

Lecture - 2

Introduction to C++ and Write first code

Computer Memory units:

Smallest unit of Memory : 1 bit
(1 transistor can represent 1 bit)

	1 Byte	=	8 bit
(Kilo Byte)	1 KB	=	2^{10} byte
(Mege Byte)	1 MB	=	2^{20} byte = 2^{10} KB
(Giga Byte)	1 GB	=	2^{30} byte = 2^{10} MB
(Tera Byte)	1 TB	=	2^{40} byte = 2^{10} GB

Computer only understand Binary number. 0 & 1.

ASCII TABLE:

Every symbol, sign present in keyboard have Unique ASCII Value.

A → 65 → 1000001

B → 66 → 1000010

0 → 48 → 00110000

a → 97 → 01100001

z → 122 → 01111010

Introduction to C++ :

C++ is extension of C language.

* Simple code to print a message:

```
int main() { → Tells from where it start reading
    cout << "Hello CoderArmy ";
    return 0; → tell end of program.
} → tells where it can stop reading
```

cout → print output

↓

character output / Console output

<< → insertion operator

→ #include <iostream> → header file, library where meaning
using namespace std; is defined.

```
int main() {
    cout << "Hello CoderArmy ";
    return 0;
}
```

→ If we don't put "using namespace std;" at
starting then we have to write at every
statement when we use cout.

std::cout << "Hello CoderArmy";


```
#include <iostream>
```

Cont << a << b << c;

✱

① int \rightarrow Example: 4, 69, 0, -9, etc.

for storing a integer value it takes 4 bytes.

Ex: 10 \rightarrow 1010

In int

[illegible]

↳ 10 in Binary number.

```
int a = 10;
```

↓ ↓ ↘

datatype variable value

Rules for Writing a variable name:-

- Name your variable is related to your code
- Variable name always start with alphabet or underscore.
- Variable name consist of number, alphabet or underscore.
- Keyword are not put as variable name.

Keywords:

The keywords are reserved and the meaning are fixed.

asm	double	new	switch
auto	else	operator	template
break	enum	private	this
case	extern	protected	throw
catch	float	public	try
char	for	register	typedef
class	friend	return	union
const	goto	short	unsigned
continue	if	signed	virtual
default	inline	sizeof	void
delete	int	static	volatile
do	long	struct	while

② char

Ex: `char c = 'b';`

↳ represent within single quote

↳ single value stored inside the quote.

Size : 1 byte.


```
→ int main() {
    int a = 10;
    cout << a;    → It prints 10.
}
```

```
→ int main() {
    int a = 10;
    char c = 'z';
    cout << "a";    → prints a
    cout << a;      → prints 10
    cout << c;      → prints z
    return 0;
}
```

③ float → decimal value

Ex: float a = 9.62;
float height = 5.9;

Size : 4 byte.

④ Double → big decimal value

Size : 8 byte

Ex: 634.982398648326

→ This can't be stored in float.

Size : 8 byte.

⑤ long int :

ex: long int a = 56789234;

for storing large value.

Which can't be stored in int.

Size = 8 byte.

```
→ int main() {
    float num = 23.45;
    cout << num << endl;
    double num1 = 29.685674;
    cout << num1 << endl;
    return 0;
}
```

* Comment :

Single line

// This is single line comment

Multi line

/* This is

Multi line

comment */

→ How to store Negative Number:

Num = 5

↳ 101

for -5

find 2's Complement of 5

5 \rightarrow 101

1's Complement (010)

2's Complement (011)

2's Complement : 011

~~Negative~~ \leftarrow 0 (_ _ _ _) positive
~~positive~~ \leftarrow 1 (_ _ _ _) Negative

for 3 space (bit)

— — —

Total possible \rightarrow 8

positive = 3

Negative = 4

zero = 1

for 32 bit

Total possible $\rightarrow 2^{32}$

positive = $2^{31} - 1$

Negative = 2^{31}

zero = 1

⑥ bool

ex : bool isprime = true;

bool isodd = 0;

