator (?:)
    i. java.lang package
    j. Methods in Math class
        i. Math.sqrt()
        ii. Math.pow()
        iii. Math.abs()
        iv. Math.ceil()

        v. Math.floor()
        vi. Math.min()
        vii. Math.max()
        viii. etc....

k. creating constants using **final** keyword
l. using
    i. **0b, 0** and **0x** as prefixes for **binary, octal and hexa decimal**
    ii. Example:
        1. int x = 0123; // stores 83 in x as $(0123)_8$ = $(83)_{10}$
        2. int y = 0b1100; // stores 12 in y as $(1100)_2$ = $(10)_{10}$
        3. int z = 0xF; // stores 16 in z as $(0xF)_{16}$ = $(16)_{10}$
m. Converting integers to Strings and vice versa using **Integer** class in java.lang package
    i. Integer.parseInt(String s)
    ii. Integer.parseInt(String s, int radix)
    iii. Integer.toString(int i)
    iv. Integer.toString(int i, int radix)
    v. Integer.toOctalString()
    vi. Integer.toBinaryString()
    vii. Integer.toHexString()
n. Scope of variables
    i. instance variables
    ii. class / static variables
    iii. local variables
o. static vs. non-static members of a class
p. Naming conventions in Java
    i. Using **camelCase** for variables (Variables are almost always nouns)
        1. firstName
        2. studentMarks
        3. totalAmountToBePaid
        4. currentAccoutBalance
    ii. Using **camelCase** for methods too (Methods are almost always verbs)
        1. getAccountBalance()
        2. createHistory()

            3. deleteBrowsingHistoryNow()
            4. wathBreakingBad()
        iii. Using **TitleCase** for class names
            1. PalindromeCheck
            2. ArrayList
            3. FirstJavaProgram
            4. AdityaEngineeringCollege

2. **Conditional & Selection Statements**
    a. Using if, else if and else for decision making
    b. Switch statement in Java
3. **Looping / Iterative Statements**
    a. while loop
    b. for loop
    c. Loop control / transfer statements (break, continue)
4. **Arrays**
    a. Creating one dimensional arrays in Java
    b. Indexing on arrays
    c. Iterating on an array using indices
    d. for each loop on 1-D arrays
    e. **Arrays** class from **java.util** package and methods in it.
        i. Arrays.sort()
        ii. Arrays.fill()
        iii. Arrays.toString()
        iv. Arrays.compare()
        v. Arrays.equals()
        vi. Arrays.mismatch()
    f. Creating and accessing 2-Dimensional arrays
    g. Arrays.deepToString() method to quickly look at a 2-D array
    h. Running a for each loop on a 2-D array
    i. Array of varying lengths / Variable sized arrays
    j. Cloning arrays for duplication

5. **Strings**
    a. Characters and their **UNICODE** code point values
    b. Using **Character** class and methods in it such as
        i. Character.isUpperCase()
        ii. Character.isLowerCase()
        iii. Character.isDigit()
        iv. Character.isAlphabet()
        v. Character.isWhiteSpace
        vi. Character.toString()
    c. Creating strings in Java
    d. **String Constant Pool (SCP)** in Java
    e. Methos in **String** class
        i. length()
        ii. charAt()
        iii. indexOf()
        iv. lastIndex()
        v. contains()
        vi. startsWith()
        vii. endsWith()
        viii. toLowerCase()
        ix. toUpperCase()
        x. substring()
        xi. **compareTo()**
        xii. **compareToIgnoreCase()**
        xiii. **equals()**
        xiv. **equalsIgnoreCase()**
        xv. isBlank()
        xvi. isEmpty()
        xvii. repeat()
        xviii. trim()
        xix. **toCharArray()**
        xx. split()
    f. Converting a **character array to a string** and vice-versa
    g. Sorting a string
    h. Mutable Strings using **StringBuffer** and **StringBuilder** classes and methods in them
        i. append()

ii. insert()
iii. delete()
iv. setCharAt()
v. deleteCharAt()
vi. reverse()

i. Converting a **String** to **StringBuffer or StringBuilder** object and vice-versa