4.2 — Void

BY ALEX ON FEBRUARY 11TH, 2015 | LAST MODIFIED BY ALEX ON OCTOBER 18TH, 2019

Void is the easiest of the data types to explain. Basically, **void** means "no type"!

Consequentially, variables can not be defined with a type of void:

```
void value; // won't work, variables can't be defined with a void type
```

Void is typically used in several different contexts.

Functions that do not return a value

Most commonly, *void* is used to indicate that a function does not return a value:

```
void writeValue(int x) // void here means no return value
{
    std::cout << "The value of x is: " << x << '\n';
    // no return statement, because this function doesn't return a value
}
```

If you use a return statement to try to return a value in such a function, a compile error will result:

```
void noReturn(int x) // void here means no return value
{
    return 5; // error
}
```

On Visual Studio 2017, this produced the error:

```
error C2562: 'noReturn': 'void' function returning a value
```

Deprecated: Functions that do not take parameters

In C, void is used as a way to indicate that a function does not take any parameters:

```
int getValue(void) // void here means no parameters
{
    int x;
    std::cin >> x;
    return x;
}
```

Although this will compile in C++ (for backwards compatibility reasons), this use of keyword *void* is considered deprecated in C++. The following code is equivalent, and preferred in C++:

```
int getValue() // empty function parameters is an implicit void
{
    int x;
    std::cin >> x;
    return x;
}
```

Best practice

Use an empty parameter list instead of *void* to indicate that a function has no parameters.

Other uses of void

The void keyword has a third (more advanced) use in C++ that we cover in section **6.13 -- Void pointers**. Since we haven't covered what a pointer is yet, you don't need to worry about this case for now.

Let's move on!

 **4.3 -- Object sizes and the sizeof operator**

 **Index**

 **4.1 -- Introduction to fundamental data types**

C++ TUTORIAL | PRINT THIS POST

42 comments to 4.2 — Void

Wallace
October 14, 2019 at 2:31 pm · Reply

Minor typo: "Most commonly, void *as* a way to indicate that a function does not return a value" seemingly should be "Most commonly, void *is* a way to indicate that a function does not return a value."

Thanks for the great site!

Alex
October 18, 2019 at 2:41 pm · Reply

Typo fixed. Thanks!

Edwin Martens
August 21, 2019 at 6:45 am · Reply

as far as I know, there IS a variable that can be of the type void... a Pointer !
void * pPointer;

nascardriver
August 21, 2019 at 7:09 am · Reply

That's a `void` pointer, not `void`. If you try to turn it into `void` by indirection (`*pPointer`), you won't get `void`, but an error.

Edwin Martens
August 21, 2019 at 10:41 pm · Reply

that's true.. ;-)

Jeroen P. Broks
February 17, 2019 at 1:35 am · Reply

Is this considered valid in C++?

```
1  void myFunc(int a){
2       if (a<0) { return; }
3       std::cout << a;
4  }
```

(I've done this a lot in other languages, and that would have a 'return' statement inside a void). :-P

nascardriver
February 17, 2019 at 6:24 am · Reply

Hi!

Yes. @return is totally fine to use in voids.

Fateh Chadha
September 21, 2018 at 8:54 am · Reply

Hi Alex..!!
//
void writeValue(int x) // void here means no return value
{
  std::cout << "The value of x is: " << X << std::endl;
  // no return statement, because the return type is void
}
//

In this example above... since void has no return, where do we get the value of capital X from.
While making a calculator, in void printValue() function.. only thing I couldn't understand was how this x is interpreted in this function. If u could clear this doubt it would be a great help.

nascardriver
September 21, 2018 at 9:00 am · Reply

Hi Fateh!

There is no capital x in this lesson. Lower case @x is a parameter of @writeValue (Lesson 1.4a).

Qais
December 13, 2018 at 10:27 am · Reply

Hello , I think you should go back to chapter one and learn back the parameters.Its always bad idea to skip because programming is complicated.

The answer to your question would be that. X is parameter. Parameters take his value from caller . That means whichever functions call the void writevalue(int x). It will initializate the value to X.

```
void writeValue(int x)
{
   std::cout << "The value of x is: " x << std::endl;
}
int main()
{
  writeValue(5);  // The caller is assigning value to the function.  Thats why we use parameters.
  return 0;
}
```

Aditi
June 14, 2018 at 12:33 am · Reply

Hello ! Just a small doubt, if the words like void , int , are used to specify data types of variables, why do we also use them before names of functions like int main (), void main(), etc.? Does the function int main () mean that the function will only accept integer values and return integer values ?

As a side note , thanks a ton for for this amazing website ! This is my only source of learning C++

nascardriver
June 15, 2018 at 7:04 am · Reply

Hi Aditi!

It's the function's return type.