Size = 1 byte

0 or 1

true or false.

```

→ int main() {
    bool num = true;
    cout << num << endl;
    return 0;
}

```

→ To find the size of any datatype:

Code:

```

int main() {
    int number = 10;
    cout << sizeof(number) << endl;    return → 4
}

```

```

float num = 10.99;
cout << sizeof(num) << endl;    return → 8

```

* Typecasting:

```

int main() {
    char c = '8';
    cout << c << endl;    → output = 8
}

```

```

char a = 65;
cout << a << endl;    → output = A
return 0;
}

```

→ Typecasting

char a = 65;

→ We take char datatype and store integer

So Computer Convert integer value to Binary number

$$65 = 1000001$$

then it is Converted and checked in the ASCII table and print the character associate With it.

$$1000001 \rightarrow A$$

→

```
int main() {  
    int c = 'A';  
    cout << c << endl;  
    return 0;  
}
```

 → Converted into Binary and then check in ASCII And return the integer Value associated With it.

Output = 65

→

```
int main() {  
    char c = 590;  
    cout << c << endl;  
    return 0;  
}
```

 Output = N.
↓
Gives a Warning Statement
on character We can store only 2^8 value.

Maximum We can store $2^8 = 256$ value.
0 to 255.

But we give as input

590

So, it take Modulo With Maximum storage capacity

$$590 \% 256$$

↳ (78)

↳ It give output which character is associated with Binary representation of 78.

64 32 16 8 4 2 1

78 → 01001110

↳ It give 'N'

↓

output of previous code.

* Conditional Statement : (If - else condition)

Find greatest among two number.

a = 10;

b = 15

```
int main() {
```

```
    int a = 10, b = 15;
```

```
    if (a > b)
```

// Output : 15

```
        cout << a;
```

```
    else
```

```
        cout << b;
```

```
    return 0;
```

```
}
```


Operator :

Multiplication : $2 * 3 = 6$

Division : $8 / 2 = 4$

Modulo : $9 \% 2 = 1$

Comparison Operator :

$<$ \rightarrow less than

$>$ \rightarrow greater than

$<=$ \rightarrow less than or equal to

$>=$ \rightarrow greater than or equal to

$!=$ \rightarrow Not equal to

\rightarrow Check given number is even or odd.

\rightarrow even : 0, 2, 4, 6, - - -

odd : 1, 3, 5, 7, - - -

Number $\% 2 = 0$ (Even)

Number $\% 2 = 1$ (Odd)

Code :

```
int main() {
```

```
    int num = 25;
```

```
    if (num % 2 == 1)
```

```
        cout << "ODD";
```

```
    else
```

```
        cout << "EVEN";
```

```
    return 0;
```

```
}
```

// output

: ODD

* For multiple condition:

```
if ( ) {  
  
}  
else if ( ) {  
  
}  
else if ( ) {  
  
}  
else {  
  
}
```

→ find given number is +ve, -ve or zero.

```
int main() {  
    int number = 16;  
    if (number > 0)  
        cout << "positive";  
    else if (number < 0)  
        cout << "negative";  
    else {  
        cout << "zero";  
    }  
    return 0;  
}
```

Output
positive.

→

&& → AND operator

|| → OR operator

Given 3 number, print Maximum

```
int main() {
```

```
    int a = 10, b = 15, c = 20;
```

```
    if (a >= b && a >= c)
```

```
        cout << a;
```

```
    else if (b >= a && b >= c)
```

```
        cout << b;
```

```
    else
```

```
        cout << c;
```

```
    return 0;
```

```
}
```

Output

20

0 AND 0 → 0

0 AND 1 → 0

1 AND 0 → 0

1 AND 1 → 1

0 OR 0 → 0

0 OR 1 → 1

1 OR 0 → 1

1 OR 1 → 1