**Java Basics: Revision Notes**

**Table of Contents**

1. [Introduction to Output in Java](https://companion.scaler.com/meeting/class/858012/notes?_gl=1*1aitnjq*_gcl_gb*R0NMLjE3NDE3MTE0NjQuQ2xrS0NBandtN3EtQmhCbEVra0F4aEFCZU5kUTB0NmFPckdKNlJ2bEQtSWVVLVRaZGo0TmdPTXE5enFGWElNUEMyVFRhMDZoUmFSbWtheE5iMHowcjRHOVRCQlRULUh4cEVTbkZCMUM0eDMzZGQ0enA5Qm5HZ0p5eGcuc0dDakNJLTZ4OGdDRUx5XzA2RUQuQlpJRUNNU2EwWmtaRUx5XzA2RUQ.*_gcl_au*MTIyMjk0OTUwOS4xNzM4MzM3MDIxLjE3ODE5NzQ1NTguMTc0NDI0Mzc1Mi4xNzQ0MjQzNzUx*FPAU*MTIyMjk0OTUwOS4xNzM4MzM3MDIx*_ga*MTEwOTQzNzc1OC4xNjkxODM5NDAy*_ga_53S71ZZG1X*MTc0NDU1Nzk4My4yNDguMS4xNzQ0NTU5MTczLjAuMC4zNzM5MzIwODc.#introduction-to-output-in-java)
2. [Printing Strings and Numbers](https://companion.scaler.com/meeting/class/858012/notes?_gl=1*1aitnjq*_gcl_gb*R0NMLjE3NDE3MTE0NjQuQ2xrS0NBandtN3EtQmhCbEVra0F4aEFCZU5kUTB0NmFPckdKNlJ2bEQtSWVVLVRaZGo0TmdPTXE5enFGWElNUEMyVFRhMDZoUmFSbWtheE5iMHowcjRHOVRCQlRULUh4cEVTbkZCMUM0eDMzZGQ0enA5Qm5HZ0p5eGcuc0dDakNJLTZ4OGdDRUx5XzA2RUQuQlpJRUNNU2EwWmtaRUx5XzA2RUQ.*_gcl_au*MTIyMjk0OTUwOS4xNzM4MzM3MDIxLjE3ODE5NzQ1NTguMTc0NDI0Mzc1Mi4xNzQ0MjQzNzUx*FPAU*MTIyMjk0OTUwOS4xNzM4MzM3MDIx*_ga*MTEwOTQzNzc1OC4xNjkxODM5NDAy*_ga_53S71ZZG1X*MTc0NDU1Nzk4My4yNDguMS4xNzQ0NTU5MTczLjAuMC4zNzM5MzIwODc.#printing-strings-and-numbers)
3. [String Concatenation](https://companion.scaler.com/meeting/class/858012/notes?_gl=1*1aitnjq*_gcl_gb*R0NMLjE3NDE3MTE0NjQuQ2xrS0NBandtN3EtQmhCbEVra0F4aEFCZU5kUTB0NmFPckdKNlJ2bEQtSWVVLVRaZGo0TmdPTXE5enFGWElNUEMyVFRhMDZoUmFSbWtheE5iMHowcjRHOVRCQlRULUh4cEVTbkZCMUM0eDMzZGQ0enA5Qm5HZ0p5eGcuc0dDakNJLTZ4OGdDRUx5XzA2RUQuQlpJRUNNU2EwWmtaRUx5XzA2RUQ.*_gcl_au*MTIyMjk0OTUwOS4xNzM4MzM3MDIxLjE3ODE5NzQ1NTguMTc0NDI0Mzc1Mi4xNzQ0MjQzNzUx*FPAU*MTIyMjk0OTUwOS4xNzM4MzM3MDIx*_ga*MTEwOTQzNzc1OC4xNjkxODM5NDAy*_ga_53S71ZZG1X*MTc0NDU1Nzk4My4yNDguMS4xNzQ0NTU5MTczLjAuMC4zNzM5MzIwODc.#string-concatenation)
