

QUIZ:C++

QUIZ: ++ / -- Operators:-

? Q1:

```
int a = 10, b = 5;  
cout << a + b * 2;
```

- a) 30
- b) 20
- c) 25
- d) 15

Ans)b)20

? Q2:

```
int a = 5;  
cout << ++a + a++;
```

- a) 11
- b) 12
- c) 10
- d) 13

 Step-by-step:

- **++a** → Pre-increment → **a** becomes 6, then used → value = 6
- **a++** → Post-increment → **a** is 6 (used first), then becomes 7 → value = 6
- So: **6 + 6 = 12**

✓ Answer: b) 12

? Q3

```
int x = 3;  
int y = ++x * 2 + x--;  
cout << y;
```

- a) 10
- b) 11
- c) 12
- d) 13

🔍 Step-by-step:

- **++x** → Pre-increment → **x** becomes 4 → used → 4
- So expression is: **4 * 2 + x--**
- Now **x--** → current **x = 4** → used as 4, then becomes 3
- So: **4 * 2 = 8**, and **8 + 4 = 12**

✓ Answer: c) 12

? Q4

```
int x = 10;  
cout << x++ - --x;
```

- a) 0
- b) 1
- c) -1
- d) 2

Ans)a)0

 Step-by-step:

- **x++** → Post-increment → use 10, then x becomes 11
- **--x** → Pre-decrement → x becomes 10, then used
- So: **10 - 10 = 0**

 Answer: a) 0

 Q5

```
int a = 5;  
int b = a++ + ++a + a++;  
cout << b;
```

- a) 18
- b) 17
- c) 16
- d) 19

 Breakdown:

Initial: **a = 5**

- **a++** → Use 5, then a = 6
- **++a** → a becomes 7, use 7

- `a++` → Use 7, then `a = 8`

→ So: `5 + 7 + 7 = 19`

✅ Answer: d) 19

? Q6

```
int x = 4;  
int y = ++x + x++ + ++x;  
cout << x << " " << y;
```

- a) 7 17
- b) 7 16
- c) 6 17
- d) 7 18

🧠 Breakdown:

Initial: `x = 4`

- `++x` → `x = 5`, use 5
- `x++` → use 5, then `x = 6`
- `++x` → `x = 7`, use 7
- `y = 5 + 5 + 7 = 17`
- Final `x = 7`

✅ Answer: a) 7 17

? Q14

```
int a = 2;
```

```
int b = 3;
```

```
int c = a++ + b++ + ++a + ++b;
```

```
cout << c;
```

a) 11

b) 12

c) 13

d) 14

🧠 Breakdown:

- **a++** → use 2, then a = 3
- **b++** → use 3, then b = 4
- **++a** → a = 4, use 4
- **++b** → b = 5, use 5

→ 2 + 3 + 4 + 5 = 14

✅ Answer: d) 14

? Q4

```
int x = 1;
```

```
cout << x++ + x++ + x++;
```

- a) 3
- b) 4
- c) 6
- d) 5

🧠 Breakdown:

- `x++` → use 1, then `x = 2`
- `x++` → use 2, then `x = 3`
- `x++` → use 3, then `x = 4`

→ `1 + 2 + 3 = 6`

✅ Answer: c) 6

? Q5

```
int i = 10;
```

```
i = i++ + ++i;
```

```
cout << i;
```

- a) 21
- b) 22
- c) 20
- d) 23

 Breakdown:

- `i++` → use 10, then `i = 11`
- `++i` → `i = 12`, use 12
→ So: `i = 10 + 12 = 22`

 Answer: b) 22

 QUIZ: Bitwise Operators:-

? Q1

```
int a = 5;      // 0101
```

```
int b = 3;      // 0011
```

```
cout << (a & b);
```

- a) 1
- b) 3
- c) 5
- d) 7

Ans) a) 1

? Q2

```
int a = 12;     // 1100
```

```
int b = 10;     // 1010
```

```
cout << (a | b);
```

- a) 6
- b) 14

- c) 2
- d) 15

Ans)b)14

? Q3

```
int x = 7;      // 0111
```

```
cout << (~x);
```

- a) -7
- b) -8
- c) 7
- d) 8

Ans)b)-8

? Q4

```
int x = 4;
```

```
cout << (x << 1);
```

- a) 2
- b) 8
- c) 6
- d) 4

► Left Shift:

- $4 = 0100$
- $4 \ll 1 = 1000 = 8$

 Answer: b) 8

✅ Q5

```
int y = 16;
```

```
cout << (y >> 2);
```

► Right Shift:

- $16 = 10000$
- Shift right by 2 $\rightarrow 00100 = 4$

✅ Answer: b) 4

Great catch, Boss! 🔍

This confusion is very common — and you're asking the right question. Let's break it down super clearly so it clicks forever.

😞 Why ~ 7 is NOT 8 (but -8 instead)?

◆ You're thinking:

$7 = 0111$

$\sim 7 = 1000 \rightarrow$ That's 8, right?

But that's not correct, because:

! C++ uses 2's Complement for representing negative integers.

✅ Step-by-step for ~ 7 :

◆ Step 1: 7 in binary (32-bit):

CopyEdit

00000000 00000000 00000000 00000111

◆ Step 2: Apply bitwise NOT (~)

11111111 11111111 11111111 11111000

◆ Step 3: What number is that?

That's -8, because:

- It's a negative number in 2's complement form
- If you reverse it (take 2's complement again), you'll get back to 8

🧠 So remember:

$\sim x = -(x + 1)$ in C/C++ for integers

✅ So:

$\sim 7 = -(7 + 1) = -8$

🧪 Test It:

Try this in code:

```
int x = 7;

cout << ~x;          // prints -8

cout << -(x + 1);    // also prints -8
```

Real Answer Recap:

- `~7` gives `-8`, not `8`
- Because it's NOT bit flip of `0111` \rightarrow `1000`, but a 2's complement flip