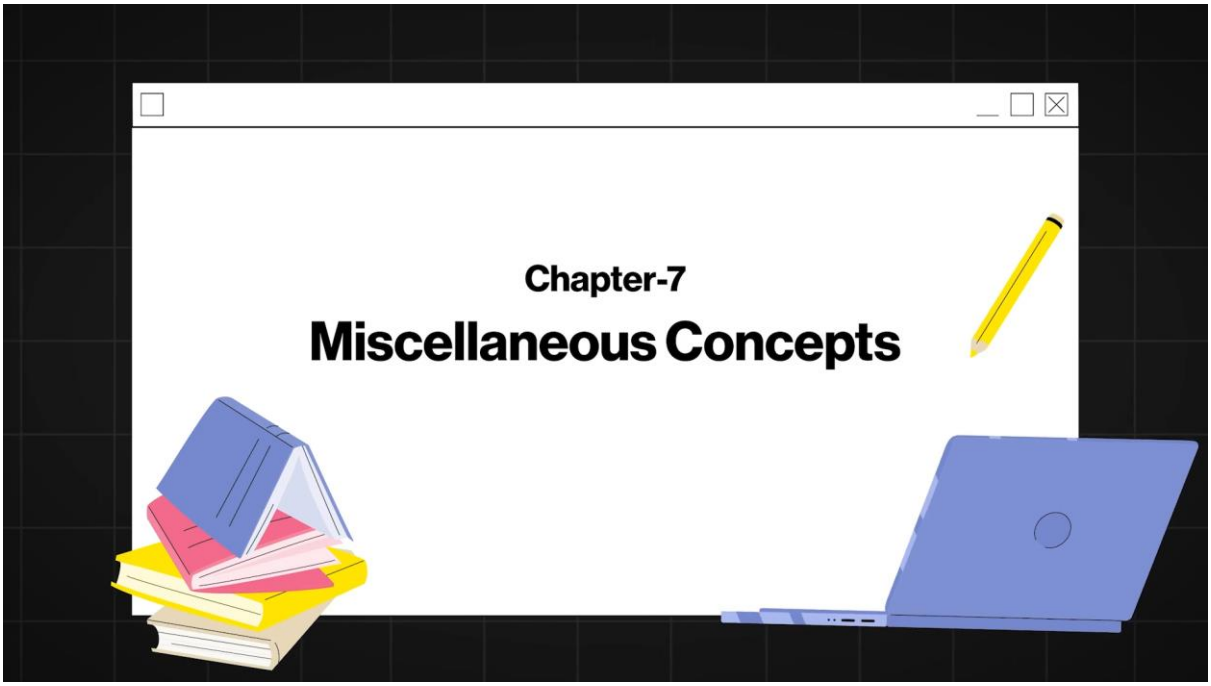
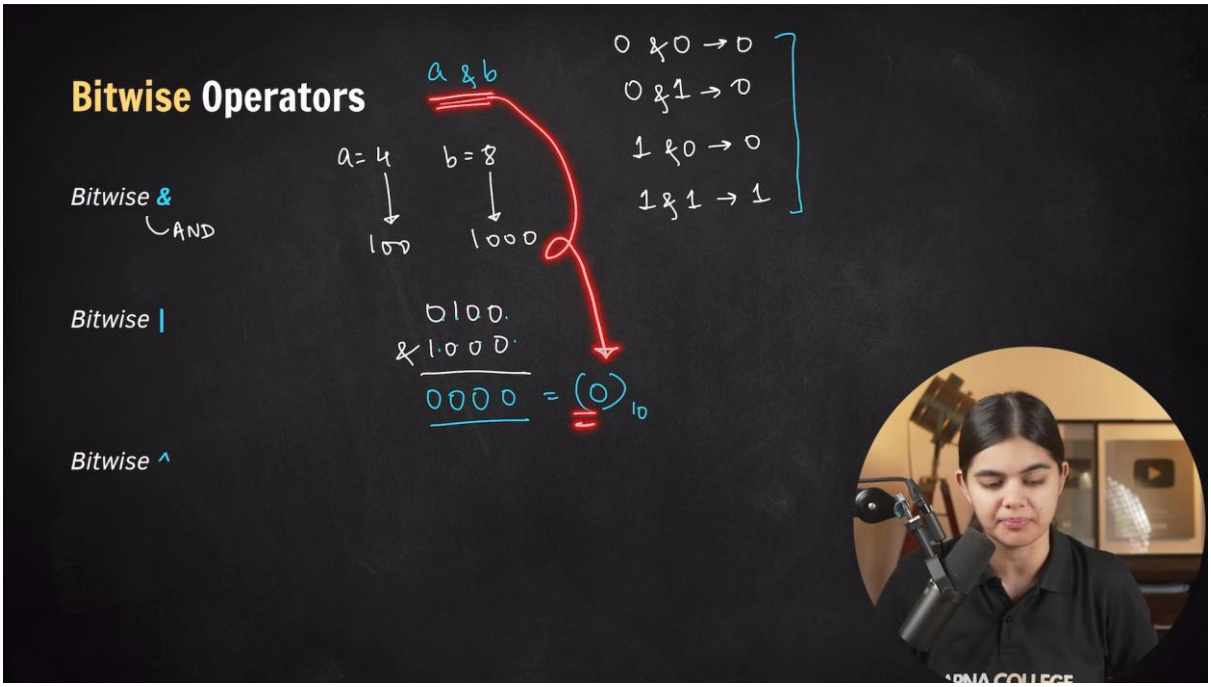
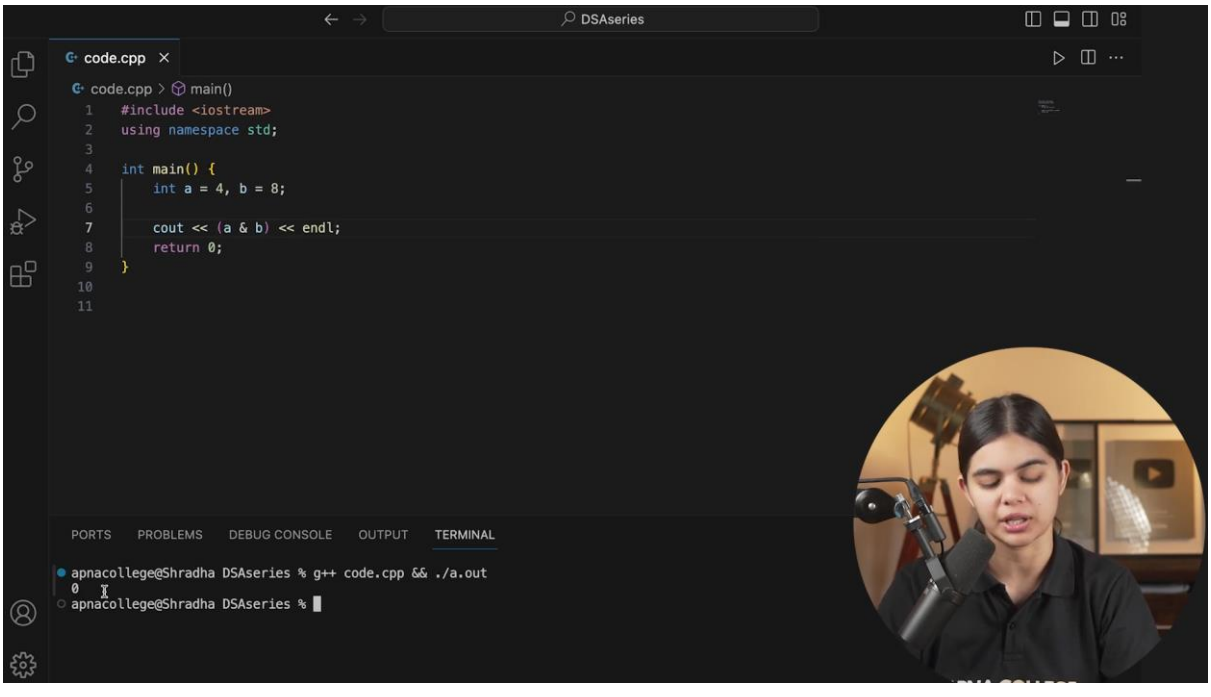