4. [Input Handling in Java](https://companion.scaler.com/meeting/class/858012/notes?_gl=1*1aitnjq*_gcl_gb*R0NMLjE3NDE3MTE0NjQuQ2xrS0NBandtN3EtQmhCbEVra0F4aEFCZU5kUTB0NmFPckdKNlJ2bEQtSWVVLVRaZGo0TmdPTXE5enFGWElNUEMyVFRhMDZoUmFSbWtheE5iMHowcjRHOVRCQlRULUh4cEVTbkZCMUM0eDMzZGQ0enA5Qm5HZ0p5eGcuc0dDakNJLTZ4OGdDRUx5XzA2RUQuQlpJRUNNU2EwWmtaRUx5XzA2RUQ.*_gcl_au*MTIyMjk0OTUwOS4xNzM4MzM3MDIxLjE3ODE5NzQ1NTguMTc0NDI0Mzc1Mi4xNzQ0MjQzNzUx*FPAU*MTIyMjk0OTUwOS4xNzM4MzM3MDIx*_ga*MTEwOTQzNzc1OC4xNjkxODM5NDAy*_ga_53S71ZZG1X*MTc0NDU1Nzk4My4yNDguMS4xNzQ0NTU5MTczLjAuMC4zNzM5MzIwODc.#input-handling-in-java)
5. [Data Types: An Overview](https://companion.scaler.com/meeting/class/858012/notes?_gl=1*1aitnjq*_gcl_gb*R0NMLjE3NDE3MTE0NjQuQ2xrS0NBandtN3EtQmhCbEVra0F4aEFCZU5kUTB0NmFPckdKNlJ2bEQtSWVVLVRaZGo0TmdPTXE5enFGWElNUEMyVFRhMDZoUmFSbWtheE5iMHowcjRHOVRCQlRULUh4cEVTbkZCMUM0eDMzZGQ0enA5Qm5HZ0p5eGcuc0dDakNJLTZ4OGdDRUx5XzA2RUQuQlpJRUNNU2EwWmtaRUx5XzA2RUQ.*_gcl_au*MTIyMjk0OTUwOS4xNzM4MzM3MDIxLjE3ODE5NzQ1NTguMTc0NDI0Mzc1Mi4xNzQ0MjQzNzUx*FPAU*MTIyMjk0OTUwOS4xNzM4MzM3MDIx*_ga*MTEwOTQzNzc1OC4xNjkxODM5NDAy*_ga_53S71ZZG1X*MTc0NDU1Nzk4My4yNDguMS4xNzQ0NTU5MTczLjAuMC4zNzM5MzIwODc.#data-types-an-overview)
6. [Typecasting in Java](https://companion.scaler.com/meeting/class/858012/notes?_gl=1*1aitnjq*_gcl_gb*R0NMLjE3NDE3MTE0NjQuQ2xrS0NBandtN3EtQmhCbEVra0F4aEFCZU5kUTB0NmFPckdKNlJ2bEQtSWVVLVRaZGo0TmdPTXE5enFGWElNUEMyVFRhMDZoUmFSbWtheE5iMHowcjRHOVRCQlRULUh4cEVTbkZCMUM0eDMzZGQ0enA5Qm5HZ0p5eGcuc0dDakNJLTZ4OGdDRUx5XzA2RUQuQlpJRUNNU2EwWmtaRUx5XzA2RUQ.*_gcl_au*MTIyMjk0OTUwOS4xNzM4MzM3MDIxLjE3ODE5NzQ1NTguMTc0NDI0Mzc1Mi4xNzQ0MjQzNzUx*FPAU*MTIyMjk0OTUwOS4xNzM4MzM3MDIx*_ga*MTEwOTQzNzc1OC4xNjkxODM5NDAy*_ga_53S71ZZG1X*MTc0NDU1Nzk4My4yNDguMS4xNzQ0NTU5MTczLjAuMC4zNzM5MzIwODc.#typecasting-in-java)
7. [Best Practices for Course and Problem-Solving](https://companion.scaler.com/meeting/class/858012/notes?_gl=1*1aitnjq*_gcl_gb*R0NMLjE3NDE3MTE0NjQuQ2xrS0NBandtN3EtQmhCbEVra0F4aEFCZU5kUTB0NmFPckdKNlJ2bEQtSWVVLVRaZGo0TmdPTXE5enFGWElNUEMyVFRhMDZoUmFSbWtheE5iMHowcjRHOVRCQlRULUh4cEVTbkZCMUM0eDMzZGQ0enA5Qm5HZ0p5eGcuc0dDakNJLTZ4OGdDRUx5XzA2RUQuQlpJRUNNU2EwWmtaRUx5XzA2RUQ.*_gcl_au*MTIyMjk0OTUwOS4xNzM4MzM3MDIxLjE3ODE5NzQ1NTguMTc0NDI0Mzc1Mi4xNzQ0MjQzNzUx*FPAU*MTIyMjk0OTUwOS4xNzM4MzM3MDIx*_ga*MTEwOTQzNzc1OC4xNjkxODM5NDAy*_ga_53S71ZZG1X*MTc0NDU1Nzk4My4yNDguMS4xNzQ0NTU5MTczLjAuMC4zNzM5MzIwODc.#best-practices-for-course-and-problem-solving)