References
* Lesson 1.4 - A first look at functions and return values

Aditi
June 15, 2018 at 8:50 am · Reply

Hello nascardriver

Yes I now remember! Thank you !

Linyuan
February 5, 2018 at 4:23 pm · Reply

can't imagine that this site was already there in 2007!

know this in 2018...it's just tooooooooooooo late!!!

if i met this website much more earlier all things would be totally different...

thanks again!

Alex
February 6, 2018 at 1:05 pm · Reply

It's never too late to create something amazing.

Sihoo
November 5, 2016 at 12:49 pm · Reply

The only site I am more than happy to disable AdBlock.

Kevin ☺
May 28, 2017 at 7:07 pm · Reply

same

SomeSHET
March 27, 2018 at 9:28 am · Reply

I just realized mine was on. I am disabling it now for this website from now on.

Brian Gaucher
May 1, 2018 at 12:23 pm · Reply

I've disabled my adblock for the media.net advertising. But I still block Google ads. Not to block ads, I'd be more than happy to see ads on this website. But due to Google's privacy breach, I disable only Google ads.

Tanay Jaiman
October 9, 2016 at 8:44 am · Reply

can someone check this code for me.

I am not able to compile and run it, if someone can compile and run it and reply me with the output I will be very thankful.

Thanks in advance.
Best Regards,
Tanay Jaiman

```
#include <iostream>
 int main()
 {
     using namespace std;
     int n1;
     int n2;

     //input number 1
     cout << "Please input number 1." << endl;
     cin >> n1;

     //line gapping
     cout <<"n" <<endl;

     //input number 2
     cout << "Please input number 2." << endl;
     cin >> n2;

     string operation;
     cout << "Please input operation in form of symbols like + for addition, - for subtraction, * for multiplication and / for division."
     cin >> operation;
```

```
22
23      cout << "n"<< endl;
24
25      //Final output
26      if (operation = "+")
27          cout << "The sum of the given number should be:- " << n1 + n2 << endl;
28      if (operation = "-")
29          cout << "The difference of the given numbers should be:- " << n1-n2 << endl;
30      if (operation = "*")
31          cout << "The product of the given numbers should be:- " << n1*n2 << endl;
32      if (operation = "/")
33          cout << "The division of the given numbers should be:- " << n1/n2 << endl;
34      else
35          cout << "Error 504, input not found, Try Again." << endl;
36          goto int n1;
37
38      return 0;
39
40  }
```

Lyle
October 12, 2016 at 9:17 pm · Reply

Lines 13, 23 just print the letter "n", is that what you intended ?

Line 20 needs a terminating ";" at the end.

Your "operation = " set each need to use "==" to do comparisons.

I commented out the "goto" because you never setup a "label:" for it to work. You can't arbitrarily pick a place for code to go-to, it's more involved than that (if you even should be using a goto, which is unlikely).

Hope that helps you along.

McSteven
June 12, 2017 at 6:41 am · Reply

And you would use single quotes for checking if the operator is legal. Consequently, you would declare the variable 'operator' with a char type.

Georges Theodosiou
September 13, 2016 at 12:39 am · Reply

Mr Alex good morning (I live in France),

Please let me express my many thanks for your reply.

With regards and friendship

P.S. I have problem connecting your site.

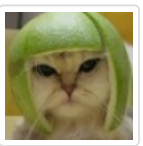
Georges Theodosiou
September 10, 2016 at 12:57 am · Reply

Mr Alex, good morning,

Please, let me ask you the complete program for:

```
int getValue() // empty function parameters is an implicit void
{
   int x;
   std::cin >> x;
   return x;
}
```

With regards and friendship



Alex
September 12, 2016 at 8:44 am · Reply

```
#include <iostream>
int getValue() // empty function parameters is an implicit void
{
    int x;
    std::cin >> x;
    return x;
}

int main()
{
    std::cin << "Enter a value: ";
    int x = getValue();
    std::cout << "You entered: " << x << std::endl;

    return 0;
}
```

Michael
April 29, 2016 at 11:41 pm · Reply

Hi I want to ask you something about void. I'm practicing making program that adds 2 to any number the user inputs and it works fine. I separate it to two files. One is for the main and one is for the functions. It all works fine but I read in this article that you can't declare a void, right? Because void means no type. But I forward declare a void and nothing's wrong the program runs. I'm just curious tho

Alex
April 30, 2016 at 12:22 pm · Reply

You can't declare a variable of type void, but you can declare a function as returning void (meaning it doesn't return anything).

Milo
April 8, 2016 at 3:53 am · Reply

Hi Alex
Please can you advise
I'm dealing with a C++ question and it states that the function should calculate the sum of two parameters passed by value and then store the result in the first variable passed by reference. It should calculate the difference between the two parameters passed by value and then store the result the second paramete passed by reference

Alex
April 9, 2016 at 11:14 am · Reply

What do you need to be advised about with regards to this question?

