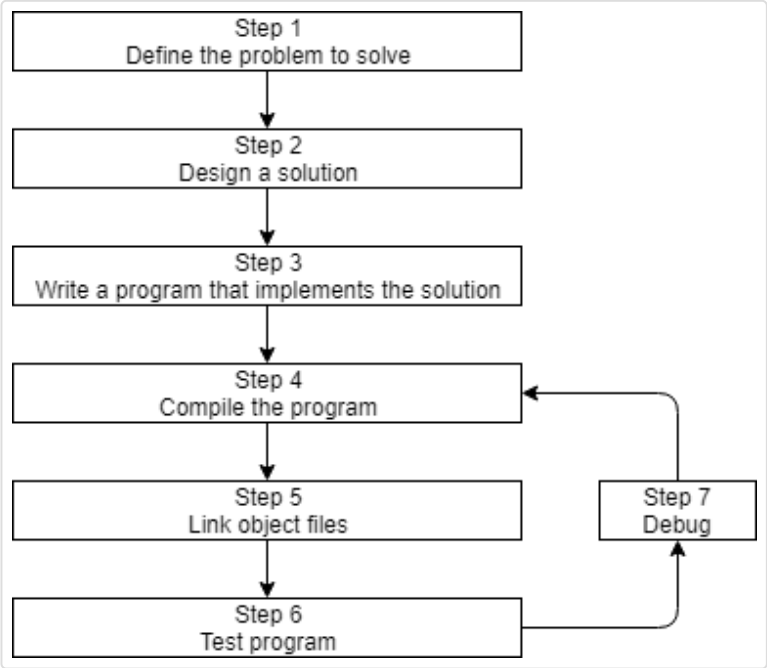
0.4 — Introduction to C++ development

BY ALEX ON MAY 27TH, 2007 | LAST MODIFIED BY ALEX ON FEBRUARY 9TH, 2020

Before we can write and execute our first C++ program, we need to understand in more detail how C++ programs get developed. Here is a graphic outlining a simplistic approach:



Step 1: Define the problem that you would like to solve

This is the “what” step, where you figure out what problem you are intending to solve. Coming up with the initial idea for what you would like to program can be the easiest step, or the hardest. But conceptually, it is the simplest. All you need is an idea that can be well defined, and you’re ready for the next step.

Here are a few examples:

- “I want to write a program that will allow me to enter many numbers, then calculates the average.”
- “I want to write a program that generates a 2d maze and lets the user navigate through it. The user wins if they reach the end.”
- “I want to write a program that reads in a file of stock prices and predicts whether the stock will go up or down.”

Step 2: Determine how you are going to solve the problem

This is the “how” step, where you determine how you are going to solve the problem you came up with in step 1. It is also the step that is most neglected in software development. The crux of the issue is that there are many ways to solve a problem -- however, some of these solutions are good and some of them are bad. Too often, a programmer will get an idea, sit down, and immediately start coding a solution. This often generates a solution that falls into the bad category.

Typically, good solutions have the following characteristics:

- They are straightforward (not overly complicated or confusing).
- They are well documented (especially around any assumptions being made or limitations).

- They are built modularly, so parts can be reused or changed later without impacting other parts of the program.
- They are robust, and can recover or give useful error messages when something unexpected happens.

When you sit down and start coding right away, you're typically thinking "I want to do <something>", so you implement the solution that gets you there the fastest. This can lead to programs that are fragile, hard to change or extend later, or have lots of **bugs** (technical defects).

As an aside...

The term *bug* was first used by Thomas Edison back in the 1870s! However, the term was popularized in the 1940s when engineers found an actual moth stuck in the hardware of an early computer, causing a short circuit. Both the log book in which the error was reported and the moth are now part of the Smithsonian Museum of American History. It can be viewed **here**.

Studies have shown that only 20% of a programmer's time is actually spent writing the initial program. The other 80% is spent on maintenance, which can consist of **debugging** (removing bugs), updates to cope with changes in the environment (e.g. to run on a new OS version), enhancements (minor changes to improve usability or capability), or internal improvements (to increase reliability or maintainability).

Consequently, it's worth your time to spend a little extra time up front (before you start coding) thinking about the best way to tackle a problem, what assumptions you are making, and how you might plan for the future, in order to save yourself a lot of time and trouble down the road.

We'll talk more about how to effectively design solutions to problems in a future lesson.