Chapter-7
Miscellaneous Concepts


Bitwise Operators
Bitwise &
AND
Bitwise |
Bitwise ^
a & b
a = 4
b = 8
100
1000
0100
&1000
0000 = (0)10
0 & 0 → 0
0 & 1 → 0
1 & 0 → 0
1 & 1 → 1


DSAseries
code.cpp
code.cpp > main()
#include <iostream>
using namespace std;
int main() {
int a = 4, b = 8;
cout << (a & b) << endl;
return 0;
}
PORTS
PROBLEMS
DEBUG CONSOLE
OUTPUT
TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
0
apnacollege@Shradha DSAseries %

Bitwise Operators

0 | 0 → 0
0 | 1 → 1
1 | 0 → 1
1 | 1 → 1

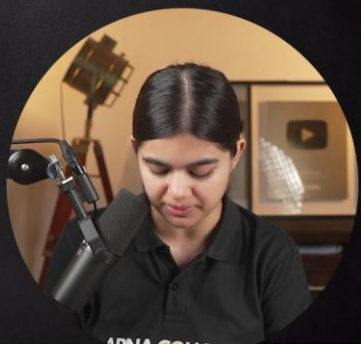
Bitwise &
AND

Bitwise |

0100
| 1000
1100 = $(12)_{10}$

2^3 2^2 2^1 2^0

Bitwise ^



DSAseries

code.cpp

code.cpp > main()

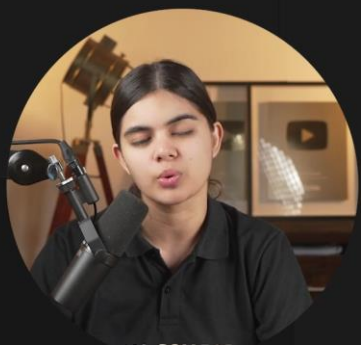
```cpp
#include <iostream>
using namespace std;

int main() {
    int a = 4, b = 8;

    cout << (a | b) << endl;
    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
0
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
12
apnacollege@Shradha DSAseries %
```



Bitwise Operators

same → 0
dif → 1

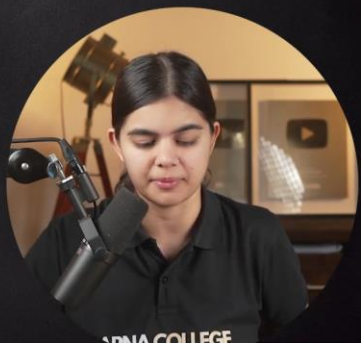
0^0 = 0 1^1 = 0
0^1 = 1 1^0 = 1

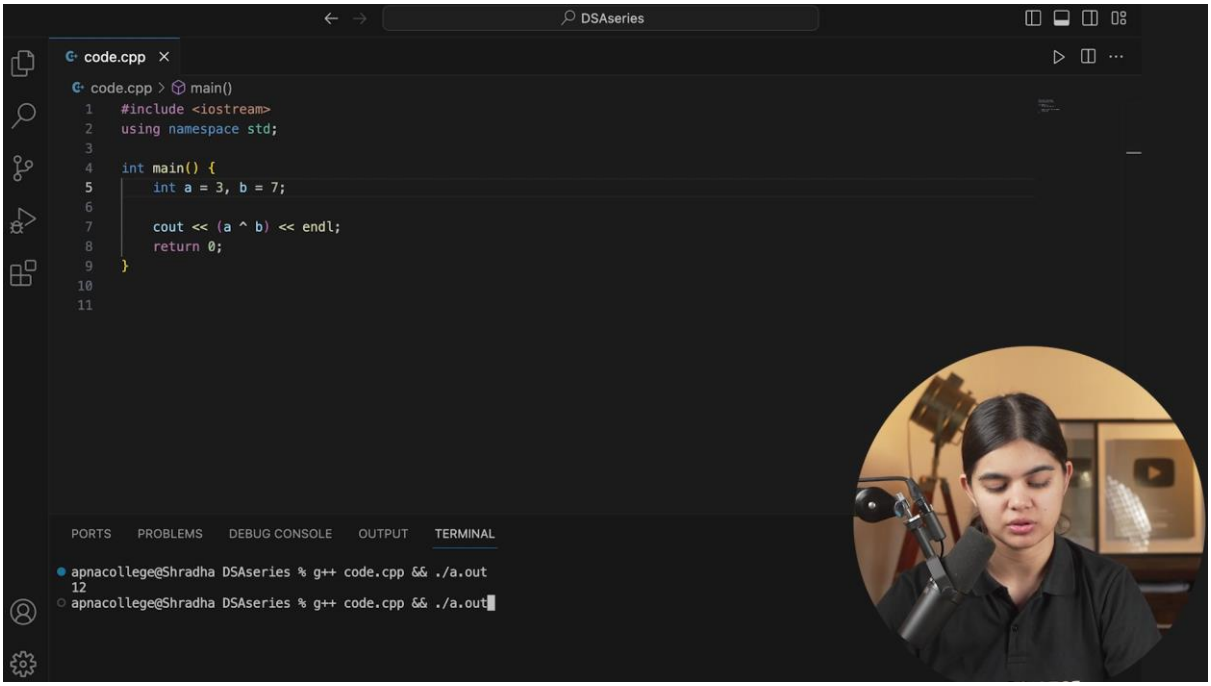
Bitwise &
AND

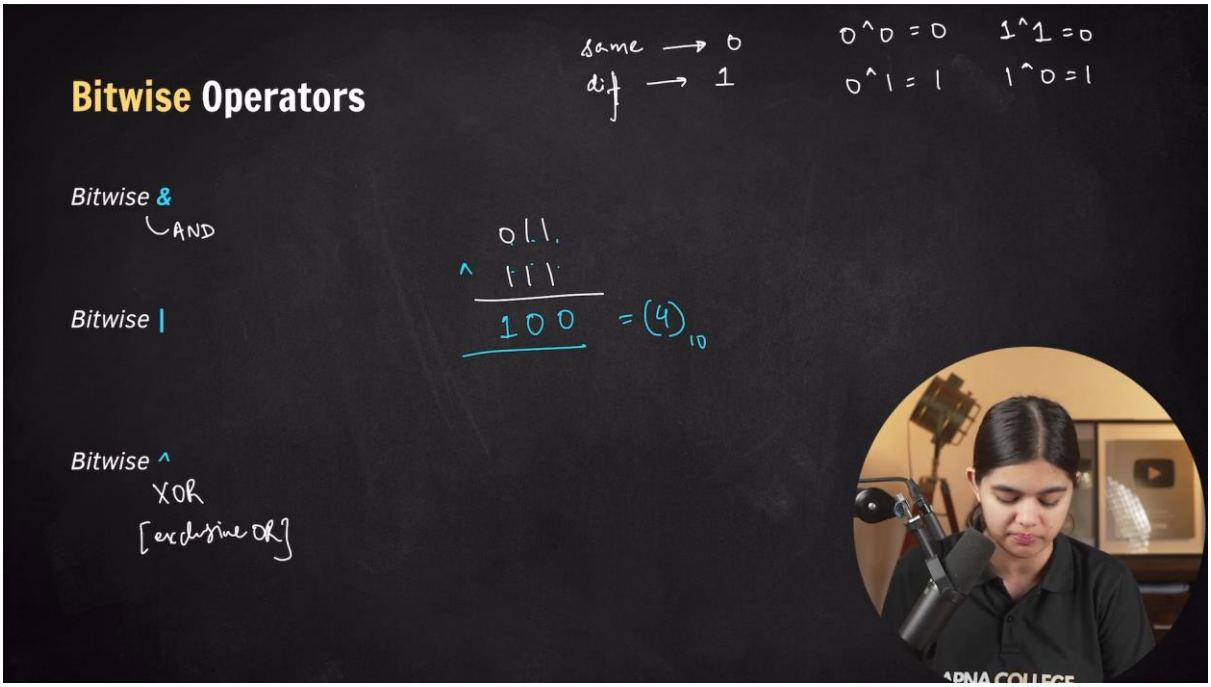
0100
^ 1000
1100 = $(12)_{10}$

Bitwise |

Bitwise ^
XOR
[exclusive OR]




DSAseries
code.cpp
code.cpp > main()
#include <iostream>
using namespace std;
int main() {
int a = 3, b = 7;
cout << (a ^ b) << endl;
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
12
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out


Bitwise Operators
same → 0
dif → 1
0^0 = 0
1^1 = 0
0^1 = 1
1^0 = 1
Bitwise &
AND
011
^ 111
100 = (4)10
Bitwise |
Bitwise ^
XOR
[exclusive OR]

Bitwise Operators

$n = 4$

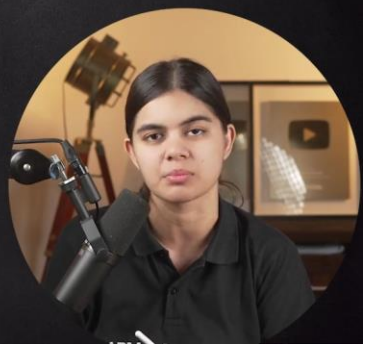
$n << i$ $\quad$ $n << 1$

Bitwise **<<**

Left shift op.

