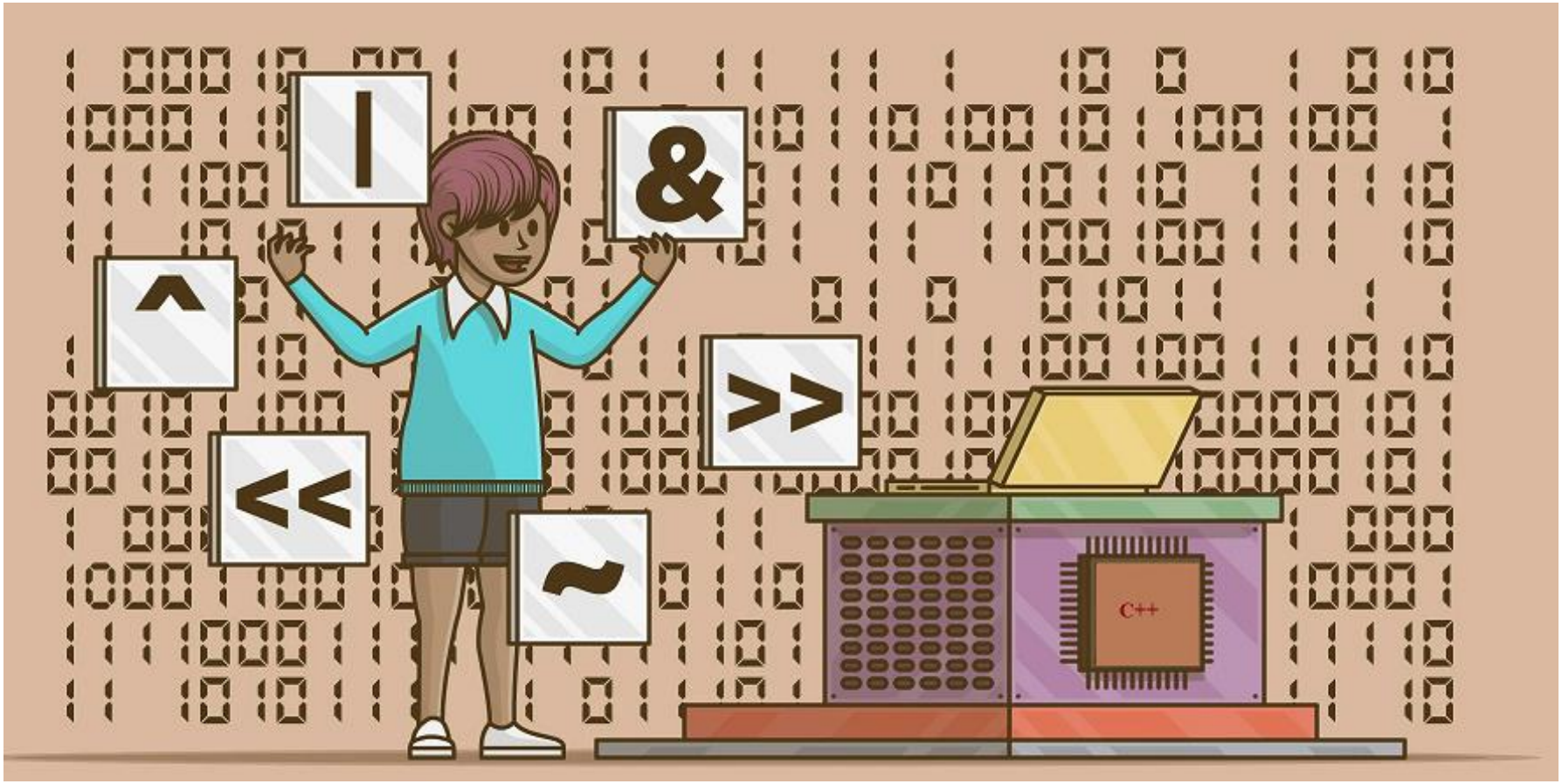
CAP444
OBJECT ORIENTED PROGRAMMING USING C++

Session #2

Created By:
Kumar Vishal
(SCA), LPU

Today we are going to learn about……?



