

Preprocessor

→ Used with the #define preprocessor command to assign a string to the name.

Ex:

```
#include <iostream>
```

```
# define END 50
```

```
using namespace std;
```

```
int main() → This is called a function.
```

```
{
```

```
cout << " END Number: " << END << "\n";
```

```
return 0; → ends the main function.
```

```
}
```

means that we can use names for objects and variables from the standard library.

→ The cout object, together with the << operator, is used to output values/print text.

↓ If it executes, you will see ~~END~~ on the screen.

```
∴ END Number: 50
```

★ \n

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
cout << "Hello World!";
```

```
cout << "I am learning C++";
```

```
return 0;
```

```
}
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
cout << "Hello World! \n";
```

```
cout << "I am learning C++";
```

```
return 0;
```

```
}
```

Hello World! I am learning C++

Hello World!

I am learning C++

(5)

* endl

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello World!" << endl;
    cout << "I am learning C++";
    return 0;
}
```

Hello World!

I am learning C++

* Both `\n` and `endl` are used to break lines. However, `\n` is used more often and is the preferred way.

Omitting Namespace

The `using namespace std` line can be omitted and replaced with the `std` keyword, followed by the `::` operator for some objects:

```
#include <iostream>
int main() {
    std::cout << "Hello World!";
    return 0;
}
```

Hello World!

* It is up to you if you want to include the standard namespace library or not.

C++ Variables

Variables are containers for storing data values.

In C++, there are different types of variables:

- **int** - stores integers (whole numbers), without decimals, such as 123 or -123
- **double** - stores floating point numbers, with decimals, such as 19.99 or -19.99
- **char** - stores single characters, such as 'a' or 'B'
Char values are surrounded by single quotes
- **string** - stores text, such as "Hello World". String values are surrounded by double quotes
- **bool** - stores values with two states: true or false

Declaring Variables

To create a variable, you must specify the type and assign it a value:

Syntax:

```
type variable = value;
```

Where type is one of C++ types (such as **int**), and variable is the name of the variable (such as x or myName). The equal sign is used to assign values to the variable.

```
int myNum = 15;  
cout << myNum;
```



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

int main() {
    int myNum = 15;
    cout << myNum;
    return 0;
}
```

15

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

int main() {
    int myNum;
    myNum = 15;
    cout << myNum;
    return 0;
}
```

15

Ex:

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

int main() {
    int myNum = 15; // Now myNum is 15
    myNum = 10; // Now myNum is 10
    cout << myNum;
    return 0;
}
```

10

Const int

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

int main() {
    const int myNum = 15; // myNum will always be 15
    myNum = 10; // error: assignment of read-only
                // variable 'myNum'
    cout << myNum;
    return 0;
}
```

Other types

`int myNum = 5;` // integer (whole number without decimals)

`double myFloatNum = 5.99;` // Floating point number (with decimals)

`char myLetter = 'D';` // Character

`string myText = "Hello";` // String (text)

`bool myBoolean = true;` // Boolean (true or false)

# Data type	size	Description
int	4 bytes	Stores whole numbers, without decimals
float	4 bytes	Stores fractional numbers, containing one or more decimals Sufficient for storing 7 decimal digits
double	8 bytes	Stores fractional numbers, sufficient 15 decimal digits
boolean	1 byte	Stores true or false values
char	1 byte	Stores a single character/letter/number


```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main () {
```

```
    // Creating variables
```

```
    int myNum = 5; // Integer (whole number)
```

```
    float myFloatNum = 5.99; // Floating point number
```

```
    double myDoubleNum = 9.98; // Floating point number
```

```
    char myLetter = 'D'; // Character
```

```
    bool myBoolean = true; // Boolean
```

```
    string myString = "Hello"; // String
```

```
    // Print variable values
```

```
    cout << "int: " << myNum << "\n";
```

```
    cout << "float: " << myFloatNum << "\n";
```

```
    cout << "double: " << myDoubleNum << "\n";
```

```
    cout << "char: " << myLetter << "\n";
```

```
    cout << "bool: " << myBoolean << "\n";
```

```
    cout << "string: " << myString << "\n";
```

```
    return 0;
```

```
}
```

int: 5

float: 5.99

double: 9.98

char: D

bool: 1

string: Hello