Bitwise **>>**

Right shift op



DSAseries

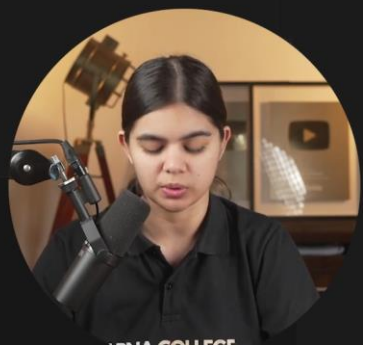
code.cpp

code.cpp > main()

```
1   #include <iostream>
2   using namespace std;
3
4   int main() {
5
6       cout << (4 << 1) << endl;
7       return 0;
8   }
9
10
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
8
apnacollege@Shradha DSAseries %
```



Bitwise Operators

$10 << 2$

1 0 1 0

[1 0 1 0 0 0]

32 16 8 4 2 1

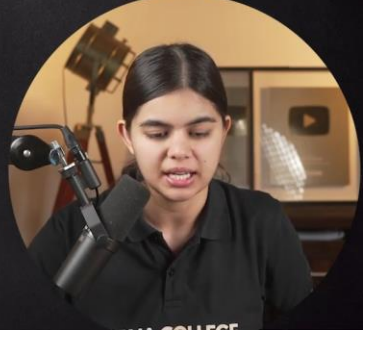
$32 + 8 = (40)_{10}$ ✓

Bitwise **<<**

Left shift op.

Bitwise **>>**

Right shift op



Bitwise Operators

Bitwise <<
Left shift op.

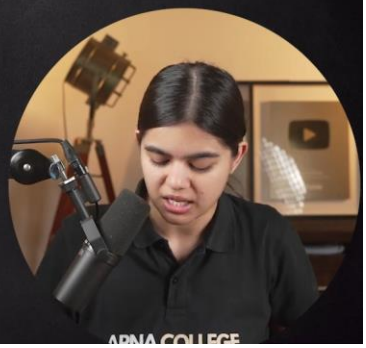
Bitwise >>
Right shift op

10 >> 1

1010

[0 1 0 1] 0

$(101)_2 = (5)_{10}$



Bitwise Operators

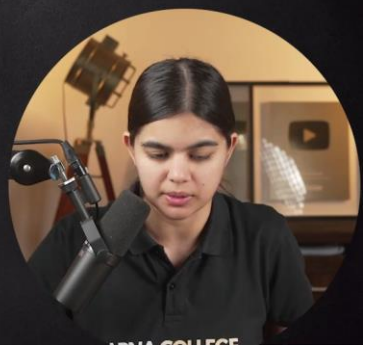
Bitwise <<
Left shift op.

Bitwise >>
Right shift op

8 >> 2

1000

0 0 1 0 $= (2)_{10}$



Bitwise Operators

Bitwise <<
Left shift op.

Bitwise >>
Right shift op

a << b

ans = $a * 2^b$

$= 8 * 2^1 = 16$

8 << 1
a b

1000

(10000) $= (16)_{10}$
2

a >> b

ans = $a / 2^b$

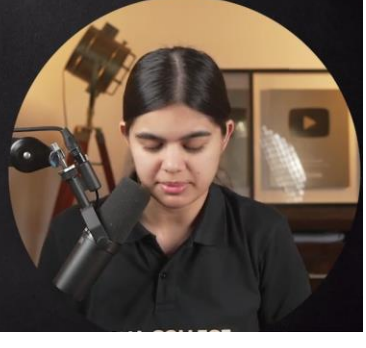
$= 8 / 2^1$

$= 4$

8 >> 1
a b

1000

$(100) = (4)_{10}$



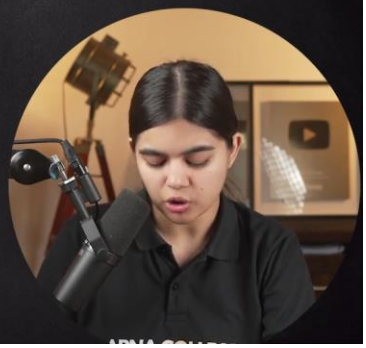
Homework

Solve for :

- *6 & 10*
- *6 | 10*
- *6 ^ 10*

Solve for :

- *10 << 2*
- *10 >> 1*



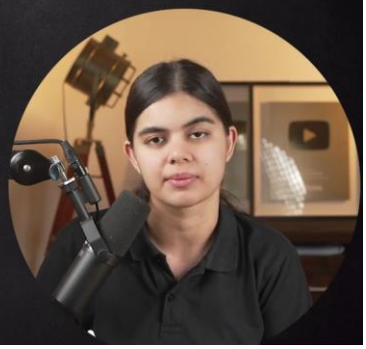
Operator Precendence

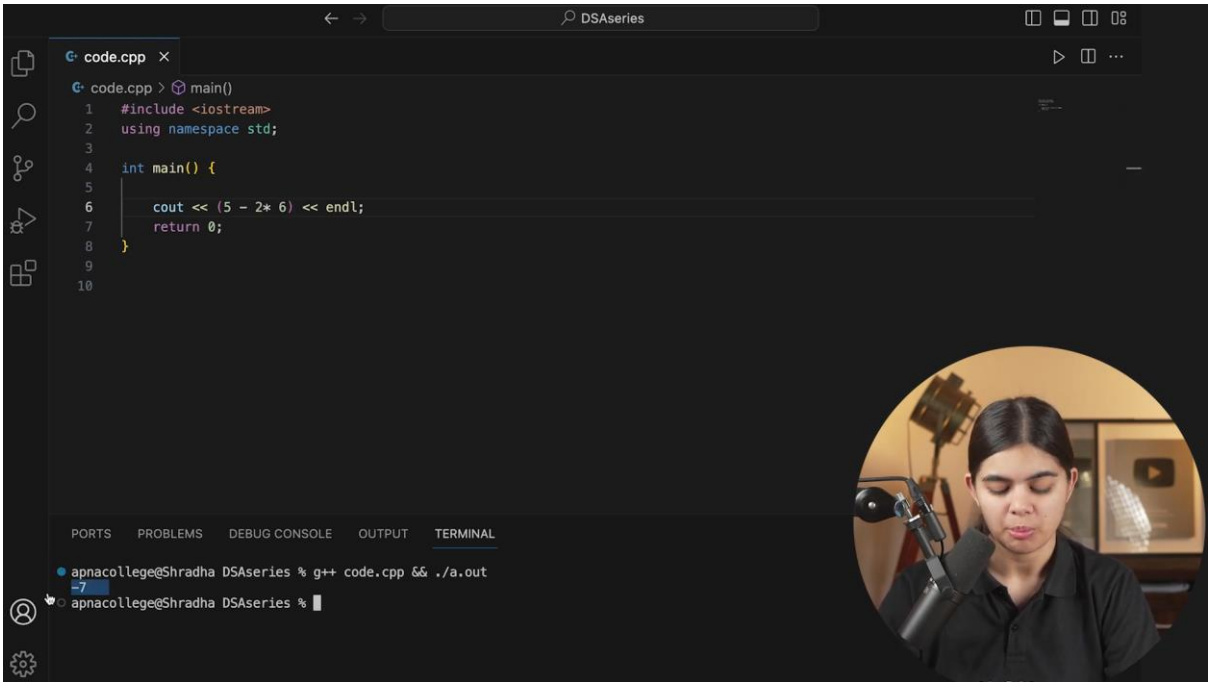
ans = 5 − 2 * 6 = 5 − (2*6)

