

# C-and-C++/C2/First-C-Program/English

From Script | Spoken-Tutorial

**Title of script:** First C program

**Author:** Ashwini Patil

**Keywords:** C Program, printf(), int main(), Video Tutorial

| Visual Cue                         | Narration   |
|------------------------------------|---|
| Slide 1                            | Welcome to the spoken tutorial on First C program.  |
| Slide 2<br>Learning Objectives     | In this tutorial we will learn,<br><br>How to write a simple C program<br><br>How to compile it<br><br>How to execute it<br><br>We will also explain some common errors and their solution.   |
| Slide 3<br>System Requirements     | To record this tutorial, I am using<br><br>Ubuntu operating system version 11.10<br><br>and GCC Compiler version 4.6.1 on Ubuntu.   |
| Slide 4<br>Prerequisites           | To practice this tutorial,<br><br>You should be familiar with Ubuntu Operating System.<br><br>And an editor.<br><br>Some editors are vim and gedit<br><br>I will use gedit in this tutorial.<br><br>For relevant tutorials please visit our website which is as shown:<br><br><a href="http://spoken-tutorial.org">http://spoken-tutorial.org</a> ( <a href="http://spoken-tutorial.org/">http://spoken-tutorial.org/</a> ) |
| /*Switch to Terminal*/             | Let me tell you how to write a C program through an example   |
| Open the terminal using Ctrl + Alt | Open the terminal Window by pressing <b>Ctrl, Alt</b> and <b>T</b> keys   |

|  |  |
|--|--|
| <p>+ T or alternately,</p> <p>Click on Applications-&gt; Accessories -&gt; Terminal</p> <p>At the command prompt type: gedit firstprog.c</p> <p>Type:</p> <p><b>//My first C program</b></p> | <p>Simultaneously on your keyboard.</p> <p>Now lets open the text editor, So at the prompt type:</p> <p><b>"gedit" space "talk" dot "c" space &amp;(ampersand sign)</b></p> <p>We used the ampersand(&amp;) to free up the prompt</p> <p>Please note that all the <b>C</b> files will have extension dot <b>"c"</b></p> <p>Now Press <b>Enter</b>.</p> <p>The text editor has opened.</p> <p>Let us start to write a program.</p> <p>Type double slash <b>"//"</b> space</p> <p><b>"My first C program"</b>.</p> |
| <p><b>//My first C program</b></p>   | <p>Here, double slash is used to comment the line.</p> <p>Comments are used to understand the flow of program.</p> <p>It is useful for documentation.</p> <p>It gives us information about the program.</p> <p>The double slash is called as <b>single line comment</b>.</p>   |
| <p>Highlight //</p> <p><b>#include &lt;stdio.h&gt;</b></p>   | <p>Now press <b>Enter</b></p> <p>Type "hash (#) <b>include</b> (space) opening bracket, closing bracket.</p> <p>It is always a good practice to complete the brackets first, and then start writing inside it.</p> <p>Now, Inside the bracket, type:</p> <p><b>"stdio" dot "h"</b></p> <p><b>stdio.h</b> is a <b>header file</b></p> <p>A program must contain this header file when it uses standard input/output functions</p> <p>Now press <b>Enter</b></p>   |

|  |   |
|--|---|
| <pre>#include&lt;stdio.h&gt;  int main()</pre>     | <p>Type: "<b>int</b> (space) <b>main()</b>" (opening bracket and closing bracket)</p> <p><b>main()</b> is a special function.</p> <p>It denotes that the execution of the program begins from this line.</p> <p>The opening bracket and closing bracket is called as paranthesis.</p> <p>Paranthesis followed by <b>main()</b> is to tell the user that main is a function.</p> <p>Here the <b>int main()</b> function takes no arguments. It returns a value of type integer.</p> <p>We will learn about data types in another tutorial.</p> <p>Now let us switch to the slides to know more about the <b>main()</b> function.</p> <p>Let us go tot he next slide.</p> |
| <p>Slide 5</p>                                     | <p>Every program should have one <b>main</b> function.</p> <p>There should NOT be more than one main function.</p> <p>Otherwise, the compiler cannot locate the beginning of the program.</p> <p>The empty pair of parentheses indicate that main has no arguments.</p> <p>The concept of arguments will be discussed in detail in the upcoming tutorials.</p>  |
| <pre>#include&lt;stdio.h&gt;  int main() {</pre>   | <p>Now let us come back to our program,</p> <p>Press <b>Enter</b></p> <p>type: { (opening curly bracket)</p> <p>The opening curly bracket marks the beginning of the function <b>main</b>.</p>  |
| <pre>#include&lt;stdio.h&gt;  int main() { }</pre> | <p>Then</p> <p>Type: } (Closing curly bracket)</p> <p>The Closing bracket indicates the end of the function <b>main</b>.</p>  |
|  | <p>Now inside the bracket</p> <p>press enter twice, move the cursor one line up</p>   |

