

FIRST C++ CODE (HELLO WORLD)

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world";
7 }
```

<<endl - means new line

Skeleton of c++

#include<iostream>

To use cout use std (using namespace std)

How to use endl, cout ?

The screenshot shows a code editor interface with a dark theme. On the left, a code editor window displays the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello World" << endl;
7     cout << "100xSchool";
8     cout << "Hello 3";
9     cout << "Hello 4";
10    cout << "Hello 5";
11 }
```

On the right, there is an "Output" panel with the following content:

- An input field labeled "Enter Input here".
- A note: "If your code takes input, add it in the above input field."
- An "Output" section containing the text "Submission Queued...".

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

int main()
{
    cout << "Hello World";
    cout << endl;

    cout << "100xSchool";
    cout << endl;

    cout << "Hello 3";
    cout << endl;

    cout << "Hello 4";
    cout << endl;

    cout << "Hello 5";
    cout << endl;
}
```

Enter Input here
If your code takes input, add it in the above box before running

Output

Submission Queued...

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello World" << endl << "100xSchool" << endl << "Hello 3";
7     cout << endl;
8
9     cout << "Hello 4";
10    cout << endl;
11
12    cout << "Hello 5";
13    cout << endl;
14 }
```

Enter Input here
If your code takes input, add it in the above box before running

Output

Status : Successfully executed

Time: 0.0000 secs Memory: 3.552 Mb

Your Output

```
Hello World
100xSchool
Hello 3
Hello 4
Hello 5
```

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

Enter Input here

If your code takes input, add it in the above box before running.

Output

Submission Queued...

The output consists of a 5x5 grid of red 'X' characters, arranged in a diamond pattern. The grid has 5 rows and 5 columns. The 'X's are positioned at the intersections of the grid lines, creating a symmetrical diamond shape centered in the grid.

Note: In C++ that semicolon is important

Output

Status : Compilation error

Error

```
sol.cpp: In function 'int main()':
sol.cpp:6:26: error: expected ';' before ')' token
      cout << "Hello world"
                           ^
;
7 | }
   | ~
```

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

int main()
{
```

```
#include<iostream>

int main()
{
    std::cout << "Hello world";
}
```

```
#include<iostream>

int main()
{
    std::cout << "Hello world";
}
```



```
#include<iostream> // header file which is helping us to use cout
using namespace std;

int main()
{
    cout << "Hello world";
```



SOME IMPORTANT RULES TO REMEMBER

- Semicolon: Every statement must end with ;
- Case Sensitive: cout will work, but COUt will not.
- Printing Text: Text must be inside double quotes.
- New Line: For a new line, use endl.
- Brackets Come in Pairs

Arithmetic operator

ARITHMETIC OPERATORS

Operator	Meaning
+	Add
-	Subtract
*	Multiply
/	Divide
%	Remainder

```
1 #include<iostream> // header file which is helping us to use cout
2 using namespace std;
3
4 int main()
5 {
6     cout << 10 + 2 << endl;
7     cout << 10 - 2 << endl;
8     cout << 10 * 2 << endl;
9 }
10 |
```

Enter Input here
If your code takes input, add it in the area below.

Output

Status : Successfully executed

Time: 0.0000 secs Memory: 3.532 Mb

Your Output

```
12
8
20
```

```
1 #include<iostream> // header file which is helping us to use cout
2 using namespace std;
3
4 int main()
5 {
6     cout << 10 + 3 << endl; // 13
7     cout << 10 - 3 << endl; // 7
8     cout << 10 * 3 << endl; // 30
9     cout << 10 / 3 << endl; // 3
10 }
11 |
```

Enter Input
Dinu

If your code takes input, add it in the area below.

Output

Status : Successfully executed

Time: 0.0000 secs Memory: 3.512 Mb

Your Output

```
13
7
30
3
```

Floating point

$$10.0 / 3 = 3.3333$$

$$10 / 3.0 = 3.3333$$

$$10.0 / 3.0 = 3.3333$$

VARIABLES

A variable is a named box in the computer's memory where we store some data.

DATA TYPES (FOUNDATION)

Type	Use
int	Whole numbers
long long	Big numbers
double	Decimal
char	Single character
bool	true / false

SOME IMPORTANT RULES TO REMEMBER

- Variable must be declared before use
- No spaces in variable names
- Case sensitive: marks ≠ Marks
- Name should be meaningful (not x1, a2 everywhere)

The screenshot shows a C++ code editor interface with a dark theme. The code in the editor is:

```
1 #include<iostream> // header file which is helping us to use cout
2 using namespace std;
3
4 int main()
5 {
6     int age = 21;           // Declaring a variable named 'age' and assigning it the value 21
7     cout << "akkkkge" << endl; // Outputting the string "akkkkge" followed by a new line
8     cout << age << endl;    // Outputting the value of 'age' (21) followed by a new line
9 }
10
11 // text, number, boolean, etc.
12
13 // if we divide 2 integers, the result is always an integer.
14 // quotient
```

The editor has a toolbar with icons for file operations like Open, Save, and Run. To the right of the editor is a sidebar with the following sections:

- Run**: A button labeled "Run".
- Visual Studio**: A small icon.
- Enter Input here**: A text input field.
- If your code takes input, add**: A placeholder text.
- Output**: A section showing the execution results.
- Status**: Status message: "Status : Successfully executed".
- Time**: 0.0000 secs
- Memory**: 3.48
- Your Output**: The output text: "akkkkge" and "21".

The status bar at the bottom of the editor shows the file path: "C:\Users\Bhuv\OneDrive\Desktop\Untitled 1.cpp" and the line number: "703521719".

The screenshot shows a code editor with the following C++ code:

```
#include<iostream> // header file which is helping us to use cout
using namespace std;

int main()
{
    double age = 21.89;
    cout << age << endl;
}

// text, number, boolean, etc.
// if we divide 2 integers, the result is always an integer.
// quotient
```

To the right of the code editor is a terminal window showing the output of the program. The terminal has tabs for "Enter Input here" and "Output". The "Output" tab shows:

- Status: Successfully executed
- Time: 0.0000 secs
- Memory: 3.5 Mb
- Your Output: 21.89

Now Lets take input

Example below:

The screenshot shows a code editor with the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int a;
7     int b;
8
9     cin >> a;
10
11    cin >> b;
12
13    cout << a + b << endl;
14 }
```

To the right of the code editor is a terminal window showing the output of the program. The terminal has tabs for "Output" and "Status". The "Status" tab shows:

- Status: Successfully executed
- Time: 0.0000 secs
- Memory: 3.532 Mb

The "Output" tab shows:

- Sample Input: 10 20
- Your Output: 30

Online C++ Compiler and Visualizer

C++ Run Visualize Code

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int a;
7     int b;
8
9     cin >> a >> b;
10
11    cout << a + b << endl;
12 }
```

Output

20 40

Submission Queued...

Online C++ Compiler and Visualizer

C++ Run Visualize Code

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int a;
7     int b;
8     double c;
9
10    cin >> a >> b >> c;
11
12    cout << a + b << endl;
13
14    cout << c << endl;
15 }
```

Output

20 40 20.7

Status : Successfully executed

Time: 0.0000 secs Memory: 3.812 Mb

Sample Input

20 40 20.7

Your Output

60
20.7

Relational Operator

RELATIONAL OPERATORS

Operator	Meaning	Example	Result
>	Greater than	5 > 3	true
<	Less than	5 < 3	false
>=	Greater than or equal to	5 >= 5	true
<=	Less than or equal to	4 <= 3	false
==	Equal to	5 == 5	true
!=	Not equal to	5 != 3	true

The screenshot shows a C++ code editor interface with the following code:

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

int main()
{
    bool ans1 = 5 == 5; // true -> 1
    bool ans2 = 6 != 5; // true -> 1
    bool ans3 = 6 < 5; // false -> 0

    cout << ans1 << endl;
    cout << ans2 << endl;
    cout << ans3 << endl;
}
```

The code defines three boolean variables: ans1, ans2, and ans3. ans1 is true (1), ans2 is true (1), and ans3 is false (0). The code then prints these values to the console.

On the right side of the interface, there is a "Run" button, a "Visualizer" section with a video thumbnail, and a "Status" bar indicating "Status : Successfully executed". Below the status bar, there are "Time" and "Memory" metrics: 0.0000 secs and 3.512 Mb respectively. The "Your Output" section shows the printed values: 1, 1, and 0.

Logical Operator



Conditionals

A screenshot of a code editor interface. The code editor shows the following C++ code:

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

int main()
{
    int a, b, c;
    cin >> a >> b >> c;

    // max
    if(a > b and a > c)
    {
        cout << a;
    }
    else if(b > a and b > c)
    {
        cout << b;
    }
    else
    {
        cout << c;
    }
}
```

7979721719

The code editor has a toolbar at the top with icons for file, edit, run, and settings. To the right of the code editor is a terminal window showing the output of the program. The terminal window includes a video feed of a person, the status "Status : Successfully executed", execution time "0.0000 secs", memory usage "3.5 Mb", sample input "12 12 7", and your output "7".

At the bottom of the code editor, there is a control bar with a play button, a progress bar, the time "02:00:11/02:27:02", and other controls.