**Introduction to Output in Java**

In Java, the execution of a program starts from the main function. Within the main function, you can print to the console using the System.out.print or System.out.println methods.

public class Main {

public static void main(String[] args) {

System.out.print("Hello, World!"); // prints without a new line

System.out.println("Hello, World!"); // prints with a new line

}

}

* System.out.print: Prints the text in the parentheses on the same line.
* System.out.println: Prints the text in the parentheses and then moves to a new line .

**Printing Strings and Numbers**

**Printing Strings**

To print a string, you place the string inside double quotes within the print or println methods.

System.out.println("Hello, World!"); // Prints: Hello, World!

**Printing Numbers**

Numbers can be printed directly without quotes.

System.out.println(12345); // Prints: 12345

**Combined Example**

When switching between types, it's crucial to understand how they interact.

System.out.print("My age is " + 25); // Prints: My age is 25

* If a string is concatenated with a number, the number becomes part of the string called *String Concatenation* .

**String Concatenation**

Concatenation is combining multiple strings into one by using the + operator.

**Basic Concatenation**

System.out.println("Hello" + " World!"); // Prints: Hello World!

**Concatenation of Strings and Numbers**

The type of the first operand determines the operation:

System.out.println("My age is " + 25 + 10); // Prints: My age is 2510

System.out.println(25 + 10 + " is my age"); // Prints: 35 is my age

* If the first operand is a string, the entire expression is treated as a string .

**Input Handling in Java**

To take input from the user, Java uses the Scanner class.

**Usage of Scanner**

1. Import the Scanner class: import java.util.Scanner;
2. Create a Scanner object: Scanner sc = new Scanner(System.in);
3. Use methods like nextInt(), nextFloat(), nextLine() to read different types of input.

**Example**

Scanner sc = new Scanner(System.in);

// Reading an integer

int num = sc.nextInt();

System.out.println("The number is: " + num);

Multiple input types in a single program:

Scanner sc = new Scanner(System.in);

int intValue = sc.nextInt();

long longValue = sc.nextLong();

double doubleValue = sc.nextDouble();

String stringValue = sc.next(); // for a single word

System.out.println(intValue + ", " + longValue + ", " + doubleValue + ", " + stringValue);

