Day-01 Introduction to C++

1. What is C++?

Answer:

C++ is a computer programming language used to build software, games, operating systems, and many applications.

It is **fast**, **powerful**, and supports **object-oriented programming** (like classes and objects).

It is basically an improved version of the C language.

2. What is a Compiler?

Answer:

A compiler is a program that **converts the code you write** (like C++ code) into **machine language** (0s and 1s) that the computer can understand.

Without a compiler, the computer cannot run your program.

3. What is a Transistor in a Computer?

Answer:

A transistor is a **tiny electronic switch** inside the computer.

It can turn **ON** or **OFF**, and these ON/OFF signals represent **1** and **0** in binary.

Millions and billions of such transistors together form the CPU and memory chips.

4.How to convert any number into Binary?

Answer:

To convert a number into binary, you keep **dividing the number by 2** and note down the **remainders**.

Then you **reverse** the remainders — that becomes your binary number.

Example: Convert 13 to Binary

1. 13 ÷ 2 = 6, remainder **1**
2. 6 ÷ 2 = 3, remainder **0**
3. 3 ÷ 2 = 1, remainder **1**
4. 1 ÷ 2 = 0, remainder **1**

Remainders (bottom to top) = **1101**

So, **13 in binary = 1101**

5.How to convert Binary to Decimal?

Answer:

Binary number ko decimal me convert karne ke liye, har digit ko **2 ki power** se multiply karte hain (rightmost digit se start).

Phir sab values **add** kar dete hain.

Example: Convert 1101 to Decimal

Binary: **1101**

Right se left powers of 2:

- $1 \times 2^0 = 1$
- $0 \times 2^1 = 0$
- $1 \times 2^2 = 4$
- $1 \times 2^3 = 8$

Now add them:

8 + 4 + 0 + 1 = 13

So, **1101 (binary) = 13 (decimal)**

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

6.What is ASCII?

Answer:
ASCII is a table that gives a **number to every character**.
Letters, digits, symbols, and even space — everything has its own number.
Computers use these numbers to understand text.

How does ASCII work?

- Computers cannot understand letters directly.
- So every character is stored as a **number**.
- These numbers come from the ASCII table.

Example:

- 'A' → 65
- 'a' → 97
- '0' → 48

7.What is Unicode?

Answer:
Unicode is a system that gives a **unique number to every character in every language**.
It supports English, Hindi, Chinese, emojis, symbols — everything.
It is bigger and more modern than ASCII.

8.What are Data Types in C++?

Answer:
Data types tell the computer **what kind of data** you want to store.
They define **size**, **type**, and **range** of a value.

Example:
Numbers, characters, decimal numbers, true/false — sabka alag type hota hai.

Main Data Types in C++

1. int

Used to store whole numbers (no decimals).
Example: 10, -5, 200

Size: Usually **4 bytes**

2. float

Used for decimal numbers with small precision.
Example: 3.14, 2.5

Size: **4 bytes**

3. double

Used for decimal numbers with **higher precision** than float.
Example: 3.141592

Size: **8 bytes**

4. char

Used to store a **single character**.
Example: 'A', 'b', '1', '@'

Size: **1 byte**

5. bool

Used to store **true** or **false** only.
Example: true, false

Size: **1 byte**

6. string

Used to store text.
Example: "Hello", "Vijay"

Size: Based on text length.

Real-Life Example

If you want to store:

- Age → **int**
- Price → **float** or **double**
- Single letter → **char**
- Yes/No → **bool**
- Name → **string**

9.Question

int = 4 bytes = 32 bits
Number = 10 → binary = **1010** (only 4 bits needed)

So remaining **28 bits are empty**.
Why waste? How to improve?

Answer

The computer does NOT store each number in different custom sizes.
It always stores an int in a **fixed size** (32 bits).
This is done for **speed**, **simplicity**, and **compatibility**.

Why 32 bits are used even if number is small?

1. Fast processing

CPU works fastest with fixed-size data (like 32-bit or 64-bit).
If every number had a different size, the CPU would become slow.

2. Easy memory alignment

Memory works in blocks (4 bytes, 8 bytes).
Using fixed size avoids complications.

3. Standardization

Same code should work on all machines.
That's why int = fixed size.