

≡ Untitled-1

C++ tut1.cpp M X



Tutorial.cpp > C++ tut1.cpp > main()

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     cout<<"Hello World";
5
6 //std::cout<<"Hello world";
7     return 0;
8 }
```

153

1

A small, dark gray icon of a document or file, consisting of a rectangle with rounded corners and a horizontal oval in the center.

2

master\*

⊗ 0 △ 0

Ln 8, Col 2

## Spaces: 4

UTF-8

LF

C++

Mac



Tutorial.cpp &gt; C++ tut3.cpp &gt; ...

```
1 #include<iostream>
2 using namespace std;
3
4 // This program is created by Priyanshu Pulak
5 /* This
6  is
7  a
8  multi
9  line
10 comment */
11
12 /* ' /n is used from where the line need to break
13 while endl is used in the last of the line ' */
14
15 int main(){
16     int sum = 89;
17     cout<< " Hello World "<<sum;
18     return 0;
19 }
```



The screenshot shows a dark-themed code editor window with a sidebar on the left containing various icons. The main area displays a C++ program named `tut4.cpp`. The code includes comments explaining global variables, built-in data types, user-defined types, derived types, and basic variable declarations. It also contains cout statements to print values of variables `a`, `b`, and `c`.

```
1 // ***** Global Variable *****
2 #include<iostream>
3 using namespace std;
4 int glo =7;
5 void sum(){
6     int a;
7     cout<< glo;
8 }
9 int main(){
10    int glow=9;
11    glow=78;
12
13
14 /* Following are the type of ' Built in data type '
15 1. int --- used for integer
16 2. float --- used for decimal of low precision
17 3. char --- used for single alphabet
18 4. double --- used for decimal of high precision
19 5. bool --- used for true (1) or false(0) */
20
21 /* User define data type (Study Later)
22 Struct
23 Union
24 Enum */
25
26 /* Derived type data type (Study later)
27 Array
28 Function
29 Pointer */
30
31 int a=14, b=15;
32 float pi=3.14;
33 char c = 'u' ;
34
35
36 cout<<"This is tutorial 4. \nHere the value of a is "<<a<<. \nThe value of b is "<<b;
37 cout<<"\nThe value of pi is: "<<pi;
38 cout<<"\nThe value of c is: "<<c ;
39
40
41
42 } return 0;
43 }
```

The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, displaying a C++ file named `tut5.cpp`. The code in the editor is as follows:

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

int main()
{
    int num1, num2;
    cout<<"Enter the value of num 1:\n"; /* '<<' is called insertion operator */
    cin>>num1; /* '>>' is called Extraction operator */
    cout<<"Enter the value of num2:\n";
    cin>>num2;

    cout<<"The sum is"<< num1+num2;

    return 0;
}
```

The left sidebar, titled "EXPLORER", lists various files under "TUTORIALS", including `Tutorial.cpp`, `tut1`, `tut3`, `tut4`, `tut5`, `tut6`, `tut7`, `tut8`, `tut9`, `tut10`, `tut11`, `tut12`, `tut13`, `tut14`, `tut15`, `tut16`, and `tut17`. The file `tut5.cpp` is currently selected, highlighted with a blue bar at the bottom of the list.

The bottom status bar indicates the file is on "master\*", has 0 changes, and shows statistics: Ln 1, Col 1, Spaces: 4, UTF-8, LF, C++, Mac, and a refresh icon.

The screenshot shows a dark-themed code editor interface with the following details:

- EXPLORER** sidebar on the left showing a tree view of files under "TUTORIALS". The file "tut6.cpp" is selected.
- OPEN EDITORS** sidebar showing multiple tabs, with "tut6.cpp" being the active tab.
- Editor Area**: The main window displays the content of "tut6.cpp". The code is a C++ program demonstrating various operators. It includes:
  - Arithmetic operators: `cout<<a+b<<endl` through `cout<<a/b<<endl`.
  - Assignment operators: `int a=3, b=9` and assignments like `a+=1`.
  - Comparison operators: `cout<<(a==b)<<endl` through `cout<<(a>=b)<<endl`.
  - Logical operators: `cout<<(a==b) && (a<b)` through `cout<<!(a==b) && (a<b)`.
- Bottom Status Bar**: Shows "Ln 1, Col 1" and other system information.