= 5 − 12 = −7

Operators	Precedence	
!, +, − (unary operators)	first	R to L
*, /, %	second	L to R
+, −	third	L to R
<, <=, >=, >	fourth	L to R
==, !=	fifth	L to R
&&	sixth	L to R
||	seventh	L to R
= (assignment operator)	last	R to L

Bitwise operators





Operator Precedence

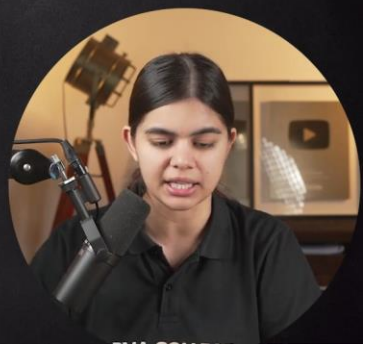
ans = 5 - 2 * 6

(5 - 2) * 6

3 * 6 = 18

Operators	Precedence	
!, +, – (unary operators)	first	R to L
*, /, %	second	L to R
+, –	third	L to R
<, <=, >=, >	fourth	L to R
==, !=	fifth	L to R
&&	sixth	L to R
||	seventh	L to R
= (assignment operator)	last	R to L

Bitwise operators



DSAseries

code.cpp

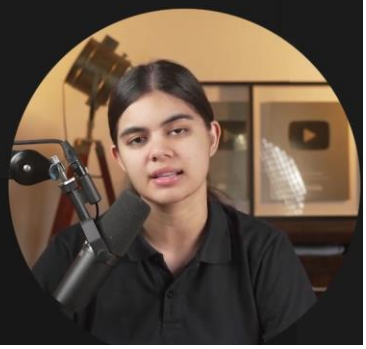
```
#include <iostream>
using namespace std;

int main() {

    cout << ((5 - 2) * 6) << endl;
    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
-7
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
18
apnacollege@Shradha DSAseries %
```



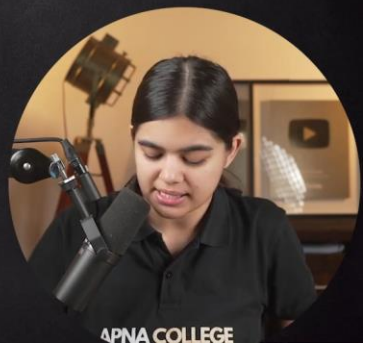
Operator Precedence

ans = 4 * 5 % 2

associativity

20 % 2 ⇒ 0

Operators	Precedence	
!, +, – (unary operators)	first	R to L
*, /, %	second	L to R
+, –	third	L to R
<, <=, >=, >	fourth	L to R
==, !=	fifth	L to R
&&	sixth	L to R
||	seventh	L to R
= (assignment operator)	last	R to L



DSAseries

code.cpp

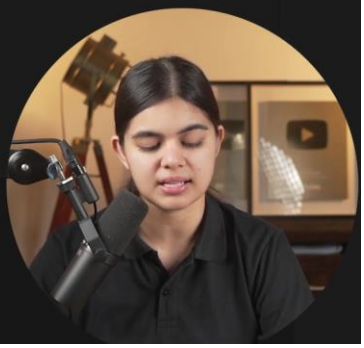
```cpp
#include <iostream>
using namespace std;

int main() {

    cout << (4 * 5 % 2) << endl;
    return 0;
}
```

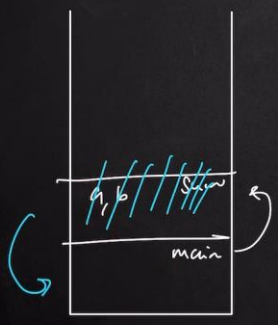
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

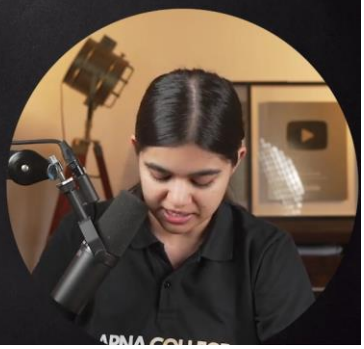
```
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
-7
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
18
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
0
apnacollege@Shradha DSAseries %
```