Step 3: Write the program

In order to write the program, we need two things: First, we need knowledge of a programming language -- that's what these tutorials are for! Second, we need an editor. It's possible to write a program using any editor you want, even something as simple as Window's notepad or Unix's vi or pico. However, we strongly urge you to use an editor that is designed for coding. Don't worry if you don't have one yet. We'll cover how to install a code editor shortly.

A typical editor designed for coding has a few features that make programming much easier, including:

1) Line numbering. Line numbering is useful when the compiler gives us an error, as a typical compiler error will state: *some error code/message, line 64*. Without an editor that shows line numbers, finding line 64 can be a real hassle.

2) Syntax highlighting and coloring. Syntax highlighting and coloring changes the color of various parts of your program to make it easier to identify the different components of your program. Here's an example of a C++ program with both line numbering and syntax highlighting:

```
1  #include <iostream>
2
3  int main()
4  {
5      std::cout << "Colored text!";
6      return 0;
7  }
```

The examples we show in this tutorial will always have both line numbering and syntax highlighting to make them easier to follow.

3) An unambiguous font. Non-programming fonts often make it hard to distinguish between the number 0 and the letter O, or between the number 1, the letter l (lower case L), and the letter I (upper case i). A good programming font will differentiate these symbols in order to ensure one isn't accidentally used in place of the other.

In C++ your programs will typically be called *name.cpp*, where *name* is replaced with the name of your choosing for the program (e.g. calculator, hi-lo, etc...). The *.cpp* extension tells the compiler (and you) that this is a C++ source code file that contains C++ instructions. Note that some people use the extension .cc instead of .cpp, but we recommend you use .cpp.

Best practice

Name your code files *name.cpp*, where *name* is a name of your choosing, and *.cpp* is the extension that indicates the file is a C++ source file.

Also note that many complex C++ programs have multiple .cpp files. Although most of the programs you will be creating initially will only have a single .cpp file, it is possible to write single programs that have tens or hundreds of .cpp files.

Once we've written our program, the next steps are to convert it into something that we can run, and then see whether it works! We'll discuss those steps (4-7) in the next lesson.

 **0.5 -- Introduction to the compiler, linker, and libraries**

 **Index**

 **0.3 -- Introduction to C/C++**

C++ TUTORIAL | PRINT THIS POST

148 comments to 0.4 — Introduction to C++ development

« Older Comments 1 2 3

kabrunko
February 3, 2020 at 1:29 pm · Reply

Good evening,

I don't know if it's an error of writing, but it's about this phrase: “I want to do _this_” at step 2, third paragraph. Isn't suppose to be in bold letters?

If don't, please ignore this message.

Best regards.

Julian
January 5, 2020 at 11:37 pm · Reply

I think the first code of your is so cool, not using namespace is awesome.

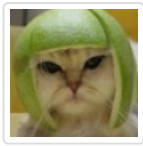
Xumora
December 29, 2019 at 12:33 am · Reply

I think so, it is good!
Thank you !

Vitaliy Sh.
December 19, 2019 at 5:12 am · Reply

...
First we need knowledge of a programming language
...

Possible typo: no comma after "First".



Alex
December 19, 2019 at 2:08 pm · Reply

I believe it's correct as written (see http://writingresources.benchmarkeducation.com/pdfs/Gr3_U4_Mini20.pdf).

Vitaliy Sh.
December 19, 2019 at 9:25 pm · Reply

When i open (just to see the main site, after downloading mini-lesson): http://writingresources.benchmarkeducation.com/

"As of June..." blinks and then i see the page with links. If i click any i'm see "This Connection is Not Secure" at https links.
Can your, please check if that true for your to?Just my ISP sometimes make "magic", so i not sure if i need to bother the "Benchmark education" tech support.

Vitaliy Sh.
December 19, 2019 at 9:56 pm · Reply

"First we need" - "mainly", not "step 1"
"knowledge ..." - subj.

"Second," - after considering "First"
"we need an editor." - enumeration.

If a sentence were "First, we need to acquire the knowledge ...", then that were procedural, with comma.

PS: comrades pupils, full list is at (as a time of writing, that is running Debian):
http://writingresources.benchmarkeducation.com/pdfs/

Vitaliy Sh.
December 19, 2019 at 2:26 am · Reply

About "As an aside..."
I've visited:
https://americanhistory.si.edu/collections/search/object/nmah_334663

The "Log Book With Computer Bug" is available online, but is "Currently not on view". Can your, please, add to the "As an aside..." lesson's part a link to the museum's online view?