Bitwise operators

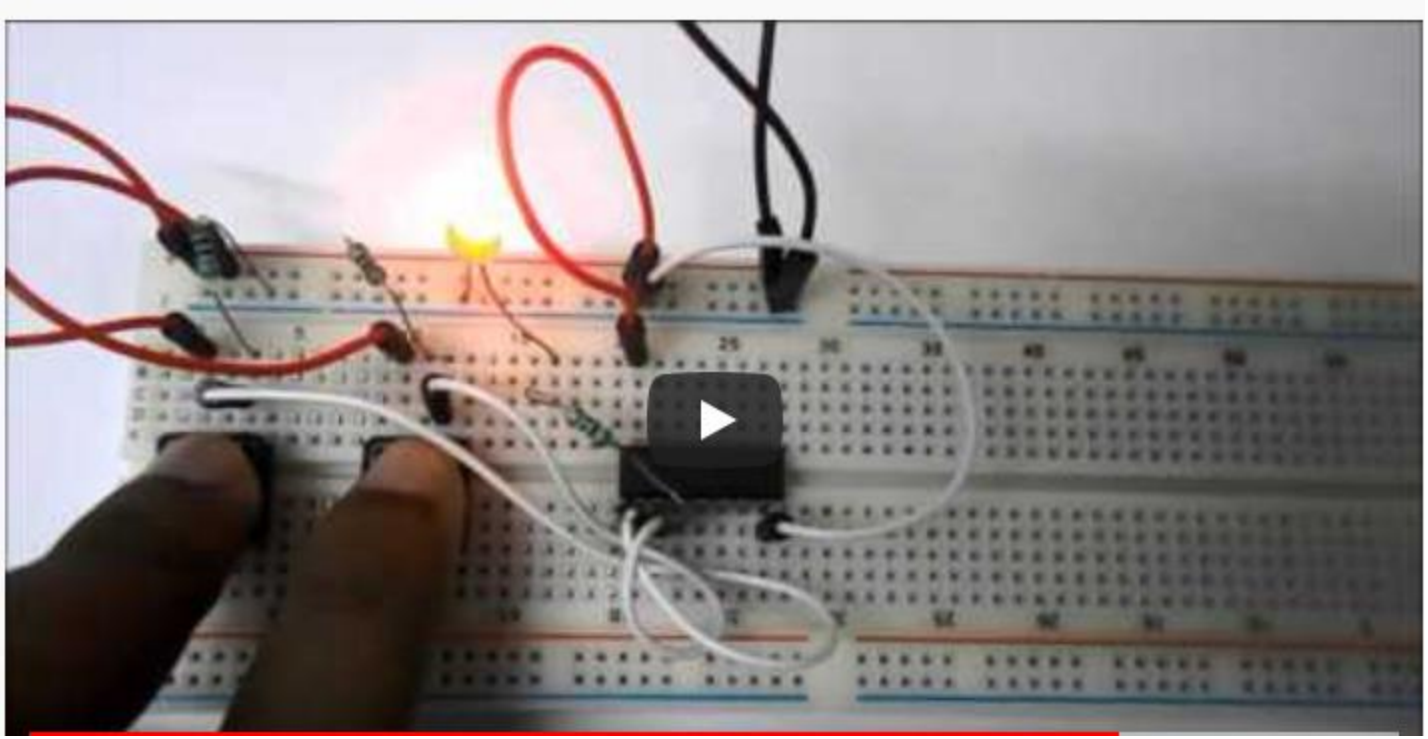
Bitwise operators

Operator	Description
&	AND Operator
|	OR Operator
^	XOR Operator
~	Ones Complement Operator
<<	Left Shift Operator
>>	Right Shift Operator

AND Operator (&)

If both side bit is on result will be On

a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1



https://www.youtube.com/watch?v=rKt1LXkcrwE&t=15s

Steps to solve:-

- **a = 12 (find binary form:1100)**
- **b = 25 (find binary form:11001)**

How to find Binary:

64	32	16	8	4	2	1	
		0	1	1	0	0	12
		1	1	0	0	1	25
			1	0	0	0	8

a & b=

01100 (12)

11001 (25)

01000 (8) Ans.