Scope → area variable → access / usage

↳ variables





Scope → area variable → access / usage

↳ variables

local

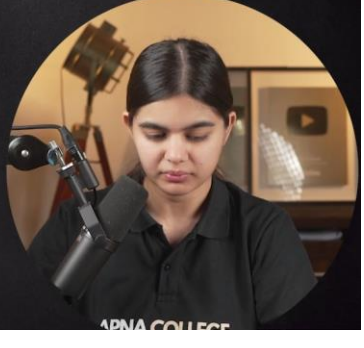
global

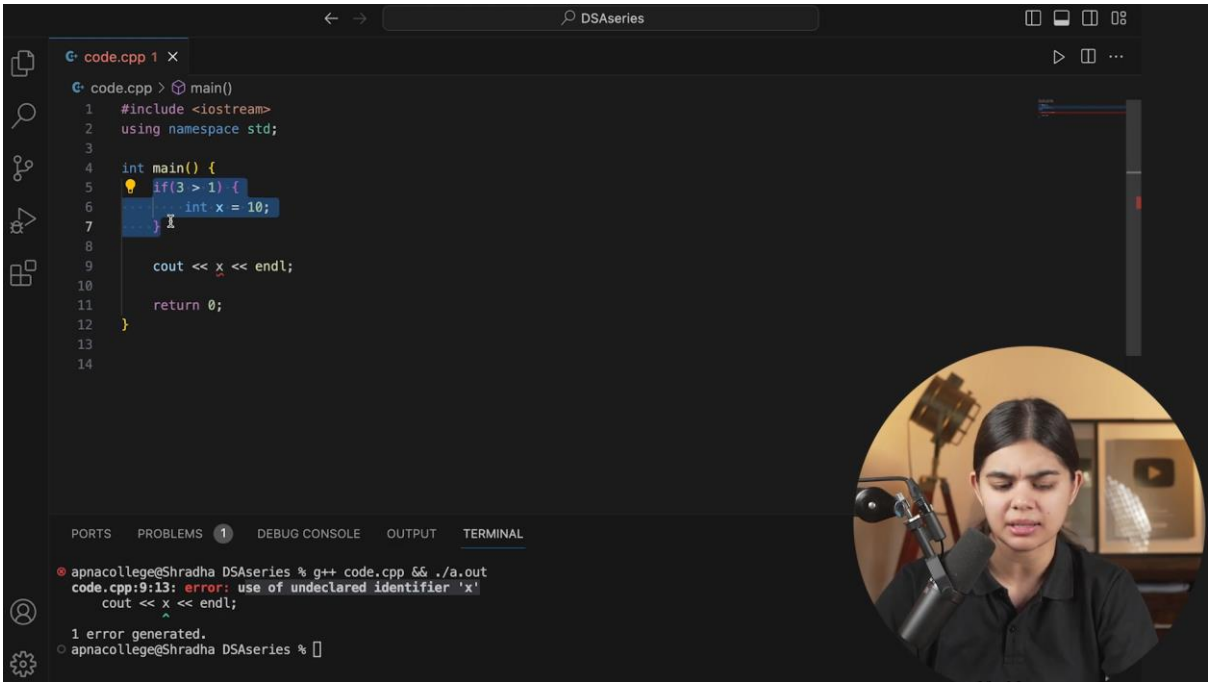
if - else, loops

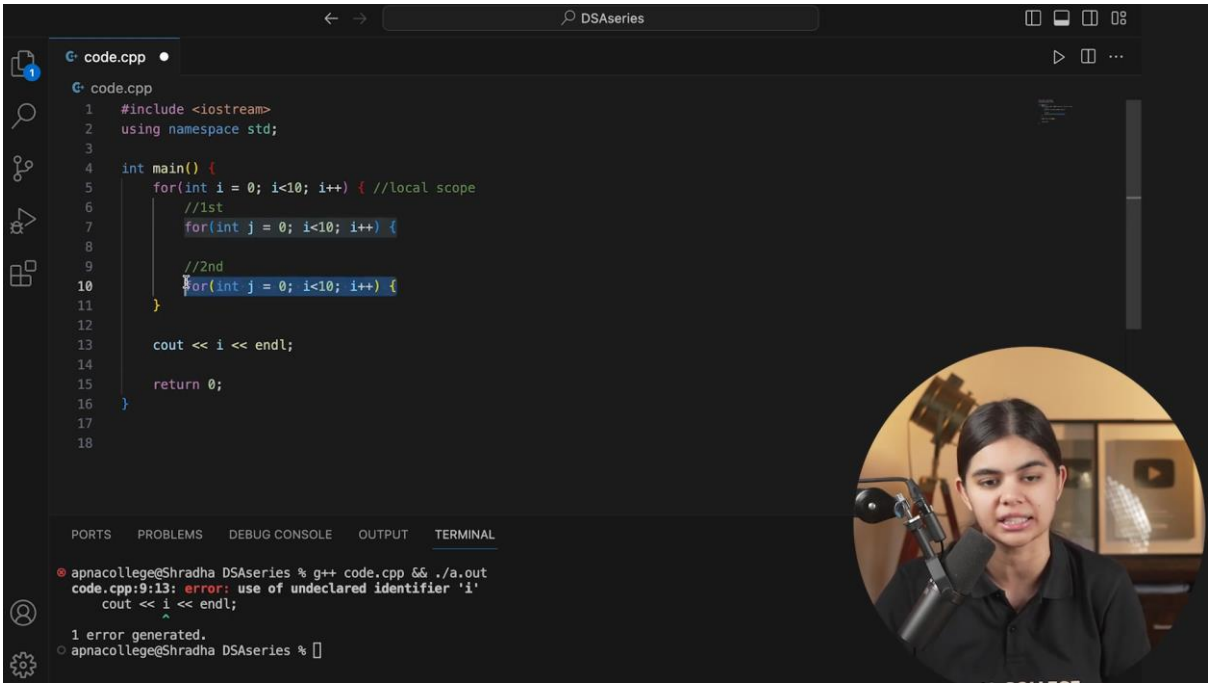
functions

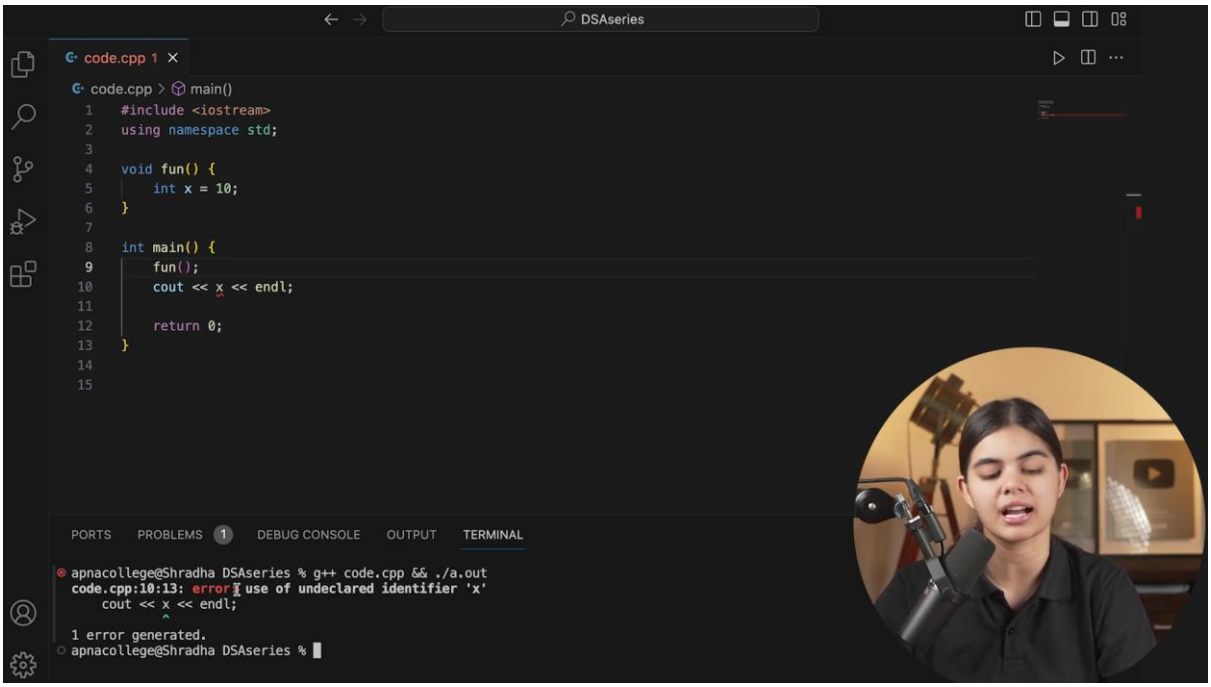
Block of code

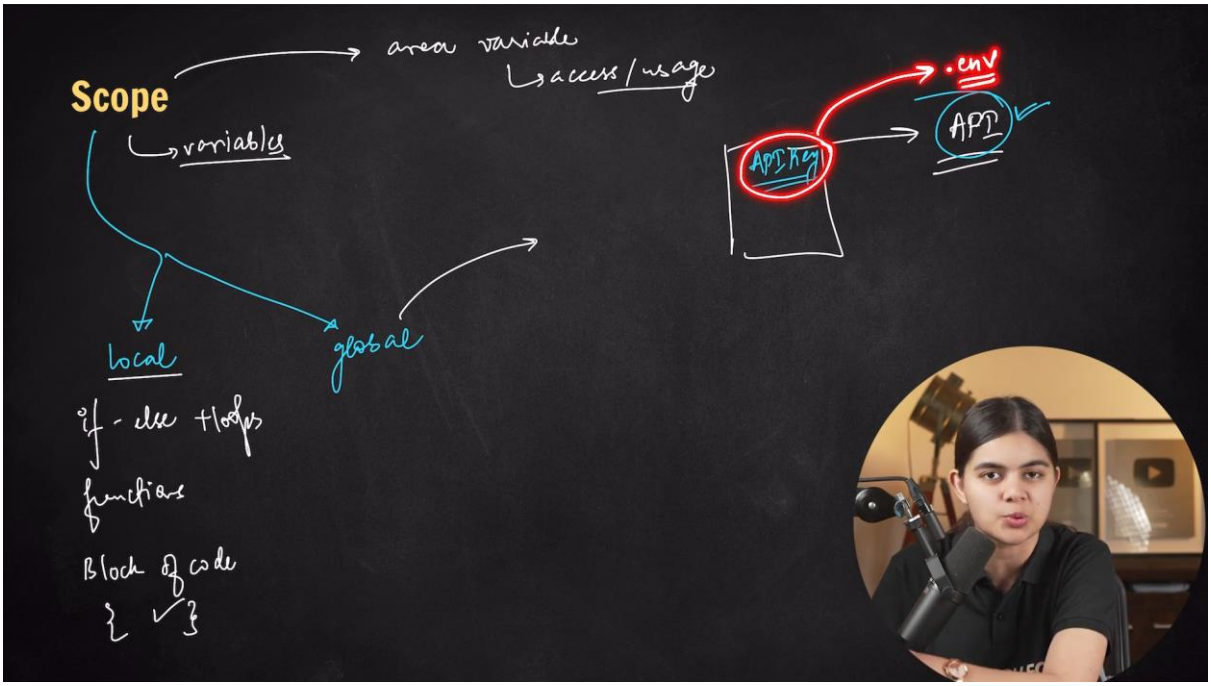
{ ✓ }

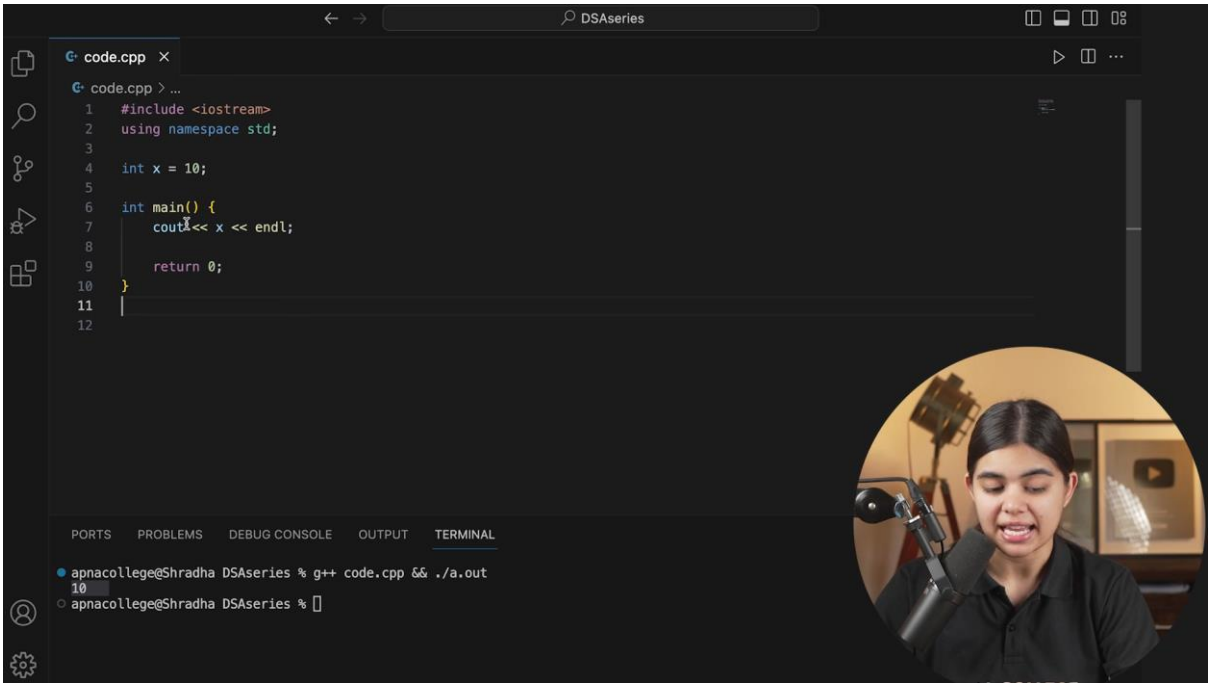


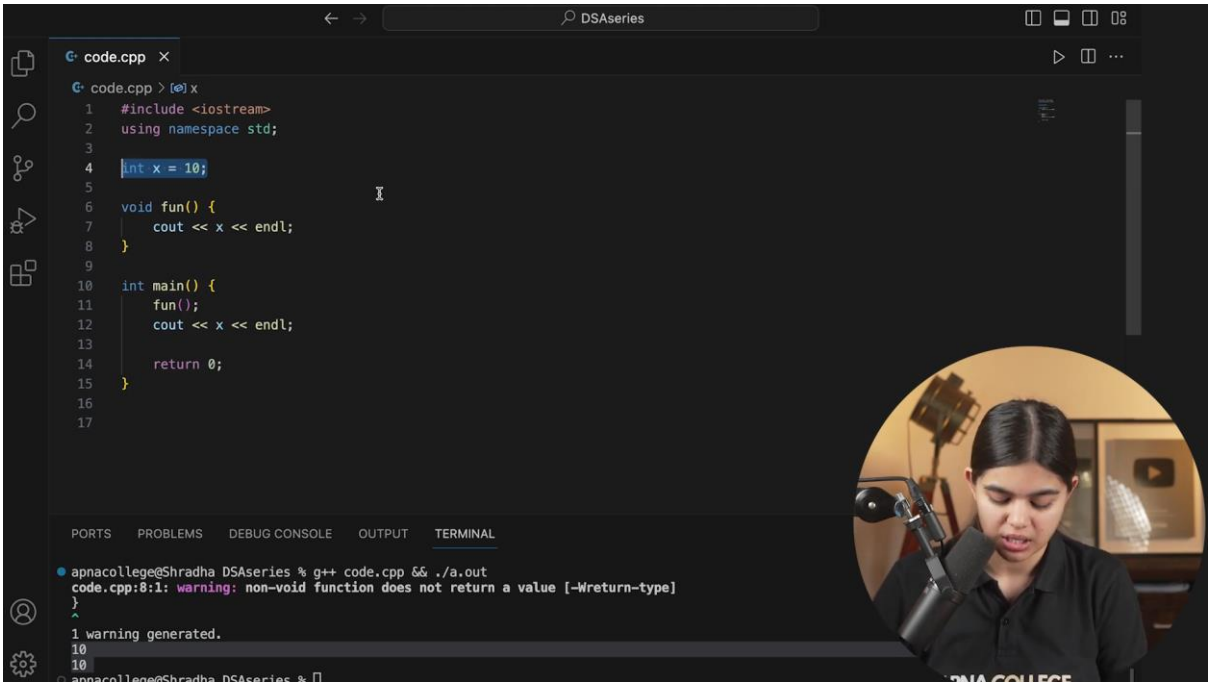

DSAseries
code.cpp 1
code.cpp > main()
#include <iostream>
using namespace std;
int main() {
if(3 > 1) {
int x = 10;
}
cout << x << endl;
return 0;
}
PORTS
PROBLEMS 1
DEBUG CONSOLE
OUTPUT
TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
code.cpp:9:13: error: use of undeclared identifier 'x'
cout << x << endl;
1 error generated.
apnacollege@Shradha DSAseries %


DSAseries
code.cpp
#include <iostream>
using namespace std;
int main() {
for(int i = 0; i<10; i++) { //local scope
//1st
for(int j = 0; i<10; i++) {
//2nd
for(int j = 0; i<10; i++) {
}
cout << i << endl;
return 0;
}
PORTS
PROBLEMS
DEBUG CONSOLE
OUTPUT
TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
code.cpp:9:13: error: use of undeclared identifier 'i'
cout << i << endl;