Alex
December 19, 2019 at 1:53 pm · Reply

Cool! Thanks for the link. I added it to the aside box.

Vitaliy Sh.
December 18, 2019 at 11:06 pm · Reply

I think that "I want to write a program" repeated in every bullet point is draws attention from examples themselves. How about that:

```
<ul>
<h4>"I want to write a program ..."</h4>
<li>“that will allow me to enter many numbers, then calculates the average.”</li>
<li>“that generates a 2d maze and lets the user navigate through it.  The user wins if they
<li>“that reads in a file of stock prices and predicts whether the stock will go up or down.
</ul>
```

Also (proposal):

```
<p>About good solutions we have to say that, typically, they are:</p>
<ul>
<li>straightforward<br>(not overly complicated or confusing)
</li>
<li>well documented<br>(especially around any assumptions being made or limitations)</li>
<li>built modularly<br>(so parts can be reused or changed later without impacting other part
<li>robust, and can recover or give useful error messages when something unexpected happens.
</ul>
```

Vitaliy Sh.
December 18, 2019 at 6:40 pm · Reply

Hi!
Is the full stops at the end of the Step 1, and Step's 2 section headings are intended?

The Web speaks that this is either according to writer's tastes, or style guide. So i wonder, if that because of word "you" inside of them.

Alex
December 19, 2019 at 1:47 pm · Reply

Do you mean the fact that the steps are separated by a section divider or something else?

Vitaliy Sh.
December 19, 2019 at 2:53 pm · Reply

Step 1: Define the problem that you would like to solve.

Step 2: Determine how you are going to solve the problem.

No, sir, i mean the full stops themselves. Other headings didn't have them, and so far i'm was unable to find out what makes the difference.
Section divider (<hr>, "ornamented"?) is fine.

Vitaliy Sh.
November 8, 2019 at 4:49 am · Reply

... In order to write the program, we need two things: First ...

Maybe a typo: capital letter F in 'first ...'.

Alex
November 13, 2019 at 1:24 pm · Reply

Capitalizing the letter following a colon is acceptable grammar if the sentence following the colon is a complete sentence (which in this case, it is).

Vitaliy Sh.
November 8, 2019 at 4:42 am · Reply

...Consequently, it’s worth your time to spend a little extra time up front before you start coding thinking about the best way to tackle a problem,...

Maybe better insert two '-' there:
... time up front - before you start coding - thinking about...
?

Alex
November 13, 2019 at 1:23 pm · Reply

Fixed, thanks!

BurstDev
October 19, 2019 at 2:52 am · Reply

I think it's necessary to provide namespace identification

kapilaithran
July 24, 2019 at 7:52 am · Reply

why do you give["return 0;"]
and [std::]
.
.
.
you can directly give ...
[#include<iostream>
main()
{
cout<<"colored text!";
}]

is there any purpose for it

nascardriver
July 24, 2019 at 7:57 am · Reply

`using namespace std;` can cause name collisions.
The return should be added for consistency.

Chayim Eliazer
May 12, 2019 at 5:46 pm · Reply

This is the only website and out of all pdf's of programming that explains everything and step by step the way it should be!!

Oppong Richard
January 18, 2019 at 3:46 pm · Reply

Thank you so much
This tutorial is very comprehensive and all what you need as a beginner programmer like me.

Shehu Bello
November 10, 2018 at 6:50 am · Reply

I was actually reading a book on c++ when I came across "composition" which the author did not explain extensively. So I decided to browse for the topic online and finally found myself in your website. In fact, your teaching style is so mesmerising that I have to abandon the book and continue with your tutorials.

Thanks a lot

André
November 1, 2018 at 11:13 am · Reply

Youy way to teach is really awesome!
Thank you!!

Caroline
October 18, 2018 at 7:34 pm · Reply

Thank you very much for this lovely initiative!

shreyam
October 12, 2018 at 1:47 am · Reply

sir,
If you have mentioned above that c++ is a compiled language , then why i need to compile my program before running it.

Alex
October 12, 2018 at 12:01 pm · Reply

The definition of a compiled language is that the code needs to be compiled by a compiler before it can be executed by the computer.

