

# TASK 2: Variables, Data Types & Console Input Application.

**Variables** :- A variable in Java is a named memory location used to store data values that can be changed during program execution.

Variable = name + data type + value

Why do we need variables?

Variables are used to:

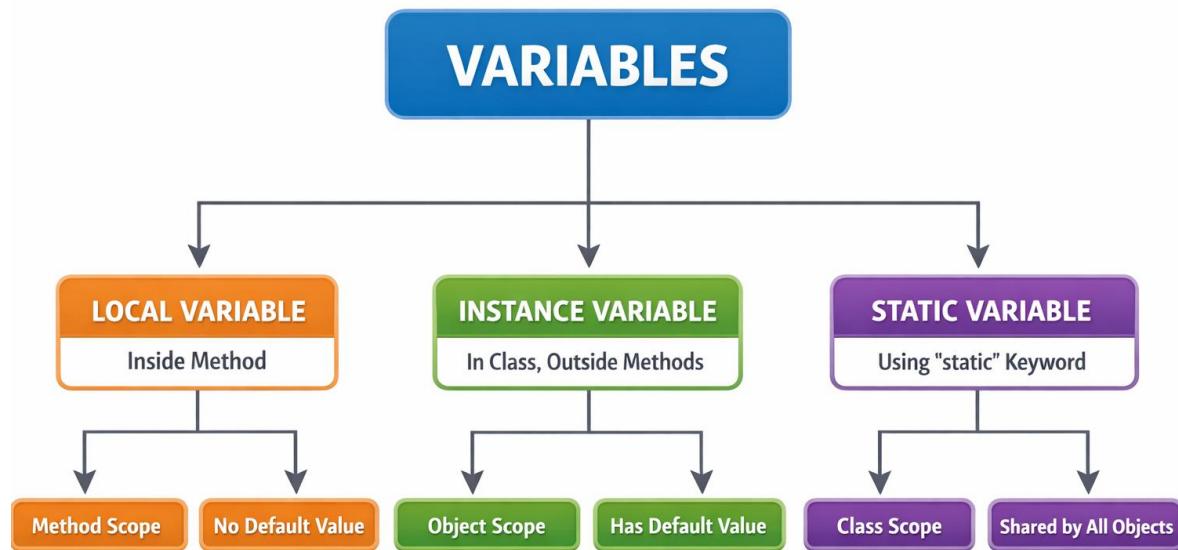
- Store user input
- Store calculation results
- Hold program state
- Reuse data efficiently

Without variables, programs cannot perform meaningful operations.

Syntax of Variable Declaration:- dataType variableName = value;

- Example:- int age = 20;
- int → data type
- age → variable name
- 20 → value

Types of Variables in Java:-



**Local Variable**:- A local variable is declared inside a method, constructor, or block and is accessible only within that block.

Characteristics:

- No default value
- Must be initialized before use
- Stored in stack memory
- Short lifetime

Example Code:

```
class Example
{
    public void show()
    {
        int x = 10; // local variable
        System.out.println(x);
    }
}
```

**Instance Variable**:-An instance variable is declared inside a class but outside any method. Each object gets its own copy of instance variables.

Characteristics:

- Has default value
- Stored in heap memory
- Lifetime = object lifetime
- Accessed using object

Example Code:

```
class Student {  
    int age; // instance variable  
  
    public void display() {  
        System.out.println(age);  
    }  
  
    public static void main(String[] args) {  
        Student s1 = new Student();  
        s1.display(); // prints 0 (default value)  
    }  
}
```

**Static Variable (Class Variable)**:-A static variable is declared using the static keyword. It is shared among all objects of the class.

Characteristics:

- Single copy
- Memory efficient
- Stored in method area
- Accessed using class name

Example Code:

```
class Company {  
    static String companyName = "Tech Corp";  
  
    public static void main(String[] args) {
```

```
        System.out.println(Company.companyName);  
    }  
}
```

## What is a Data Type in Java?

A data type specifies:

- What kind of data a variable can store
- How much memory it occupies
- What operations can be performed on it

Without data types, Java cannot manage memory efficiently.

## Classification of Data Types in Java

- 1.) Primitive Data Types
- 2.) Non-Primitive Data Types

### Example Code (Primitive Types)

```
class PrimitiveDemo  
{  
    public static void main(String[] args) {  
        int age = 20;  
        double salary = 25000.50;  
        char grade = 'A';  
        boolean isActive = true;  
  
        System.out.println(age);  
        System.out.println(salary);  
        System.out.println(grade);  
        System.out.println(isActive);  
    }  
}
```

### Non-Primitive Data Types:-

- String

- Array
- Class
- Interface
- Object

String Example

```
class StringDemo
{
    public static void main(String[] args) {
        String name = "Akash";
        System.out.println(name);
    }
}
```

Array Example

```
class ArrayDemo
{
    public static void main(String[] args) {
        int[] marks = {85, 90, 88};
        System.out.println(marks[0]);
    }
}
```

What is Type Casting?

Converting one data type into another.

- Implicit: small → large (int → double)
- Explicit: large → small (double → int)

What is a Console Input Application?