1 error generated.
apnacollege@Shradha DSAseries %


DSAseries
code.cpp 1
code.cpp > main()
#include <iostream>
using namespace std;
void fun() {
int x = 10;
}
int main() {
fun();
cout << x << endl;
return 0;
}
PORTS
PROBLEMS 1
DEBUG CONSOLE
OUTPUT
TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
code.cpp:10:13: error: use of undeclared identifier 'x'
cout << x << endl;
1 error generated.
apnacollege@Shradha DSAseries %


Scope
area variable
access / usage
variables
.env
API Key
API
local
global
if - else
loops
functions
Block of code
{ }

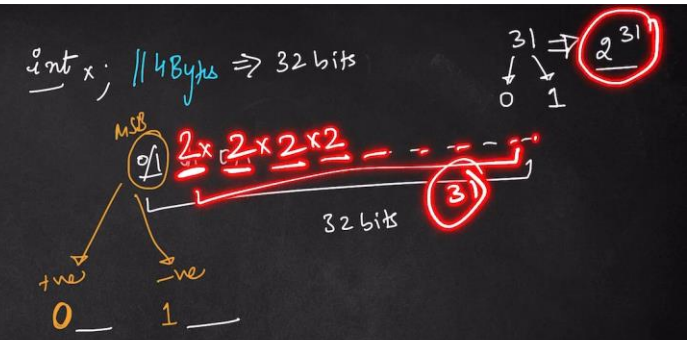

DSAseries
code.cpp
#include <iostream>
using namespace std;
int x = 10;
int main() {
cout << x << endl;
return 0;
}
PORTS
PROBLEMS
DEBUG CONSOLE
OUTPUT
TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
10
apnacollege@Shradha DSAseries %

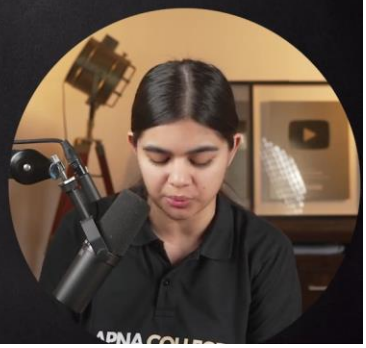

DSAseries
code.cpp
#include <iostream>
using namespace std;
int x = 10;
void fun() {
cout << x << endl;
}
int main() {
fun();
cout << x << endl;
return 0;
}
PORTS
PROBLEMS
DEBUG CONSOLE
OUTPUT
TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
code.cpp:8:1: warning: non-void function does not return a value [-Wreturn-type]
1 warning generated.
10
10