Kevin Rommens
August 4, 2018 at 8:18 am · Reply

Can you make "Dive inn" Topics that we can read from? I need a quick tutorial on how to make a declaratoin.

Alex
August 7, 2018 at 12:13 pm · Reply

You can use the site index to find specific topics.

Kevin Rommens
August 8, 2018 at 3:59 pm · Reply

:thinking:
Well. I'm trying to get how to make a declaration from scratch.
And I don't get it, even if I try my hardest.
So maybe give people some tips on how to get topics when I read something and I don't get it?

Alex
August 13, 2018 at 7:59 pm · Reply

Find the article on declarations (using the site index), and leave a comment there with what you're trying to do, and what you've tried that didn't work. I'm sure some nice soul will help.

vinayak ruhela

April 7, 2018 at 9:22 pm · Reply

sir i have some doubt in "what and how" section
if my what section says that " I want to create a program which remove a particular type of files from my pc so what "How section" consist
please give me some more examples

knelse
April 10, 2018 at 7:36 am · Reply

Hi,
Some of the "how" examples can be:
1. (naive, slow) Find all the files on your PC, then check for those you want to remove and remove them.
2. (pro-active) Make a system service that would look for file creation events and trigger removal program when specific file type is created.
3. (somewhat strange) Modify file system code to make that file type work as if it was free space.
4. (resembling actual solution development) Limit possible folders/locations to reasonable ones (for example, you're probably not looking for .pdf in Windows DLL folders), find out the best time to run such program (for example, if you need to delete temp files that some software creates once a month, you don't need to run removal every second), think about speed and concurrency issues (what happens if you delete the file that your operating system is currently trying to read? Is it a problem for everything else on your PC if you try to delete 1M files simultaneously?), then check for theoretically possible bugs, then implement solution 1 with all the data you collected.

Iman Rosignoli
March 27, 2018 at 2:26 am · Reply

I work in a unix environment with Mac OS X.
How can i execute step 5 if the terminals gives back this message:
<< MacBook-Air-di-Iman:calcoloipotenusa iman.rosignoli$ g++ -o prog 2numeri.o file.o
duplicate symbol _main in:
2numeri.o
file.o
ld: 1 duplicate symbol for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation) >>
In syntesis, there's this message of "duplicate symbol _main"
Can you explain it to me in order to fix the files to execute step 5?
Good day.

nascardriver
March 27, 2018 at 3:36 am · Reply

Hi Iman!
Can you show us the source code you're trying to compile?

Alex
March 30, 2018 at 4:01 pm · Reply

It looks like the code files for 2numeri and file both have a function named main.
A program can only have one main function.

Oneslayed
February 8, 2018 at 7:03 pm · Reply

Is it possible to create your own coding engine? (e.g. Java, Python, C++, etc. etc.)

Alex
February 13, 2018 at 5:12 pm · Reply

What's a "coding engine"?

If you mean programming language, sure. You can write your own language, develop a parser that converts it into C++ (or some other language), and then use an existing compiler for that language to compile it.

If you mean compiler, yes, but this is a lot more complicated. I wouldn't even try to do this.

TheMightyChuy!
February 6, 2018 at 8:41 pm · Reply

Hi there Mr. Alex, first of all, many thanks for creating these lessons. I have faith that they will turn out very useful (i.e. They will actually work). However, I have a yearning question. Why shouldn't we use .cc as a file extension? What is the difference between .cc and .cpp. Please tell us, we are curious minds.

*raises right eyebrow and gives sinister smile

Alex
February 8, 2018 at 3:22 pm · Reply

Because the website is named learncpp.com, not learncc.com.

I kid, I kid.

There's really no difference. If you want to use .cc, you can. Just be consistent.

Matthew
December 31, 2017 at 11:28 am · Reply

I started learning, or trying to learn c++ like a week ago and I started with this one site, that I won't name but it really sucked. Anyways so far I'm enjoying this one but before I start learning to code in c++ do I need to learn HTML first? Also is using ideone just as good as an IDE, or would you prefer the IDE?

nascardriver
January 1, 2018 at 3:29 am · Reply

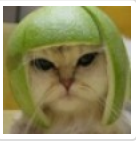
Hi Metthew!
HTML has nothing at all to do with c++, you can start with c++.
I have no experiences with ideone, judging by a quick glance it doesn't seem to let you choose the compiler or compiler options for your program and I don't see any file management. I'd go with a regular IDE.

