

Loop Error

Logical Error

A logical error happens when your program runs without crashing, but it gives the wrong output because the logic or reasoning in your code is incorrect.

- Using the wrong operator (> instead of <)
- Wrong variable used in calculation
- Wrong formula or condition
- Misplaced loop or incorrect order of steps

EXAMPLES

PROGRAM

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

int main()
{
    int a = 10; b = 5; int average; average = a + b / 2; cout << "Average = " << average << endl; return 0; }
```

Output: Average = 12

The correct average of 10 and 5 is 7.5, but the program gives 12. **Why?** Because of operator precedence — division happens before addition. So the code does this:

average = 10 + (5 / 2); average = 10 + 2; // integer division average = 12;

Correct code: average=(a+b) /2;

Now, output: Average=7

(If you use float, you can get 7.5)

PROGRAM

```
#include <iostream>
using namespace std;
int main() { int age = 18; if (age < 18) {
    cout << "You are an adult."; } else { cout << "You are not an adult."; } return 0; }
```

Explanation:

The program runs without any syntax or runtime error.

But the logic is wrong — the condition age < 18 is opposite of what we want.

For age = 18, it will print “You are not an adult.”, which is incorrect logically.

Correct code:

```
if (age >= 18) { cout << "You are an adult."; } else { cout << "You are not an adult."; }
```

Now it will correctly print:

You are an adult.

PROGRAM

```
#include using namespace std; int main() { int marks; cout << "Enter your marks: "; cin >> marks; if (marks > 50) { cout << "You passed!" << endl; } else { cout << "You failed!" << endl; } return 0; }
```

Explanation:

Suppose the passing marks are 50 or above.

The program uses the condition `marks > 50`, so if a student scores exactly 50, the program will show "You failed!", which is incorrect.

This is a logical error, because the code runs fine but the logic doesn't match the real rule.

Correct version:

```
if (marks >= 50) { cout << "You passed!"; } else { cout << "You failed!"; }
```

PROGRAM

```
#include using namespace std; int main() { int num; cout << "Enter a number: "; cin >> num; cout << "Square of " << num << " is " << square(num); return 0; }
```

Output:

Enter a number: 5 Square of 5 is 10

→ This is wrong! The square of 5 should be 25, not 10.

Explanation:

The logical error is in the function's return statement:

`return n * 2;` // doubles the number instead of squaring it

It should be:

`return n * n;` // square of the number

Correct Code:

```
#include using namespace std;  
  
int square(int n) { return n * n; }  
  
int main() { int num; cout << "Enter a number: "; cin >> num; cout << "Square of " << num << " is " << square(num); return 0; }
```

Correct Output:

Enter a number: 5 Square of 5 is 25

PROGRAM

```
#include using namespace std; int main() { int a=1; if(a=1) cout<<a; cout<<a+1; cout<<a+2; return 0; }
```

OUTPUT 123

If curly braces are not used then the statement after if is considered in if condition the other two statements are executed separately. Program will run but it shows logical error.

PROGRAM

```
#include <using namespace std> int main() { int a=3; if(a) { cout<<a; } return 0; }
```

OUTPUT 3

In this if condition is always true and always runs in program.

`= =` is not equal to `=`

PROGRAM

```
#include <using namespace std> int main() { int a=13; if(a=3) { cout<<a; } return 0; }
```

Program will run because it is **logical error**. `a= =13`(whether `a=13` or not) `a=3`(assigned a value to a integer `a`)

PROGRAM

```
#include <using namespace std> int main() { int a=5; if( "a>=3" ) { cout<<a; } return 0; }
```

The program will run because its is **logical error**.the statement written in double quote is always consider string and in if else string is always true.

`"a>=3"` is not equal to `a>=3`

`"a>=3"` in if always true because it is string.

`a>=3` in if true or false because it is condition.

SYNTAX ERROR

A syntax error means there is a mistake in the way your code is written — it breaks the grammar rules of the programming language.

Easy meaning: Your code is written in a way the computer doesn't understand.

Example:

```
cout "Hello"; // X syntax error (missing <<)
```

Correct:

```
cout << "Hello";
```

So, a syntax error is like a spelling or grammar mistake in your code.

PROGRAM

```
#include <using namespace std> int main() { int a; a=7; if(a>=3) { cout<<a; } cout<<a; else { cout<<"hello"; } return 0; }
```

ERROR

No statement written between if and else.After if eles will be directly written.

PROGRAM

```
#include using namespace std; int main() { int a=7; if(a<=3) { cout<<a; } else(a>3)  
{cout<< "hello"; } return 0; }
```

ERROR There is no condition in eles.

PROGRAM

```
#include using namespace std; int main() { int a=3; if(a!=1) { cout<<a; } else if(a>3) cout<<a;  
cout<<a+1; else cout<<a; cout<<a+1; return 0; }
```

ERROR

Else without if

PROGRAM

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

ERROR Syntax error shows The value of if is important.the condition is not written in if only operator or string is written in if then it will run but it shows logical error.