Data Type Modifiers

change meaning of data types

- long
- short
- long long
- signed
- unsigned

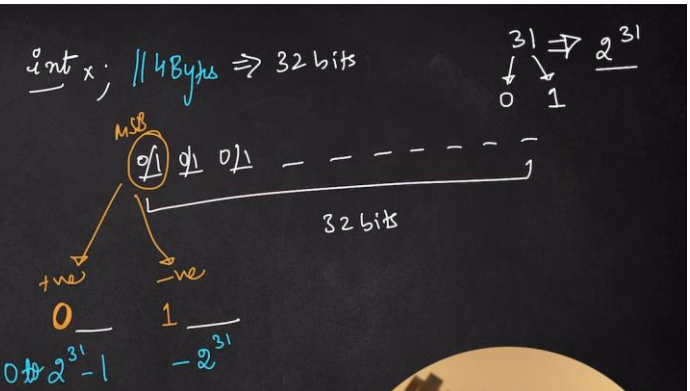


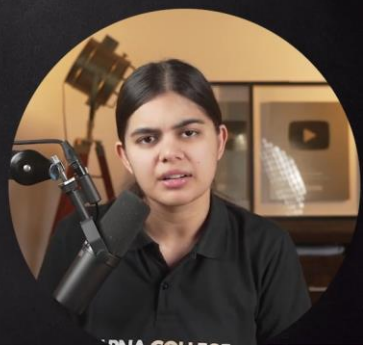


Data Type Modifiers

change meaning of data types

- long
- short
- long long
- signed
- unsigned

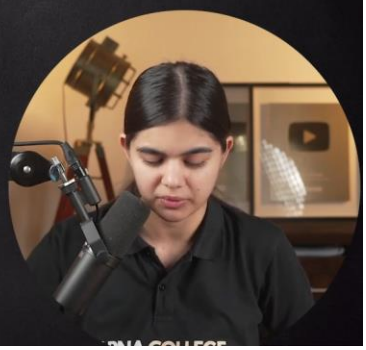




Data Type Modifiers

change meaning of data types

- long
- short
- long long
- signed
- unsigned



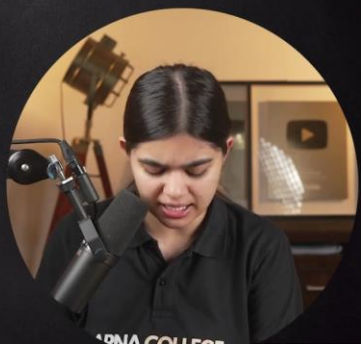
Data Type Modifiers

change meaning of data types

- long >=4Bytes (long int, long double)
- short
- long long
- signed
- unsigned

int x; // 4Bytes ⇒ 32 bits ⇒ -2^{31} to $+2^{31}-1$

long int ⟶ 64 bits ⇒ -2^{63} to $+2^{63}-1$



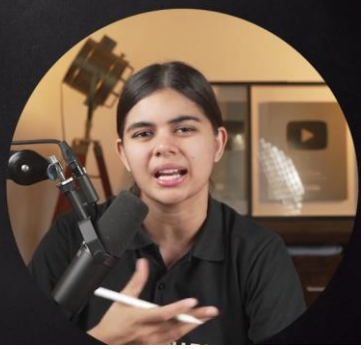
Data Type Modifiers

change meaning of data types

- long
- short
- long long
- signed
- unsigned

signed int → +ve, -ve ⇒ -2^{31} to $+2^{31}-1$

customerId int → +ve



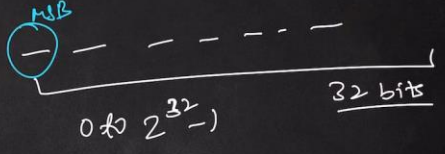
Data Type Modifiers

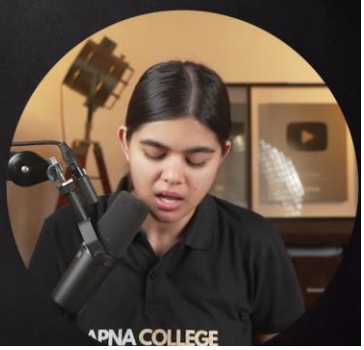
change meaning of data types

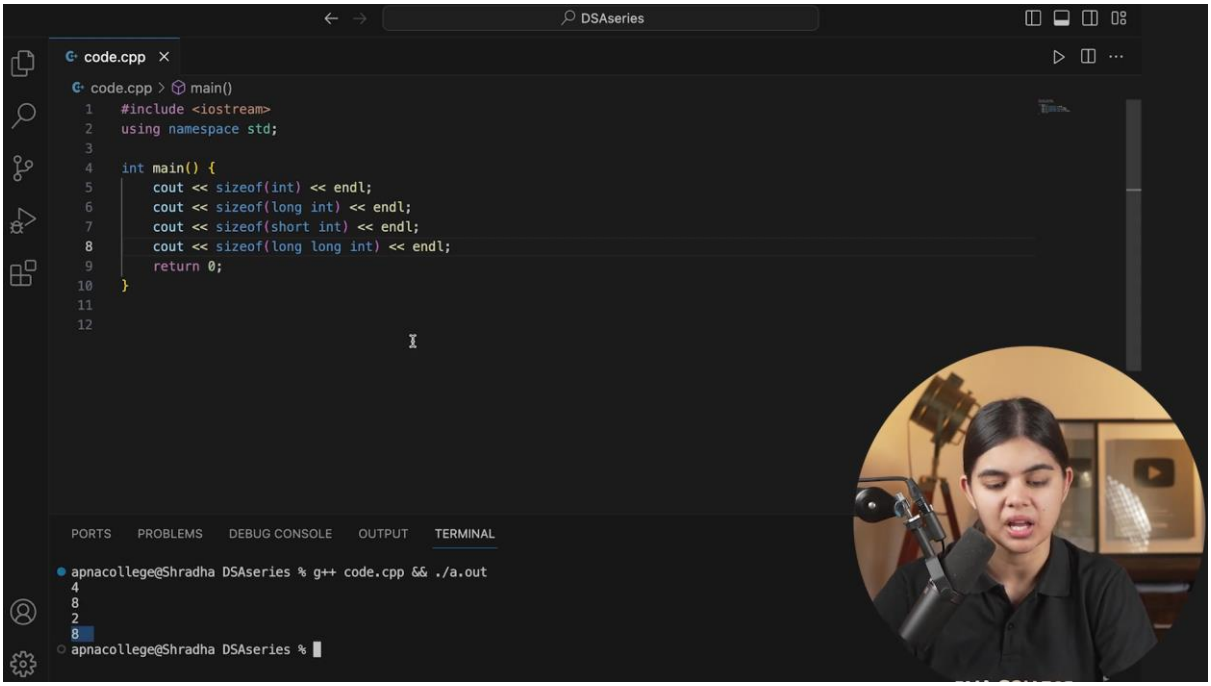
- long
- short
- long long
- signed
- unsigned

