Lecture 01: Introduction to C++

What is C++?

C++ is a **programming language** used to write computer programs.

Simple Definition

- A language that humans can write
- Computer understands it and follows instructions
- Used to create software, games, apps

Created By

- **Bjarne Stroustrup** in 1979
- Built on top of C language
- Made to be fast and powerful

Why Do We Need C++?

1. Speed and Performance

C++ programs run very fast.

Used in:

- Games (GTA, Call of Duty, Fortnite)
- Operating Systems (Windows, macOS)
- Browsers (Chrome, Firefox)

2. Control Over Computer

C++ gives direct control over:

- Memory
- Hardware
- CPU

3. Real-World Uses

Field	Examples
Games	Unreal Engine, Unity
Systems	Windows, Linux
Apps	Photoshop, Microsoft Office
Databases	MySQL, MongoDB
Browsers	Chrome (V8 engine)
Embedded	Car systems, Drones
Finance	Trading systems

4. Foundation Language

Learning C++ helps you understand:

- How computers work
- Memory management
- Other languages become easier

Binary Number System

What is Binary?

Binary = Number system with only **2 digits**: `0` and `1`

- Computers only understand 0 and 1
- 0 = OFF (no electricity)
- 1 = ON (electricity flowing)

Why Binary?

Computers use electricity:

- **0** = Low voltage (OFF)
- **1** = High voltage (ON)

Easy for computers to understand ON/OFF states.

Decimal vs Binary

Decimal	Binary	Why?
0	0	Zero
1	1	One
2	10	Two
3	11	Three
4	100	Four
5	101	Five
10	1010	Ten

Decimal to Binary Conversion

Method: Divide by 2

Steps:

1. Divide number by 2
2. Write remainder

3. Divide quotient by 2
4. Repeat until quotient = 0
5. Read remainders from **bottom to top**

Example 1: Convert 13 to Binary

```
Step 1: 13 ÷ 2 = 6, remainder = 1  ↓
Step 2:  6 ÷ 2 = 3, remainder = 0  ↓
Step 3:  3 ÷ 2 = 1, remainder = 1  ↓
Step 4:  1 ÷ 2 = 0, remainder = 1  ↓

Read from bottom to top: 1101
```

Answer: 13 = 1101

Example 2: Convert 25 to Binary

```
25 ÷ 2 = 12, remainder = 1  ↓
12 ÷ 2 = 6,  remainder = 0  ↓
6 ÷ 2 = 3,   remainder = 0  ↓
3 ÷ 2 = 1,   remainder = 1  ↓
1 ÷ 2 = 0,   remainder = 1  ↓

Read from bottom to top: 11001
```

Answer: 25 = 11001

Example 3: Convert 8 to Binary

```
8 ÷ 2 = 4, remainder = 0  ↓
4 ÷ 2 = 2, remainder = 0  ↓
2 ÷ 2 = 1, remainder = 0  ↓
1 ÷ 2 = 0, remainder = 1  ↓

Read from bottom to top: 1000
```

Answer: 8 = 1000

Quick Practice

Decimal	Binary
5	101
7	111
10	1010
15	1111
20	10100

Binary to Decimal Conversion

Method: Powers of 2

Steps:

1. Write powers of 2 from right to left (start with 2^0)
2. Multiply each binary digit with its power
3. Add all results

Powers of 2 Table

Position	Power	Value

0 (rightmost)	2^0	1
1	2^1	2
2	2^2	4
3	2^3	8
4	2^4	16
5	2^5	32
6	2^6	64
7	2^7	128

Example 1: Convert 1101 to Decimal

```
Binary:  1    1    0    1
Power:   2³   2²   2¹   2⁰
Value:   8    4    2    1

Step 1: 1 × 8 = 8
Step 2: 1 × 4 = 4
Step 3: 0 × 2 = 0
Step 4: 1 × 1 = 1

Total: 8 + 4 + 0 + 1 = 13
```

Answer: 1101 = 13

Example 2: Convert 1010 to Decimal

```
Binary:  1    0    1    0
Power:   2³   2²   2¹   2⁰
Value:   8    4    2    1
```

```
Step 1: 1 × 8 = 8
Step 2: 0 × 4 = 0
Step 3: 1 × 2 = 2
Step 4: 0 × 1 = 0

Total: 8 + 0 + 2 + 0 = 10
```