Reference are covered in lessons 6.11 and 7.3.

Zigzem30
February 7, 2016 at 7:43 am · Reply

Hi, I was making a program where it finds the circumference of a circle. I looked online but I can't find out how to use augments to do a function call to circumference in void. Thanks for any help!

```
#include "stdafx.h"
#include <iostream>
#include "myconstans.h"
double getDiameter() {
    std::cout << "Please enter the diameter of the Circle" << std::endl;// user is suppose
d to already have the diameter
    double diameterOfCircle;
    std::cin >> diameterOfCircle;
    return diameterOfCircle;
}
double calculateCircumference(double diameterOfCircle) {
    double circumfrence(equation::pi * diameterOfCircle); // equation for circumference is
pi * diameter
    return circumfrence;
}
void printAnswer(double circumfrence) {
    std::cout << "The circumfrence of the circle is " << circumfrence << std::endl;
}
int main()
{
    double diameter = getDiameter();
    double calculateCircumference(double s);

    printAnswer(double g);//i've tried just "g" in the augument, but it says that g is und
efined
    return 0;
}
```

Alex
February 8, 2016 at 6:17 pm · Reply

Your helper functions look good, but your main() is messed up.

Try this:

```
int main()
{
    double diameter = getDiameter();
    double circumference = calculateCircumference(diameter);
    printAnswer(circumference);
    return 0;
}
```

Zigzem30

February 9, 2016 at 2:28 pm · Reply

Thank You!

smail messaoudi

September 20, 2015 at 9:46 am · Reply

hi Alex
I'm confused about

```
int add(int x, int y)
{
 return x+y;
}
```

and

```
void add(int x,int y)
{
return x+y;
}
```

in this case the keyword void doesn't return any value or 0

and

```
#include <iostream.h>
int main()
{
   using namespace std;
int add(int x, int y)
{

 return x+y ;
}

   system("PAUSE");
   return 0;//could you explain to me 'return 0'

}
```

I have fun with your Tutorial Alex
thanks

Alex

September 20, 2015 at 5:41 pm · Reply

A function with a void return value will not return anything to the caller.

For example:

```
1 void add(int x,int y)
2 {
3 return x+y; // this is illegal, a void function can't return a value
4 }
```

But this would be okay:

```
1 void add(int x,int y)
```

```
{
std::cout << x+y; // this is okay, nothing is returned
}
```

Function main() returns a value to the operating system indicating whether the program ran succesfully or not. A return value of 0 means success. A return value otherwise indicates a failure.

Vex
August 29, 2015 at 4:20 am · Reply

```
#include <iostream>

/** void
 *
 * 1) As a way to indicate that a function does not return a value
 * 2) As a way to indicate that a function does not take any parameters:
 *
 *
 */


// int getNumber(void)  // c
int getNumber()         // c++
{
    std::cout << "Please input a number : ";
    int x;
    std::cin >> x;
    return x;

}

void printNumber(int x)
{
    std::cout << "number is : " << x << std::endl;
}

int main()
{
    int x = getNumber();
    printNumber(x);

    return 0;
}
```

Ian
July 20, 2015 at 8:37 am · Reply

I am still confused it is so weird like you can't declare an variable and I heard about that they can write functions like iostream

R4Z3R
May 18, 2015 at 9:07 pm · Reply

Hello.
Why in the function `main` we use void?

```
int main(void){}
```

what is the different between `main(void)` and `main()` ?

Alex
May 19, 2015 at 8:28 am · Reply

The following two function declarations are identical in C++:

```
int main(); // preferred in C++
int main(void);
```

Jeroen
October 11, 2015 at 5:13 am · Reply

In practice, you may not do that with the main function.
Windows or what o.s. than also expected a value back.

For non-pointer success/fail:

```
return 0;     // success
return -1;     // fail
```

For pointer functions:

```
NULL => fail
everything else => success
```

Neeraj
April 3, 2015 at 8:31 pm · Reply

Does C++ retain the function property in C- "if no return type is declared, it will be assumed to be 'int' by the compiler"?

deltron zero
April 5, 2015 at 3:10 pm · Reply

As far as I can tell (I barely started learning here), it does so with main(), but you'll get an error when you compile if you don't declare a return type for any other function.

Alex
April 8, 2015 at 8:23 am · Reply

No. The C++ standard says that a function missing a return type is ill-formed.

That said, some compilers that do not strictly adhere to the C++ standard will still assume integer, presumably for compatibility with C. It looks like Visual Studio allows you to omit the return type for main(), but not other functions.

That said, you should always declare a return type for functions in C++, even if your compiler allows you to do otherwise.