unsigned int customerId → +ve ⇒ -2^{31} to $+2^{31}-1$

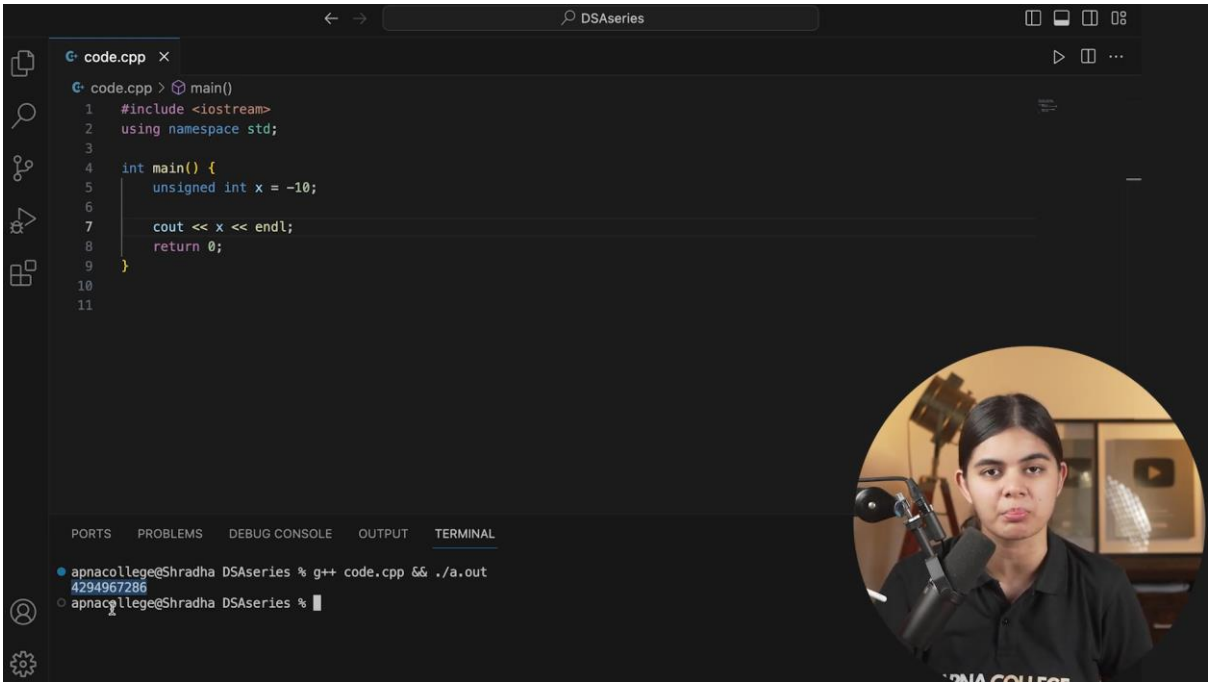
customerId int → +ve

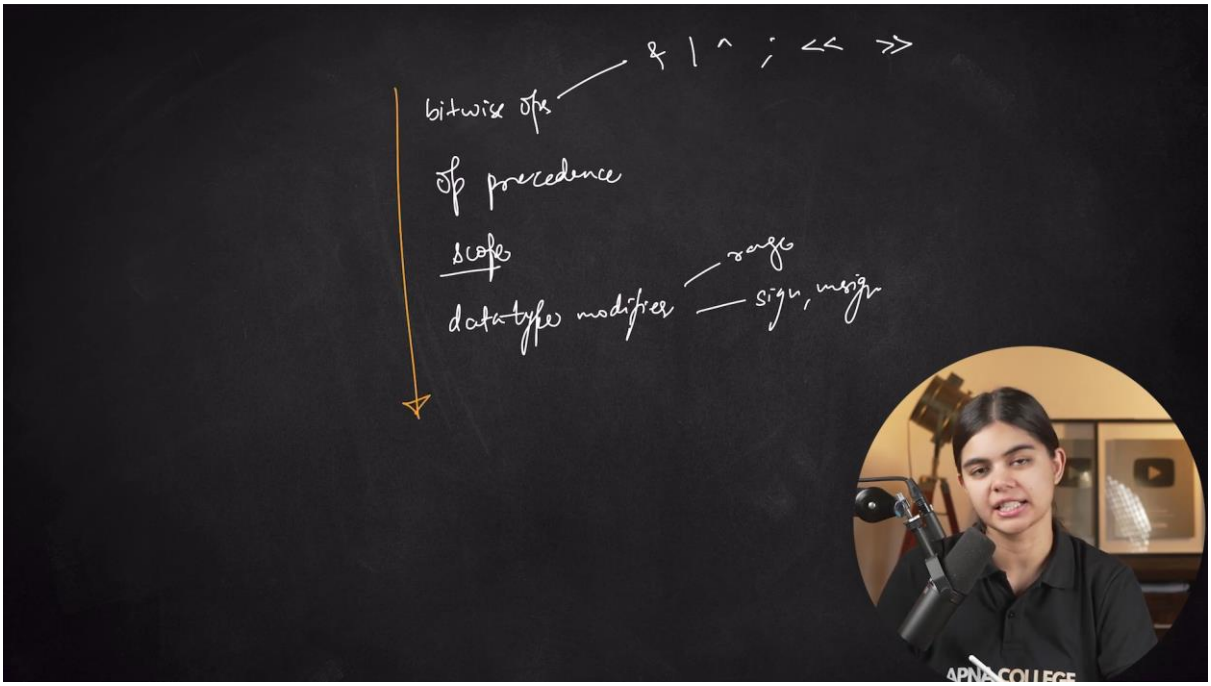


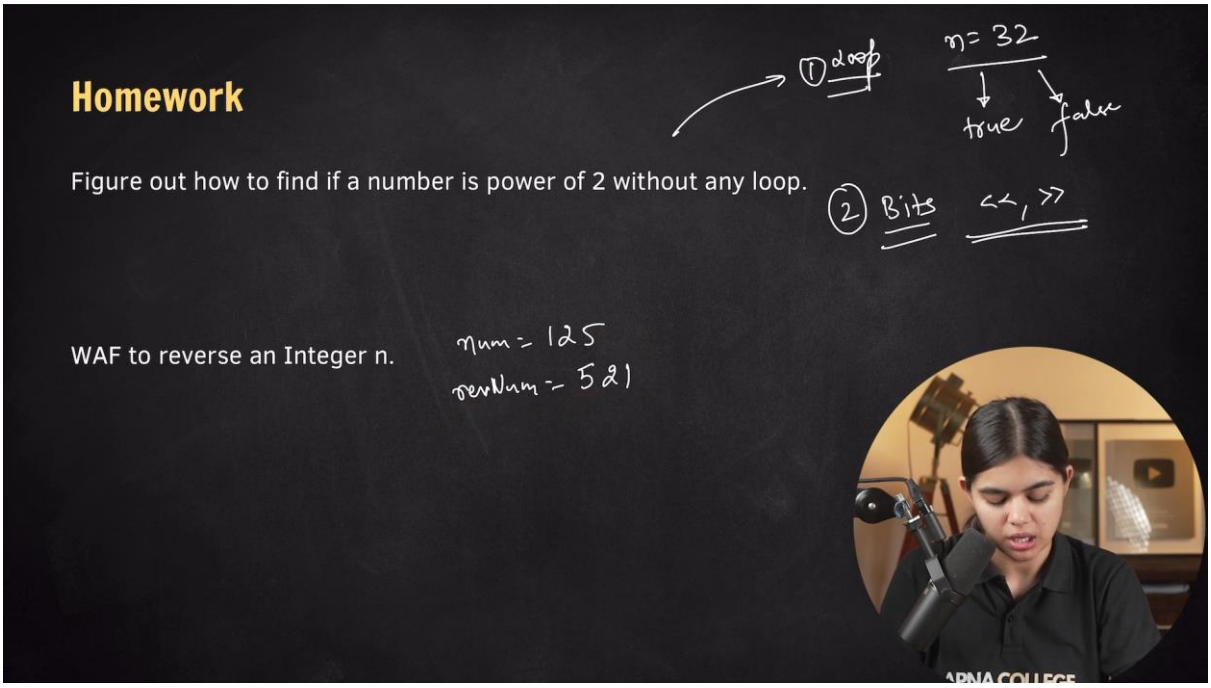



DSAseries
code.cpp
code.cpp > main()
#include <iostream>
using namespace std;
int main() {
cout << sizeof(int) << endl;
cout << sizeof(long int) << endl;
cout << sizeof(short int) << endl;
cout << sizeof(long long int) << endl;
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
4
8
2
8
apnacollege@Shradha DSAseries %


Data Type Modifiers
change meaning of data types
x = -10
2's complement
long
short
long long
signed
unsigned


DSAseries
code.cpp
code.cpp > main()
#include <iostream>
using namespace std;
int main() {
unsigned int x = -10;
cout << x << endl;
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
4294967286
apnacollege@Shradha DSAseries %





Homework Solution:

// Figure out how to find if a number is power of 2 without any loop.

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

bool isPowerOf2(int x) {
   // Check if x is greater than 0 and if x & (x - 1) equals 0
   return (x > 0) && ((x & (x - 1)) == 0);
}
```

```
int main() {
    int x = 64;

    if (isPowerOf2(x)) {
        cout << x << " is a power of 2" << endl;
    } else {
        cout << x << " is not a power of 2" << endl;
    }

    return 0;
}
```

```
// WAF to reverse an Integer n.

#include <iostream>
using namespace std;

int reverseInteger(int n) {
    int reversed = 0;
    // Loop to reverse the digits of the number
    while (n != 0) {
        int digit = n % 10;             // Extract the last digit
        reversed = reversed * 10 + digit;   // Append the digit to the reversed number
        n /= 10;                        // Remove the last digit from n
    }
    return reversed;
}

int main() {
    int n = 12345;
    cout << "Reversed integer: " << reverseInteger(n) << endl;
    return 0;
}
```