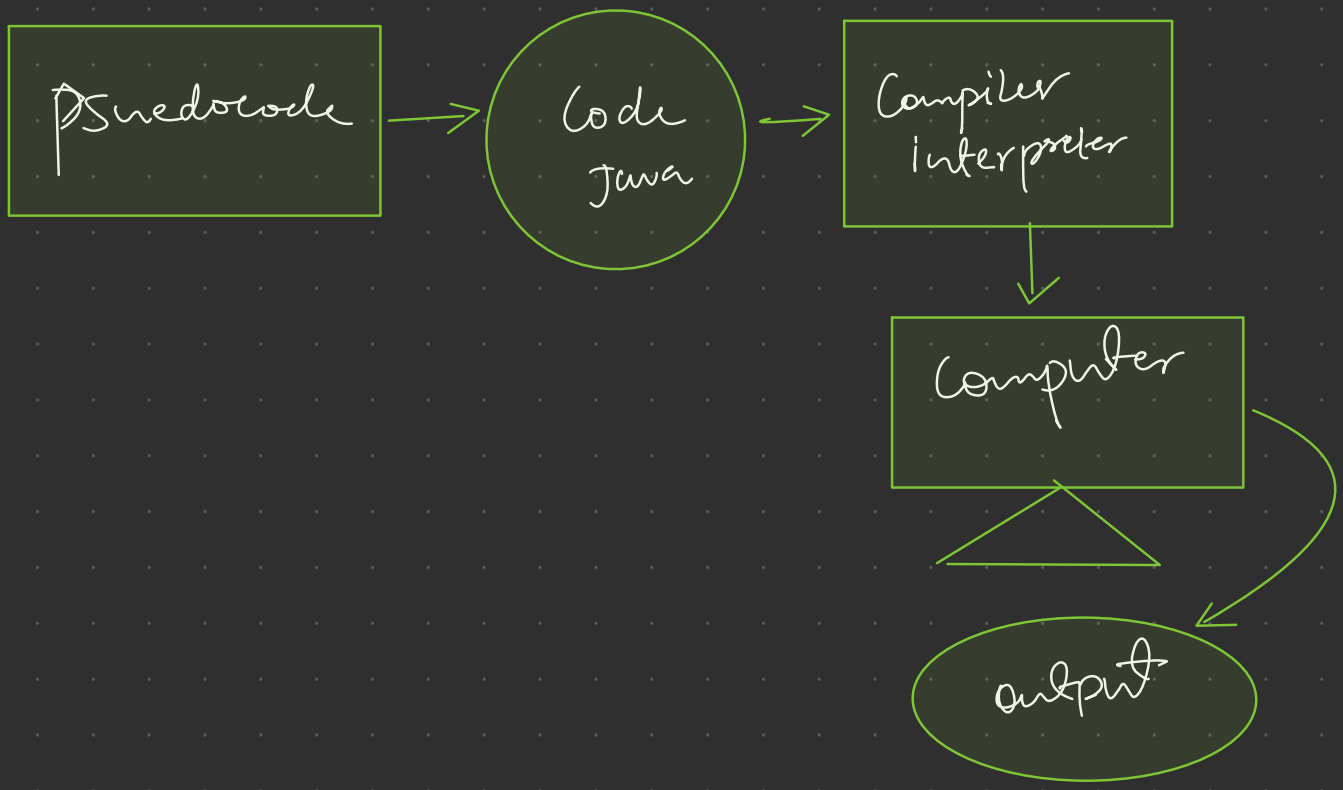
Lecture-3

→ Introduction



How to transistor store data in memory

0 0 1 0 1 → 1 Bit (2)

Store 5

Convert Binary 101

1 Bit = Binary
8 bit = 1 byte
(2^{10}) 1024 byte = 1 KB
(2^{10}) 1024 KB = 1 MB
(2^{10}) 1024 MB = 1 GB
(2^{10}) GB = 1 TB

ASCII TABLE

→ 25 → Convert Binary → Store Transistor

But what about characters like A, + -- etc how can store it

→ you store any characters. By ASCII Table

All world मानती है

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]

First code in Cpp

error

```
int main()
{
    std::cout<<"HelloG";

}
```

```
#include<iostream>
using namespace std;
int main()
{
std::cout<<"Hello G";
std::cout<<"Cool";
    return 0;
}
```

First code in Cpp on computer

Variable and data types

Coding World → Computer

1. Alphabet = char
2. number = int
3. words = String
4. yes/no = boolean

Real life how can talk

1. Alphabet → a, b, A, B ---
2. number → 0, 1, 2 ---
3. words → how are you?
4. gesture → yes or no

Variable = ek naam (label) jo ek memory location ko refer karta hai.