Answer: 1010 = 10

Example 3: Convert 11001 to Decimal

```
Binary:  1    1    0    0    1
Power:   2⁴   2³   2²   2¹   2⁰
Value:   16   8    4    2    1

Step 1: 1 × 16 = 16
Step 2: 1 × 8  = 8
Step 3: 0 × 4  = 0
Step 4: 0 × 2  = 0
Step 5: 1 × 1  = 1

Total: 16 + 8 + 0 + 0 + 1 = 25
```

Answer: 11001 = 25

Example 4: Convert 1111 to Decimal

```
Binary:  1    1    1    1
Power:   2³   2²   2¹   2⁰
Value:   8    4    2    1

Step 1: 1 × 8 = 8
Step 2: 1 × 4 = 4
Step 3: 1 × 2 = 2
Step 4: 1 × 1 = 1

Total: 8 + 4 + 2 + 1 = 15
```

Answer: 1111 = 15

Quick Practice

Binary	Decimal
101	5
111	7
1000	8
10100	20
11111	31

Summary

What is C++?

- Programming language
- Fast and powerful
- Used in games, systems, apps

Why Binary?

- Computers understand only 0 and 1
- 0 = OFF, 1 = ON
- Easy for electronics

Decimal to Binary

- Divide by 2 repeatedly
- Write remainders
- Read bottom to top

Binary to Decimal

- Use powers of 2
- Multiply and add
- Get decimal number

Quick Reference

Decimal to Binary Steps

```
1. Divide by 2
2. Write remainder
3. Repeat
4. Read reverse
```

Binary to Decimal Steps

```
1. Write powers of 2
2. Multiply each digit
3. Add all results
```

Common Conversions

```
Decimal → Binary
0  → 0
1  → 1
2  → 10
3  → 11
4  → 100
5  → 101
10 → 1010
15 → 1111
```

Practice these conversions daily! 🚀

Where C++ is used

1: Database

2: NodeJs

3: Javascript

4: Game Development

5: HFT

6: Machine Learning Libraries(Tensorflow)

1. Variables

What is a Variable?

A **container** that stores data in memory.

```
int age = 25;
```

- `int` = type of container
- `age` = name of container
- `25` = value stored

Rules for Variable Names

✅ **Allowed:**

```
int age;
int student_name;
int rollNo123;
int _value;
```

❌ **Not Allowed:**

```
int 123roll;     // Can't start with number
int student-name; // No hyphens
int int;         // Can't use keywords
int my name;     // No spaces
```

Declaring Variables

```
// Method 1: Declare then assign
int age;
age = 25;

// Method 2: Declare and assign together
int age = 25;
```

```
// Method 3: Multiple variables
int a = 5, b = 10, c = 15;
```

2. Data Types

Basic Data Types

Data Type	Used For	Size	Example
`int`	Whole numbers	4 bytes	`int age = 25;`
`double`	Decimal numbers	8 bytes	`double price = 99.99;`
`char`	Single character	1 byte	`char grade = 'A';`
`bool`	True/False	1 byte	`bool isPassed = true;`
`string`	Text	Varies	`string name = "Rohit";`

Examples

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int age = 25;
    double height = 5.9;
    char grade = 'A';
    bool isStudent = true;
    string name = "Rohit";

    cout << "Name: " << name << endl;
    cout << "Age: " << age << endl;
    cout << "Height: " << height << endl;
    cout << "Grade: " << grade << endl;
    cout << "Student: " << isStudent << endl;
    return 0;
}
```

Important Notes

For `char`: Use single quotes `'A'`

```
char letter = 'A';  // ✅ Correct
char letter = "A";  // ❌ Wrong
```

For `string`: Use double quotes `"text"`

```
string name = "Rohit";  // ✅ Correct
string name = 'Rohit';  // ❌ Wrong
```

Heading 2

3. Strings

Basic String Operations

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string name = "Rohit";
    // Length of string
    cout << name.length() << endl;  // 5

    // Combine strings
    string firstName = "Rohit";
    string lastName = "Kumar";
    string fullName = firstName + " " + lastName;
    cout << fullName << endl;  // Rohit Kumar

    // Access characters (starts from 0)
    cout << name[0] << endl;  // R
    cout << name[1] << endl;  // o

    return 0;
}
```

String Input with Spaces

Problem:

```
string name;
cin >> name;  // Only reads one word!
// Input: "Rohit Kumar"
// Stores: "Rohit"
```

Solution:

```
string name;
getline(cin, name);  // Reads entire line
// Input: "Rohit Kumar"
// Stores: "Rohit Kumar"
```