``】

For whole lines, `nextLine()` can be used:

```java

String fullInput = sc.nextLine(); // Reads an entire line

System.out.println(fullInput);

`` .

## Data Types: An Overview

Java supports several data types for storing numbers, characters, and boolean values. Key data types include:

### Integer Data Types

1. \*\*byte\*\*: 1 byte, stores whole numbers from -128 to 127.

2. \*\*short\*\*: 2 bytes, stores whole numbers from -32,768 to 32,767.

3. \*\*int\*\*: 4 bytes, stores whole numbers from -2^31 to 2^31-1.

4. \*\*long\*\*: 8 bytes, stores whole numbers from -2^63 to 2^63-1.

### Floating Point Data Types

1. \*\*float\*\*: 4 bytes, stores fractional numbers up to 7 decimal digits.

2. \*\*double\*\*: 8 bytes, stores fractional numbers up to 16 decimal digits .

### Other Data Types

1. \*\*char\*\*: 2 bytes, stores a single 16-bit Unicode character.

2. \*\*boolean\*\*: 1 bit, stores `true` or `false`.

## Typecasting in Java

Typecasting is converting one data type into another. Types are casted to avoid data loss and to ensure the correct type of data handling.

### Implicit Typecasting (Widening)

```java

int i = 100;

long l = i; // Automatically done, no explicit cast needed

**Explicit Typecasting (Narrowing)**

When converting from a larger to a smaller type, the cast must be explicit to avoid data loss.

double d = 100.04;

int i = (int) d; // d is explicitly casted to int, floating part lost

System.out.println(i); // Prints: 100

Special care must be taken when narrowing to avoid unexpected behavior.

**Best Practices for Course and Problem-Solving**

1. **Attend Classes**: Regular attendance ensures you stay up to date with the material .
2. **Daily Practice**: Work on problems daily or the very next day to retain new concepts better.
3. **Assignments and Homework**: Focus on assignment problems first, then challenge yourself with additional problems .
4. **Engage in Public Forums**: Ask questions publicly to help yourself and others .

Remember to keep pace with assignments and use discussion forums proactively to clarify doubts. Consistency and practice are key to mastering Java programming.

Integer To Long Conversion:

It’s a implicit Type casting where an integer is converted to a long without needing explicit conversion.

Long to Integer Conversion:

Requires explicit typecasting since data might be lost if the long value exceeds the integer range.

Float to Integer conversion:

Converting a float to an integer truncates the functional part causing some information loss.

String Concatenation:

Combining multiple strings into one using the plus (+) operator.

Print VS Println Vs Printf:

Print prints text on the same line while println moves to the next line after printing and printf prints the formatted output.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Adds New Line?** | **Supports Formatting** | **Typical Use** | **Example** |
| print | ❌ No | ❌ No | Simple output on the same line | System.out.print("Hello"); System.out.print("World");  Output: HelloWorld |
| println | ✅ Yes | ❌ No | Output with line breaks | System.out.println("Hello"); System.out.println("World");  Output: Hello World |
| printf | ❌ No | ✅ Yes | Formatted output | double pi = 3.14159; System.out.printf("Pi = %.2f", pi);  Output: Pi = 3.14 |

Byte Data Type:

A data type that uses 1 byte of memory and can store values from -128 to 127.

Short Data Type:

Uses 2 bytes of memory and can store values from -32768 to 32767.

Integer Data Type:

Uses 4 bytes of memory and can store values from -2147483648 to 2147483647.

Long Data Type:

Uses 8 bytes of memory and can store very large integer values.

Character Data Type:

A primitive data type that can store a single 16-bit Unicode character.

Boolean Data Type:

Stores only two possible values: true and false.

Default Data Type:

Whole numbers are by default int, and fractional numbers are by default double.

Typecasting:

Converting one data type to another.

Implicit Typecasting:

Automatic conversion of smaller data type to larger data type by the compiler.

Explicit Typecasting:

Manual conversion of one data type to another using cast operator.

Garbage Value:

Incorrect value stored due to typecasting beyond valid range.

Scanner class:

Class used to take input from user in java.

String Concatenation:

Combining multiple strings into one using the ‘+’ operator.

Integer to Long Typecasting:

Automatically converting an integer to a long type.

Long to Integer Typecasting:

Converting a long to an integer manually; may result in loss of information.

Main Method:

Starting point of execution in a java program.

Primitive Data Types:

Predefine data types in java like int, long, float, double etc.

Non-Primitive Data Types:

User-defined data types in java like arrays, string and classes.

System.out.print:

Method to print output without a newline in java.

Java program start with main method

Public static void main (){

System.out.print(“----");

}