

Input and Output in Java – Complete Notes

1. Introduction

Input and Output (I/O) in Java is used to read data from input sources and write data to output destinations. Java provides a rich set of classes in the `java.io` and `java.util` packages to handle I/O operations efficiently.

2. Types of Input in Java

Java supports different ways of taking input depending on the use case such as keyboard input, file input, and stream-based input.

2.1 Command Line Input

Command line arguments are passed to the main method while executing the program. They are stored as strings in the `args` array.

Example: `public static void main(String[] args) { System.out.println(args[0]); }`

2.2 Scanner Class

The `Scanner` class (`java.util.Scanner`) is the most commonly used class for taking input from the keyboard. It can read different data types such as `int`, `float`, `double`, and `String`.

- `nextInt()` – reads integer value
- `nextFloat()` – reads float value
- `nextDouble()` – reads double value
- `next()` – reads single word
- `nextLine()` – reads full line

Example:

```
Scanner sc = new Scanner(System.in); int a = sc.nextInt(); String name = sc.next();
```

2.3 BufferedReader Class

`BufferedReader` is used to read text efficiently from character input streams. It is faster than `Scanner` and commonly used in competitive programming.

Example: `BufferedReader br = new BufferedReader(new InputStreamReader(System.in)); String data = br.readLine();`

3. Output in Java

Output in Java is used to display or write data to the console, files, or other output streams.

3.1 System.out.print()

Prints output without moving to a new line.

3.2 System.out.println()

Prints output and moves the cursor to the next line.

3.3 System.out.printf()

Used for formatted output similar to C language formatting.

Example: `System.out.printf("Value = %d", num);`

4. File Input and Output

Java supports file handling using `FileInputStream`, `FileOutputStream`, `FileReader`, and `FileWriter` classes.

4.1 File Input

`FileReader` and `BufferedReader` are commonly used to read data from files.

4.2 File Output

`FileWriter` and `BufferedWriter` are used to write data to files.

5. Byte Streams vs Character Streams

Byte streams handle binary data while character streams handle text data.

6. Common Interview Questions

- Difference between `Scanner` and `BufferedReader`
- What is `System.in` and `System.out`?
- What are streams in Java?
- Difference between `print` and `println`

7. Summary

Java provides flexible and powerful input and output mechanisms suitable for beginners, enterprise applications, and competitive programming.