What will be output?

```
#include <iostream>

using namespace std;

int main()
{
    int a=20;
    int b=25;
    cout<<(a&b);
     return 0;
}
```

A. 15
B. 16
C. 20

OR Operator (|)

If any side bit is on result will be On

a	b	a | b
0	0	0
0	1	1
1	0	1
1	1	1

Steps to solve:-

- **a = 12 (find binary form:1100)**
- **b = 25 (find binary form:11001)**

How to find Binary:

64	32	16	8	4	2	1	
		0	1	1	0	0	12
		1	1	0	0	1	25
		1	1	1	0	1	29

a | b=

01100 (12)

11001 (25)

11101 (29) Ans.

What will be output?

```
#include <iostream>

using namespace std;

int main()
{
    int a=20;
    int b=15;
    cout<<(a|b);
    return 0;
}
```

A. 31
B. 32
C. 22
D. 32

XOR Operator (^)

If both side bit is opposite result will be On

a	b	a ^ b
0	0	0
0	1	1
1	0	1
1	1	0

Steps to solve:-

- **a = 12 (find binary form:1100)**
- **b = 25 (find binary form:11001)**

How to find Binary:

64	32	16	8	4	2	1	
		0	1	1	0	0	12
		1	1	0	0	1	25
		1	0	1	0	1	21

a ^ b=

01100 (12)

11001 (25)

10101 (21) Ans.

Left Shift Operator(<<)

a=10 (1010)

a<<1

10100

10100(20) Ans.

a<<2

101000

101000(40) Ans.

Right Shift Operator(>>)

a=10 (1010)

a>>1

101

101(5) Ans.

a>>2

10

10(2) Ans.

What will be output?

```
#include <iostream>
using namespace std;
int main()
{
  int a=15;
  cout<<(a>>1);
return 0;
}
```

Options:

A. 5

B. 6

C. 7

D. 8

Increment/Decrement Operator

++: Increment
++x
--: Decrement
--x

```
int main()
{
    int a=10;
    a++;
    cout<<a;
    return 0;
}
```


LOVELY PROFESSIONAL UNIVERSITY
PUNJAB (INDIA)
L
P
U

What will be output?

```
#include <iostream>

using namespace std;

int main()
{
   int a=10;
   int c=a++;
   cout<<c;

   return 0;
}
```

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

int main()
{
   int a=10;
   int c=++a;
   cout<<c;

   return 0;
}
```

What will be output?

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

 int main()
 {
    int x = 5, y = 5, z;
    x = ++x; y = --y;
    z = x++ + y--;
    cout << z;
    return 0;

 }
```

insertion operator(<<):

The cout is used in conjunction with stream insertion operator (<<) to display the output on a console

extraction operator (>>):

The cin is used in conjunction with stream extraction operator (>>) to read the input from a console.

Control structure

- Conditional structure: if and else
- Selective structure: switch case
- Iteration structures (loops): while, do while, for
- Jump statements: break, continue, goto

While loop

The syntax of a while loop in C++ is −

```
while(condition)
 {
   statement(s);
}
```

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

int main ()
 {
     int a = 10;
     while( a < 20 )
     {
     cout<< a << endl;
     a++;
       }
   return 0;
}
```

Do While loop: at least one time will be execute

The syntax of a do while loop in C++ is −

```
do {
   statement(s);
}
while( condition );
```

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

int main ()
 {
     int a = 10;
  do
     {
      cout<< a << endl;
     a++;
    } while( a > 20 );
   return 0;
}
```

For loop:

The syntax of a for loop in C++ is –

```
for ( initialization; condition; increment )
{
   statement(s);
}
```

Any Query?