

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



$8 \div 2 = 4$, remainder = 0 ↓
 $4 \div 2 = 2$, remainder = 0 ↓
 $2 \div 2 = 1$, remainder = 0 ↓
 $1 \div 2 = 0$, 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	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
8	2^8	256
9	2^9	512
10	2^{10}	1024
11	2^{11}	2048
12	2^{12}	4096
13	2^{13}	8192
14	2^{14}	16384
15	2^{15}	32768
16	2^{16}	65536
17	2^{17}	131072
18	2^{18}	262144
19	2^{19}	524288
20	2^{20}	1048576
21	2^{21}	2097152
22	2^{22}	4194304
23	2^{23}	8388608
24	2^{24}	16777216
25	2^{25}	33554432
26	2^{26}	67108864
27	2^{27}	134217728
28	2^{28}	268435456
29	2^{29}	536870912
30	2^{30}	1073741824
31	2^{31}	2147483648
32	2^{32}	4294967296
33	2^{33}	8589934592
34	2^{34}	17179869184
35	2^{35}	34359738368
36	2^{36}	68719476736
37	2^{37}	137438953472
38	2^{38}	274877906944
39	2^{39}	549755813888
40	2^{40}	1099511627776
41	2^{41}	219902325552
42	2^{42}	439804651104
43	2^{43}	879609302208
44	2^{44}	1759218604416
45	2^{45}	3518437208832
46	2^{46}	7036874417664
47	2^{47}	14073748835328
48	2^{48}	28147497670656
49	2^{49}	56294995341312
50	2^{50}	112589990682624
51	2^{51}	225179981365248
52	2^{52}	450359962730496
53	2^{53}	900719925460992
54	2^{54}	1801439850921984
55	2^{55}	3602879701843968
56	2^{56}	7205759403687936
57	2^{57}	14411518807375872
58	2^{58}	28823037614751744
59	2^{59}	57646075229503488
60	2^{60}	115292150459006976
61	2^{61}	230584300918013952
62	2^{62}	461168601836027904
63	2^{63}	922337203672055808
64	2^{64}	1844674407344111616
65	2^{65}	3689348814688223232
66	2^{66}	7378697629376446464
67	2^{67}	14757395258752892928
68	2^{68}	29514790517505785856
69	2^{69}	59029581035011571712
70	2^{70}	118059162070023143424
71	2^{71}	236118324140046286848
72	2^{72}	472236648280092573696
73	2^{73}	944473296560185147392
74	2^{74}	1888946593120370294784
75	2^{75}	3777893186240740589568
76	2^{76}	7555786372481481179136
77	2^{77}	1511157274496292238272
78	2^{78}	3022314548992584476544
79	2^{79}	6044629097985168953088
80	2^{80}	12089258195970337906176
81	2^{81}	24178516391940675812352
82	2^{82}	48357032783881351624704
83	2^{83}	96714065567762703249408
84	2^{84}	193428131135525406498816
85	2^{85}	386856262271050812997632
86	2^{86}	773712524542101625995264
87	2^{87}	1547425049084203251990528
88	2^{88}	3094850098168406503981056
89	2^{89}	6189700196336813007962112
90	2^{90}	1237940039267362601592424
91	2^{91}	2475880078534725203184848
92	2^{92}	4951760157069450406369696
93	2^{93}	9903520314138900812739392
94	2^{94}	1980704062827780162547872
95	2^{95}	3961408125655560325095744
96	2^{96}	7922816251311120650191488
97	2^{97}	1584563252262240130038296
98	2^{98}	3169126504524480260076592
99	2^{99}	6338253009048960520153184
100	2^{100}	1267650601809792104030632



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:	2^3	2^2	2^1	2^0
Value:	8	4	2	1

Step 1: $1 \times 8 = 8$

Step 2: $1 \times 4 = 4$

Step 3: $0 \times 2 = 0$

Step 4: $1 \times 1 = 1$

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

Answer: $1101 = 13$

Example 2: Convert 1010 to Decimal

Binary:	1	0	1	0
Power:	2^3	2^2	2^1	2^0
Value:	8	4	2	1



Step 1: $1 \times 8 = 8$

Step 2: $0 \times 4 = 0$

Step 3: $1 \times 2 = 2$

Step 4: $0 \times 1 = 0$

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

Answer: $1010 = 10$

Example 3: Convert 11001 to Decimal

Binary: 1 1 0 0 1

Power: 2^4 2^3 2^2 2^1 2^0

Value: 16 8 4 2 1

Step 1: $1 \times 16 = 16$

Step 2: $1 \times 8 = 8$

Step 3: $0 \times 4 = 0$

Step 4: $0 \times 2 = 0$

Step 5: $1 \times 1 = 1$

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

Answer: $11001 = 25$

Example 4: Convert 1111 to Decimal

Binary: 1 1 1 1

Power: 2^3 2^2 2^1 2^0

Value: 8 4 2 1

Step 1: $1 \times 8 = 8$



Step 2: $1 \times 4 = 4$

Step 3: $1 \times 2 = 2$

Step 4: $1 \times 1 = 1$

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.



Generated by Notion to PDF

```
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"
```





Generated by Notion to PDF