Data type = ek contract jo batata hai: how many bits, how to interpret them, kaunsi operations allowed.

Int data type.? long int - 8byte

```
int num = 10
```

num is a variable of type int

= → Assignment operator.

→ memory location Refer

int = 4byte (32bit) → Store

32bit → 00000000000000000000000010

10

Transistors

example of indigo sit

note .Data like 10 Pixel can not store in memory first Convert data in binary language / Machine language Core Computer only Understand machine langu

→ 4 byte (32bit) int fix to Compiler Read 32 no

Confision

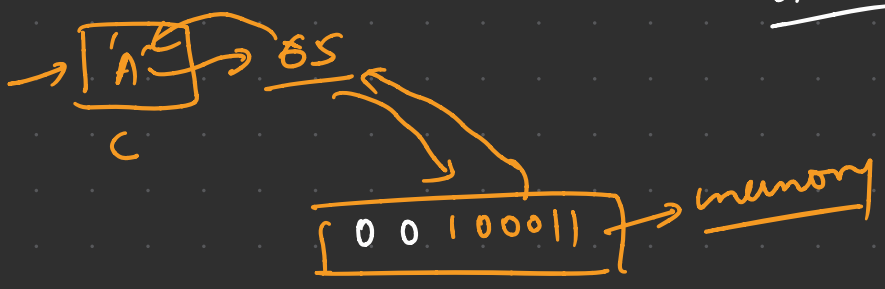
Char data type.?

1byte char c = 'A' ;

Variable naming Rule

not start × { A-Z, a-z ; 0-9 ; _ }

→ How To store in Memory ?



'A' → Convert to Ascii Value - 65 → Convert to binary 1000011

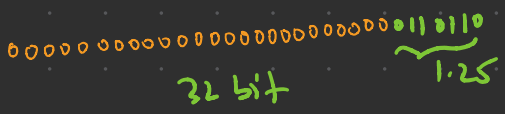
Compiler Read only 8 bit (1 char) in Memory

Float & double data type.?

4byte
8byte

float f = 1.25;

1.25 → Convert binary 01101010



Boolean data type.?

Boolean b = (0,1)

1byte

false true

00000000

How to use variable & data on computer .?

→ vs code

→ how To Store Negative Number or positive Number in Computer ?

⇨ what is 1's complement and 2's Complement.

why use ?

⚡ Sticky Note: Negative Numbers & Complements

Problem: Machine sirf 0/1 samajhti → negative numbers kaise store kare?

Sign-Magnitude: MSB = sign (0=+, 1=-), but 2 zeros (±0).

1's Complement: Flip bits for negative, still 2 zeros, extra carry.

2's Complement ✅ (used in C/C++/Java):

Negative = (1's complement + 1)

Only 1 zero, addition & subtraction same circuit → hardware simple.

Example (8-bit, -5):

+5 = 00000101
Flip = 11111010
+1 = 11111011 → -5

👉 Range (8-bit signed int): -128 ... +127

👉 MSB = Sign Bit (0=Positive, 1=Negative)

Conclusion

1. introduction
2. How transistor store data
3. ASCII table
4. First code in cop
5. Code on computer screen
6. Variable and data type
7. Int
8. Char
9. Float
10. Double
11. Boolean
12. How to use variable and data in code
13. Negative positive integer storage
14. 1's and 2's complement explained

Lecture-4 if else statement in Cpp

How to take input from user and how to use cin

🔷 How to take input from user in C++

1. First Thought Principle – Why Input Needed?

Computer ek stupid machine hai → apne aap kuch nahi kar sakta.

Agar hum bas code me fix values dalte rahenge, to har baar wahi result aayega.

Lekin real life me har baar input change hota hai (salary, marks, age).
👉 Isiliye programmer ko ek system chahiye jisme user apna data de sake aur program usko process kare.

2. C++ me Input lene ka syntax

C++ me input ke liye use hota hai cin (console input).

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

int main() {
    int age;
    cout << "Enter your age: ";   // prompt
    cin >> age;                  // input from user
    cout << "Your age is " << age;
    return 0;
}
```

3. How cin Works Internally?

cin ek object hai jo input stream ko represent karta hai.

Input stream = data flow keyboard → program.

Jab user kuch type karta hai aur Enter dabata hai:

OS usko ek buffer me store karta hai.

cin us buffer se data uthata hai aur variable me daal deta hai.

👉 Example:
User types 25 → OS buffer → cin reads → age variable me store ho gaya.

4. Why >> (extraction operator)?

>> ka matlab hai "nikalna" (extract karna).

cin >> age; → matlab input stream se ek integer nikaal kar age me daal do.

Agar aap multiple inputs lena chahte ho:

```
int a, b;
cin >> a >> b;   // space/enter se separate
```

🔷 Compiler Working (Step by Step)

1. First Thought Principle – Why Compiler?

Human likhta hai → English-like code (cout << "Hello")

Computer samajhta hai → Machine code (0 aur 1).

To hume ek translator chahiye → jisse hum samajhne layak code likhe aur machine usko 0/1 me samjhe.
👉 Yehi translator hai Compiler.

Example Code

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

int main()
{
    ===    } cin>> code
}
```

Operator in Cpp

① Assignment operator = a = 5

② Comparision operator ⇒ == a==6

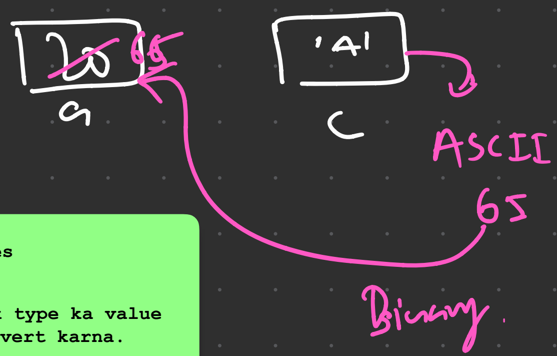
5==6 no

5==5 yes

Concept of type casting

```
int a = 20;
char c = 'A';
```

a = c



Data loss in type casting

🔷 Quick Sticky Notes

✅ Type Casting = ek type ka value ko dusre type me convert karna.

✅ Implicit → automatic safe conversion (int → float).

✅ Explicit → forced conversion (float → int) → data loss possible.

✅ Data Loss Types:

Fractional part truncate (3.14 → 3).

Precision loss (big int/double → float).

Overflow (large int → short).

type casting Hand on

If else condition start

Salary example

```
int package = 200;
if ( package > 180 ) ⟶ 200 > 180 yes → (1)
    cout << " Offer Accepted";
else
    cout << " offer Rejected";
}
```

150 > 180 No (0)

note If-else Rule { } if statement no need to use

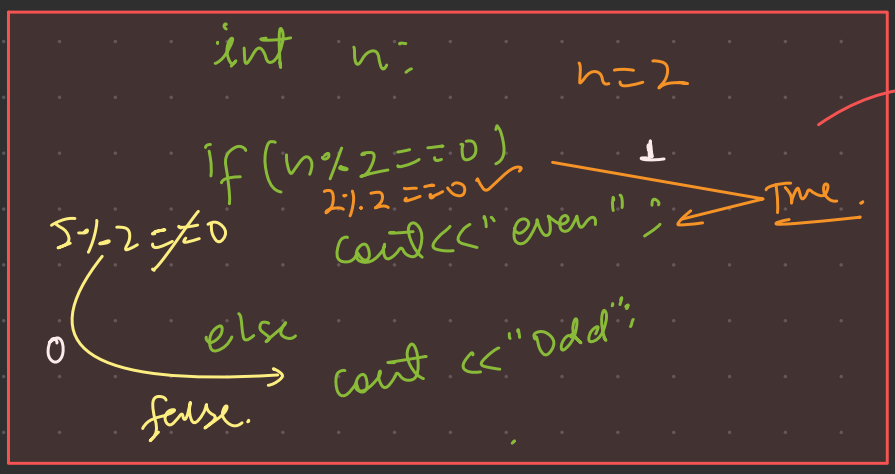
Compare two variable

```
int a, b;
if ( a > b)
    cout << "yes";
else
    cout << "No";
```

True: 20 > 10 ✓ (1) → yes

False: 10 > 20 ✗ (0) → No

Check number even or odd



Code in VS Code

Homework Time

(1) check, a human Adult or Teenager.

```
age=18
if (age>18)
    count<<Adult.
else
    count<<Teenegar.;
```

Code Vs Code

Q ② homework Voters. eligibility

If else if else

Q. check Number is positive, negative, Zero

Way-1

- num > 0 +ve
- num < 0 -ve
- num == 0 zero

```
if(num>0)
    positive
else
    if(num==0)
        zero
    else
        negative
```

```
if(num>0)
    cout<<positive;
else if(num==0)
    cout<<"zero";
else
    cout<<"negative";
```

Better.