The screenshot shows a dark-themed code editor interface with the following details:

- Explorer Bar:** On the left, it lists numerous files under the "TUTORIALS" folder, including "tut1.cpp" through "tut26.cpp".
- Active Editor:** The main editor window displays the content of "tut7.cpp".
- Code Content:** The code in "tut7.cpp" demonstrates various C++ concepts:
  - Using the `#include<iostream>` header.
  - Defining a global variable `c=655;`.
  - Writing a `main()` function that reads values for `a` and `b`, calculates their sum, and prints the results.
  - Comments explaining built-in data types.
  - Working with `float`, `double`, and `long double` literals.
  - Using `cout` to print variable values and sizes.
  - Using `Reference Variables` (e.g., `x=43`, `y=x`).
  - Performing `Typecasting` (e.g., `(float)m`, `int(n)`).
  - Performing arithmetic operations like `m+n`.
- Status Bar:** At the bottom, it shows the file path as "Tutorial.cpp > tut7.cpp", the line number as "Ln 1, Col 1", and the status as "Spaces: 4, UTF-8, LF, C++, Mac".

EXPLORER ... tut8.cpp

OPEN EDITORS tut8.cpp Tutorial.cpp ...

TUTORIALS 153 Revision Tutorial.cpp M tempCodeRu... M tut1 M tut1.cpp M tut3 M tut3.cpp M tut4 M tut4.cpp M tut5 M tut5.cpp M tut6 M tut6.cpp M tut7 M tut7.cpp M tut8 M tut8.cpp M tut9 M tut9.cpp M tut10 M tut10.cpp M tut11 M tut11.cpp M tut12 M tut12.cpp M tut13 M tut13.cpp M tut14 M tut14.cpp M tut15 M tut15.cpp M tut16 M tut16.cpp

OUTLINE

TIMELINE

```
1 #include<iostream>
2 // #include<iomanip> (setw function)
3 using namespace std;
4 int main(){
5     /* int a=34;
6     // char c='c';
7     cout<<"The value of A was : "<<a<<endl;
8     // cout<<"The value of C was : "<<c<<endl;
9     | a=45;
10    | // c='d';
11    cout<<"The value of A is : "<<a<<endl;
12    // cout<<"The value of C is : "<<c; */
13
14 /* const int a=4;
15    cout<<"The value of A is "<<a;
16    // a=5; */
17
18 // *****Manipulators in C++ *****
19 /*cout<<endl;
20 int a=2,b=12,c=123,d=1234;
21 cout<<"The value of A is : "<<a<<endl;
22 cout<<"The value of B is : "<<b<<endl;
23 cout<<"The value of C is : "<<c<<endl;
24 cout<<"The value of D is : "<<d<<endl;
25 cout<<endl;
26 cout<<"The value of A is : "<<setw(4)<<a<<endl;
27 cout<<"The value of B is : "<<setw(4)<<b<<endl;
28 cout<<"The value of C is : "<<setw(4)<<c<<endl;
29 cout<<"The value of D is : "<<setw(4)<<d<<endl; */
30
31
32 // Operator Precedence
33 int a=3, b=4;
34 // int c=a*5+b;
35 int c= ((a*5)+b)-45)+87;
36 cout<<c;
37 return 0;
38 }
```

EXPLORER ... tut9.cpp x

TUTORIALS 153

Tutorial.cpp > tut9.cpp > ...

```
1 // **** Basic Control Structure ****
2 /* 1. Sequence Structure
3 2. Selection structure
4 3. Loop structure */
5
6 #include<iostream>
7 using namespace std;
8 int main(){
9     // cout<<"This is tutorial 9";
10    int age;
11    cout<<"Tell me your age : ";
12    cin>>age;
13    // **** Selection control structure: if-else if-else ladder
14    /* if((age<18) && (age>0)){
15        cout<<"You cannot come to my party."<<endl;
16    }
17    else if(age==18){
18        cout<<"You will get kid pass in my party."<<endl;
19    }
20    else if(age<1){
21        cout<<"You are yet not born."<<endl;
22    }
23    else{
24        cout<<"You can come to my party."<<endl;
25    }*/
26
27 // Selection control structure: switch case statements
28 switch (age){
29     case 18:
30         cout<<"You are 18 yrs old."<<endl;
31         break;
32
33     case 22:
34         cout<<"You are 22 yrs old."<<endl;
35         break;
36
37     case 2:
38         cout<<"You are 2 yrs child only."<<endl;
39         break;
40
41     default:
42         cout<<"No special cases"<<endl;
43         break;
44     }
45     return 0;
46 }
```

tut10 M tut11 M tut12 M tut13 M tut14 M tut15 M tut16 M tut17 M tut18 M tut19 M tut20 U tut21 U tut22 U