Linux: eclipse or JetBrains CLion, I prefer eclipse
Windows: Visual Studio

There are many other IDEs around but these are the major ones which you'll find the most help for should you encounter any problems.

Matthew
January 2, 2018 at 1:09 pm · Reply

Yea I actually went with the free visual studio 2017 and so far I love it. Thanks to whoever started this site and took their time to actually teach beginners on how to step by step. It means a lot and I'll learn way more here with it being hands-on than I ever would've in some classroom. I just hope I'll be able to learn enough to get a job with it...This is something I've been wanting to do for a long time and I now have an opportunity to do it...

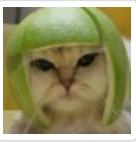


Alex
January 4, 2018 at 1:45 pm · Reply

You definitely don't need to know HTML.
Ideone is an online compiler, and looks pretty limited. It will work for simple programs, may not be able to support multiple files or robust debugging. I'd install an IDE application if that's an option for you.

ayyubali
November 27, 2017 at 1:22 am · Reply

How can i do steps in 0.4



Alex
November 28, 2017 at 5:49 pm · Reply

We cover this in future chapters. This just gives you an outline for the process in general.

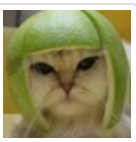
Prabhat Kumar Roy
November 14, 2017 at 8:39 am · Reply

Sir!
I wanted to know if I could program in C++ without using any IDE.

Best regards.

Prabhat Kumar Roy
November 14, 2017 at 5:22 pm · Reply

I do use Windows 7



Alex
November 14, 2017 at 11:24 pm · Reply

Yes, you can use any editor you like. But IDE's have several functions that make them better (namely, an integrated compiler, syntax highlighting, and integrated debuggers).

Prabhat Kumar Roy
November 10, 2017 at 2:18 am · Reply

Dear Sirs!

Sub: Tutorial on C++

Re: Lesson 0.4 -Step: 3

I have learnt HTML and CSS online from html.net where they have said(and I also do opine same):

"
What is needed?

Most likely you already have everything you need.

You have a "browser." A browser is the program that makes it possible to browse and open websites. Right now you are looking at this page in your browser.

It is not important which browser you use. The most common are Google Chrome, Firefox, and Internet Explorer. But there are others such as Opera and Safari and they can all be used and they are all free.

You might have heard about, or even used, programs such as Microsoft FrontPage, Macromedia Dreamweaver or even Microsoft Word, which can — or claim that they can — create websites for you. Forget these programs for now! They are not of any help to you when learning how to code your own website.

Instead, you need a simple text editor. If you are using Windows you can use Notepad.

The problem with many of the programs that claim they can create websites is that they have a lot of standard functions that you can choose from. The downside is that everything needs to fit into these standard functions. Thus, this type of programs often cannot create a website exactly as you want it. Or — even more annoying — they make changes to your hand-written code. With Notepad or other simple text editors, you only have yourself to thank for your successes and errors.

"
I started learning from your tutorial today only with great enthusiasm and liking it and enjoying it; because I myself is the proof of the system(way of learning anything).

Still I would appreciate your comment in the context.

Thank you.

Best regards.
[N.B.: I have given my website (hosted for FREE) which i have done myself based on their tutorial. I have seen that you also offer HTML Tutorial which I will learn to enrich my knowledge further.]

Alex
November 12, 2017 at 3:29 pm · Reply

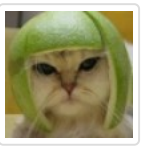
I'm not sure what you would like me to comment on.

With websites, there are many programs that exist to help people layout website visually without having to write HTML and CSS. These programs generate the code for you. However, if you want the best level of control, you need to write the HTML and CSS yourself.

The C++ core language doesn't include visual elements. So in these tutorials, you'll write all of your own code.

Phillip M.
September 4, 2017 at 8:33 am · Reply

Would it be advisable to try designing the solution by writing in a different language (e.g. Python) than what you intend to use for the ACTUAL solution, so you can see how it completely fails?



Alex
September 5, 2017 at 9:48 am · Reply

That sounds like a lot of work for a questionable benefit. So no, I don't think I'd advise that.

If you're interested in making sure your solutions are robust and error-free, I'd focus your time on learning how to _really_ test your code well. Lesson 5.11 has some tips in this regard.

« Older Comments 1 2 3