Q. check character. Vowel or not

Vowels = a e i o u , Consonant - - -

```
char ch;
if(ch=='a')
    Vowel;
else if(ch=='e')
    vowel
else if(ch=='i')
    vowel
else if(ch=='o')
    vowel
else if(ch=='u')
    Vowel
else
    cout<<Consonant;
```

Q. Print weekdays with Number:

Logic
1 : Monday
2 : Tue
3 : web
4 : Thu
5 : fri
6 : Sat
7 : Sunday

```
int n; n=1
if(n==1)
    cout<<Mon;
else if(n==2)
    cout<<Tue
else if(n==3)
    cout<<"wed";
else if (n==4)
    Cout<<Thu
else if (n==5)
    cout<<fri;
else if (n==6)
    cout<<Sat;
else if (n==7)
    cout<<Sund;
else
    cout<<enter valid no;
```

Loops Concept Introduction

→ Loops Work on pattern

Let Say we want to print Coder Army 5 times.

```
count = 1 , count <= 5
    — print ("Coder Army")
count = count + 1;
```

→ Solve By Loops (for)

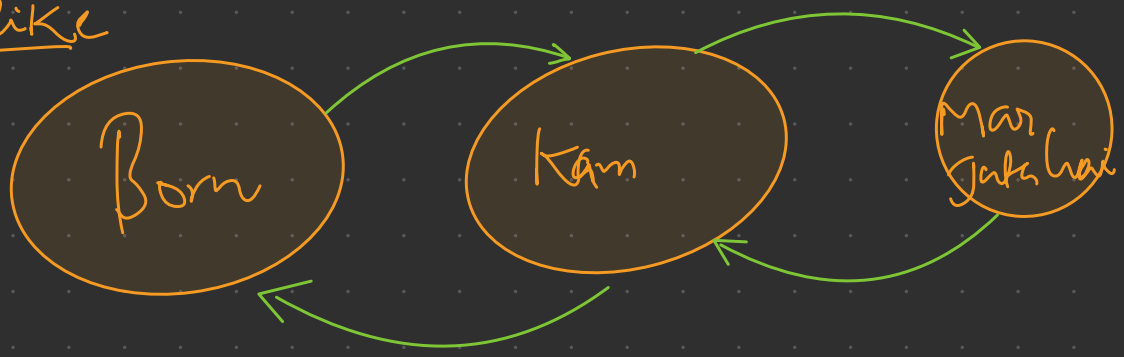
Now want to print 1000 ?

- Coder Army - 1
- Coder Army - 2
- Coder Army - 3
- Coder Army - 4
- Coder Army - 5

Syntax of for Loop

```
for(count=1; count<=5; count=count+1) {
    Count<<"Coder Army";
}
```

Loops like



उतरे काम करते पर स्वर्ग करना पर्व

Working of for loop

print hello 10 Times

```
for ( int i=0;  i<10;  i++)
{
    cout <<"hello";
}
```

1 → i<10 → 2 → cout <<"hello" → 3 → i++ → i<10 → ...

Looping Until True Condition

⭐ print Sanjay 20 time

```
for(i=1  i<=20; i++)
{
    Sanjay
}
```

→

```
for(i=10; i<=30; i++)
{
    Sanjay
}
```

⭐ print 1 to 5 ? find pattern

Count = 1 2 3 4 5

Count<=5.

Count = Count + 1 ✗ 3

Count=1 Count<=5.

1	1<=5 T
2	2<=5 T
3	3<=5 T
4	4<=5 T
5, 6	5<=5 T 6<=5 F

1
2
3
4 5

1 +1
2 +1
3 +1
4 +1
patten. 5

Solve for loop

```
for(int Count=1; Count<=5; Count=Count+1){
    cout<<Count;
}
```

☆ print Squar

1 to n → 5

n=5

logic

1 = 1
2 = 2×2 = 4
3 = 3×3 = 9
4 = 4×4 = 16
5 = 5×5 = 25

```
for(i=1; i<=5; i++){
    Count<<i*i;
}
```

Code

i=1	i<=5	i++	1
i=2	1<=5T		4
i=3	2<=5T		9
i=4	3<=5T		16
i=5	4<=5T		25
i=6	5<=5T		
	6<=5F		

☆ print all even number within given number.

2, 4, 6, 8, 9, 10 ---

2 2 2 2

pattern

```
for(int i=2; i<=n; i=i+2)
{
    cout<<i<<endl;
}
```

```
loo(——){
  if(i%2==0)
      even
  else
      odd
}
```

→ Conclusion lecture-4

1. How to take input from user & how to use cin?
2. Take user input to sum two numbers
3. Operator Cpp
4. Typecasting concept
5. Data loss concept
6. Type casting hand on
7. If else condition start
8. Ef else example with salary package
9. Marks grading system
10. Hands on marks grading system
11. If else rule
12. Compare two variable
13. Hands on
14. Check number even or odd
15. Homework voter eligibility
16. Number positive negative or zero
17. Hands on
18. Character is vowel or not
19. Print weekday with number
20. Loop concept introduction
21. Syntax of for loop
22. Explain the working of for loop
23. Print number 1 to 5
24. Print square 1 to n
25. Print coder army 10 times
26. Print n natural number
27. Hands on print square print all even numbers up to 20
28. Code print even number up to 20