

How to Build Logic in Programming



Great programmers
don't memorize code,
they build logic.
Learn the right way to
think in programming.

How to Build Logic in Programming

Introduction

Logic building means thinking step by step to solve a problem using **programming**.

Before writing any program, we must understand the problem clearly.

- **Logic** is the base of **Programming**
- **Logic** helps in problem solving
- Every program follows a logical flow

Programming is all about **logic**.

Step 1: Understand the Problem

First read the problem carefully. Identify what is given and what output is required.

Problem: Add two numbers

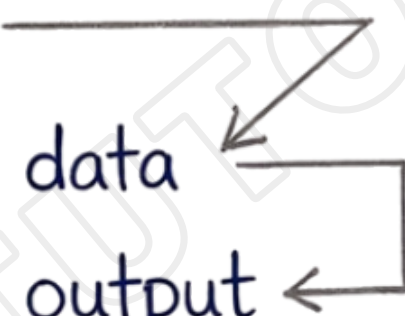
Input: Two numbers

Output: Sum of two numbers

Never start coding without understanding the problem.

Step 2: Break the Problem into Steps

Large problems are difficult to solve directly. So we divide the problem into small and simple **steps**.

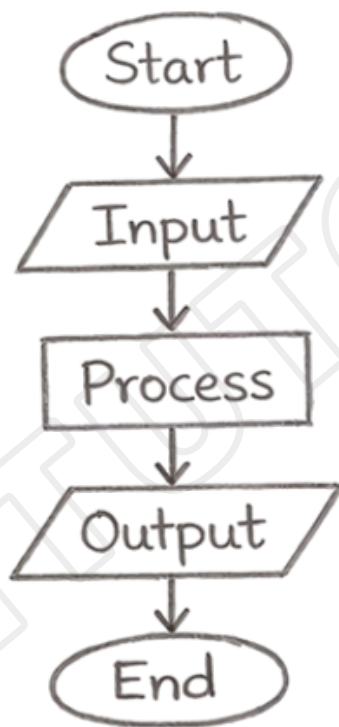
1. Take input
 2. Process the data
 3. Display the output
- 
- ```
graph TD; A[1. Take input] --> B[2. Process the data]; B --> C[3. Display the output]; C --> B;
```

Small steps make logic easy.

### Step 3: Flowchart and Dry Run

A **flowchart** represents the logic using symbols.  
**Dry run** means checking logic using sample values.

Dry run helps to find errors before coding.



Input : 2, 3

Process:  $2 + 3$

Output : 5

@hinditutorials

## Step 4: Convert Logic into Code

Once the logic is clear, convert it into code. Focus on logic first, language comes later.

Strong logic makes programming easy.

Start

Read a, b

$\text{sum} = a + b$

Print sum

End

- Write simple code
- Test the output
- Improve logic with practice

Practice daily to build strong logic.

# Logic Building Practice – Java

## Program 1: Check Odd or Even

Write a program to check whether a number is odd or even.

### Logic

1. Take a number as input
2. Check  $\text{number \% 2}$
3. If remainder is 0, number is Even
4. Otherwise, number is Odd

```
Read number
```

```
If $\text{number \% 2} == 0$
```

```
 Print "Even"
```

```
Else
```

```
 Print "Odd"
```

Use modulo operator to build logic.

## Program 2: Find Largest of Two Numbers

This program compares two numbers and finds the **largest** one.

Logic steps:

1. Take two numbers a and b
2. **Compare** a and b
3. Print the greater number

```
Read a, b
```

```
If $a > b$
```

```
 Print a is largest
```

```
Else
```

```
 Print b is largest
```

Dry run

$a = 5$  ,  $b = 3$

Output: 5 is largest

Comparison builds decision making logic.