> OUTLINE > TIMELINE

The screenshot shows a dark-themed code editor interface with the following details:

- EXPLORER**: Shows a tree view of open editors. The current file is `tut10.cpp`. Other files listed include `TUTORI...`, `Revision`, `Tutorial.cpp`, `tempCodeRu...`, `tut1`, `tut3`, `tut1.cpp`, `tut3.cpp`, `tut4`, `tut4.cpp`, `tut5`, `tut5.cpp`, `tut6`, `tut6.cpp`, `tut7`, `tut7.cpp`, `tut8`, `tut8.cpp`, `tut9`, `tut9.cpp`, `tut10`, and `tut10.cpp` (which is selected).
- TUTORIALS**: A folder containing various tutorial files.
- OPEN EDITORS**: A list of open files, with `tut10.cpp` currently active.
- CODE**: The main editor area containing the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     /* Loops in C++
5      1.For Loop
6      2.While Loop
7      3.do-while loop */
8
9     // ***** 1.For Loop in C++ *****
10
11    // Syntax for 'forLoop'
12    /* for(initialization; condition; updation){
13        loop body(C++ code);
14    } */
15
16
17    // for (int i = 0; i <40; i++){
18    //     cout<<i<<endl;
19    // }
20
21    // ***** 2.While Loop in C++ *****
22    // Syntax:
23    //While (condition)
24    /* {
25        c++ statemens;
26    } */
27
28    // Printing 1 to 40 using while loop
29    /* int i =1;
30    while(i<40){
31        cout<<i<<"," ;
32        i++;
33    }
34    cout<<i<<".";*/
35
36    // Printing using do while Loop---->Print the initial condition i.e. in below example it will print 1 no matters condition satisfy or not.
37    int i;
38    i=1;
39    do{
40        cout<<i<<endl;
41        i++;
42    }while(i<=9);
43
44 }
```

**STATUS BAR**: Shows the file name `tut10.cpp`, line `Ln 1, Col 1`, and encoding `UTF-8`.

The screenshot shows a dark-themed code editor interface with various toolbars and panels. The main area displays a C++ program titled `tut11.cpp`. The code includes a `for` loop that prints integers from 0 to 8, skipping the value 4 by using a `continue` statement. A note at the bottom specifies printing values 1 to 15 except 6 and 10 using while and do-while loops.

```
//boiler plate taught
#include<iostream>
using namespace std;
int main(){
    for(int i=0; i<=8; i++){
        if(i==4){
            continue;
        }
        cout<<i<<endl;
    }
}
// Q Print 1 to 15 using while Loop except 6 & also same for do while Loop;
```

The left sidebar lists other files in the project, such as `Tutorial.cpp`, `tut1`, `tut3`, `tut4`, `tut5`, `tut6`, `tut7`, `tut8`, `tut9`, `tut10`, `tut11`, `tut12`, `tut13`, `tut14`, `tut15`, `tut16`, and `tut17`. The file `tut11.cpp` is currently selected.

Tutorial.cpp &gt; C tut12.cpp &gt; ...

```
1 // ***** Pointers *****
2 #include<iostream>
3 using namespace std;
4 int main(){
5     // Pointer is a data type which store the address of other data types.By using the pointer we can access the variable and also changes its value.
6
7     int a=3;
8     int * b = &a; //(& ---> mpercent)
9     // & ----> (Address of) operator
10    cout<<endl;
11    cout<<"The adddress of a is "<<&a<<endl;
12    cout<<"The address of a is "<<b<<endl; //currently not understand this line.
13    cout<<endl;
14
15    // * ----> (Value at) Dereference operator
16    cout<<"The value at address b is "<<*b<<endl;
17    *b=16;
18    cout<<"The new value at address b is "<<*b<<endl;
19    cout<<endl;
20    // Pointer to Pointer
21    int** c = &b;
22    cout<<"The address of b is "<<&b<<endl;
23    cout<<"The address of b is "<<c<<endl;
24    cout<<endl;
25    cout<<"The valaue at address c is "<<*c<<endl;
26    cout<<"The value at address_at(value_at(c)) is "<<**c<<endl; //currently not understand this line.
27
28    cout<<endl;
29    return 0;
30 }
```