A console input application is a program that:

- Takes input from the keyboard (console)
- Processes the data
- Displays output in the console window
- Scanner Class (Core of Console Input)

- Step 1:
- Import Scanner

```
import java.util.Scanner;
```

Step 2:

Create Scanner Object

```
Scanner sc = new Scanner(System.in);
```

Step 3:

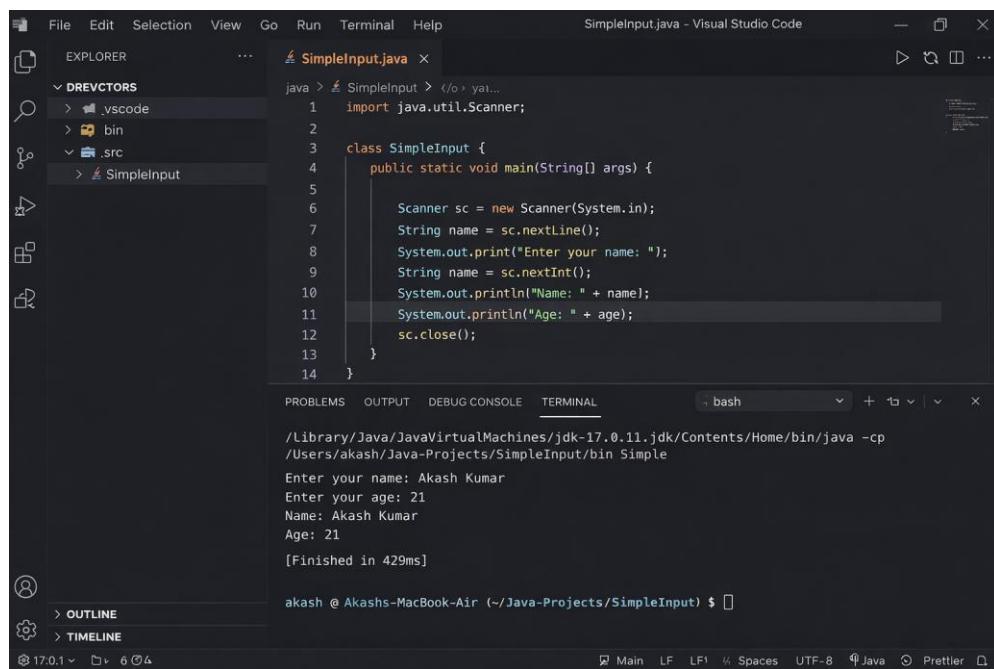
Read Input

```
int a = sc.nextInt();
```

```
double b = sc.nextDouble();
```

```
String name = sc.next();
```

### **Example :- Simple Console Input Program**



```
SimpleInput.java - Visual Studio Code
```

```
File Edit Selection View Go Run Terminal Help
```

```
EXPLORER DREVCOTRS
```

```
SimpleInput.java
```

```
java > SimpleInput > </o> val...
```

```
1 import java.util.Scanner;
2
3 class SimpleInput {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         String name = sc.nextLine();
7         System.out.print("Enter your name: ");
8         String name = sc.nextInt();
9         System.out.println("Name: " + name);
10        System.out.println("Age: " + age);
11        sc.close();
12    }
13
14 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL bash + 10 | v x
```

```
/Library/Java/JavaVirtualMachines/jdk-17.0.11.jdk/Contents/Home/bin/java -cp
/Users/akash/Java-Projects/SimpleInput/bin Simple
```

```
Enter your name: Akash Kumar
Enter your age: 21
Name: Akash Kumar
Age: 21
[Finished in 429ms]
```

```
akash @ Akashs-MacBook-Air (~/Java-Projects/SimpleInput) $
```

```
OUTLINE TIMELINE
```

```
17.0.1 v 6 4
```

```
Main LF LF1 Spaces UTF-8 Java Prettier
```

## Example : Console-Based Calculator

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Calculator.java - Visual Studio Code.
- Explorer View:** Shows a project structure with a folder named "DREVCTORS" containing ".vscode", "bin", and ".src" folders. Inside ".src", there is a file named "Calculator.java".
- Code Editor:** Displays the content of "Calculator.java". The code uses the Scanner class to read two numbers from the user and perform addition, subtraction, and division operations.

```
java > Calculator.java
1 import java.util.Scanner;
2
3 class Calculator {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print("Enter first number: ");
7         double num1 = sc.nextDouble();
8         System.out.print("Enter Second number: ");
9         int age = sc.nextDouble();
10        System.out.println("Addition: " + num1 + num2);
11        System.out.println("Subtraction: " + num1 - num2);
12        if (num2 != 0) {
13            System.out.print("Division: " + num1 / num2);
14        } else {
15            System.out.println("Cannot divide by zero");
16        }
17        sc.close();
18    }
}
```

- Terminal View:** Shows the output of running the Java code. It prompts for two numbers (20 and 5), performs addition (25.0), subtraction (15.0), and division (4.0). The command used is `java -cp /Users/akash/Java-Projects/Calculator/bin Calculator`.

```
/Library/Java/JavaVirtualMachines/jdk-17.0.11.jdk/Contents/Home/bin/java -cp
/Users/akash/Java-Projects/Calculator/bin Calculator
Enter first number: 20
Enter second number: 5
Addition: 25.0
Subtraction: 15.0
Division: 4.0 [I]
[Finished in 324ms]
```

- Status Bar:** Shows the current file is "Calculator.java", the editor mode is "Main", line endings are "LF", and the encoding is "UTF-8". It also indicates Java and Prettier language support.