|  |   |
|--|---|
|  | <p>Indentation makes the code easier to read</p> <p>It also helps to locate errors faster</p> <p>So let us give three space here.</p>   |
| <pre>#include&lt;stdio.h&gt;  int main() { printf("Talk To a Teacher \n");</pre>             | <p>And Type:</p> <p><b>printf</b> opening bracket closing bracket ()</p> <p><b>printf()</b> is a standard <b>C</b> function to print the output on the terminal.</p> <p>Here inside the brackets, within double quotes,</p> <p>Anything within the double quotes in the <b>printf</b> statement will be printed on the terminal.</p> <p>Type:</p> <p><b>Talk To a Teacher</b> backward slash (\) and "<b>n</b>"</p> <p><b>\n</b> (BackSlash n) signifies newline,</p> <p>As a result after the execution of the printf function the cursor moves to the new line.</p> <p>Every <b>C</b> statement must end with a semicolon(;</p> <p>Hence Type it at the end of this line.</p> <p>Semicolon(;) acts as a statement terminator.</p> |
| <pre>#include&lt;stdio.h&gt;  int main() { printf("Talk To a Teacher \n"); return 0; }</pre> | <p>Now press <b>Enter</b></p> <p>Give three space here.</p> <p>And type '<b>return</b> (space)<b>0</b>' and a semicolon ';'.</p> <p>This statement returns the integer zero.</p> <p>An integer has to be returned for this function.</p> <p>Because the function type is int.</p> <p>The return statement marks the end of the executable statements.</p> <p>We will learn more about the returned values in another tutorial.</p>  |
| <pre>#include&lt;stdio.h&gt;  int main() {</pre>   | <p>Now Click on the "<b>Save</b>" button to save the file.</p> <p>It is a good habit to save files frequently.</p> <p>This will protect you from sudden power failures.</p>   |

|  |   |
|--|---|
| <pre>printf("Talk To a Teacher \n"); return 0; }</pre> <p><b>Type:</b></p> <p><b>gcc talk.c -o myoutput</b></p> <p><b>Highlight</b></p> <p><b>gcc</b></p> <p><b>talk.c</b></p> <p><b>-o myoutput</b></p> | <p>It will also be useful in case the applications were to crash.</p> <p>Let us now compile the program</p> <p>Come back to our terminal</p> <p>Type: "<b>gcc</b>" space "<b>talk.c</b>" space hyphen o space "<b>myoutput</b>"</p> <p><b>gcc</b> is the compiler.</p> <p><b>talk.c</b> is the filename.</p> <p><b>-o myoutput</b> says that the executable should go to the file myoutput.</p> <p>Now press <b>Enter</b></p> <p>We see that the program is compiled.</p> |
| <p>Type:</p> <p><b>ls -lrt</b></p>   | <p>By typing <b>ls -lrt</b>, we can see that <b>myoutput</b> is the last file to be created.</p>  |
| <p><b>Type:</b></p> <p><b>./myoutput</b></p>   | <p>To execute the program,</p> <p>type <b>./myoutput</b> (dot slash "myoutput")</p> <p>press <b>Enter</b></p> <p>Here the output is displayed as: "Talk To a Teacher".</p>  |
|  | <p>As I said before, return is the last statement to be executed.</p> <p>Thus after the return statement nothing will be executed.</p> <p>Let us try it out.</p> <p>Come back to our program.</p>   |
|  | <p>After the return statement let us include one more printf statment.</p> <p>Give space here.</p> <p><b>printf</b> opening bracket closing bracket ()</p> <p>inside the brackets, within double quotes,</p> <p>type: "<b>Welcome</b>" backslash <b>n</b></p> <p>At the end type a semicolon ;</p>  |

|  |  |
|--|--|
|  | <p>Click on <b>Save</b></p> <p>Let us compile and execute.</p> <p>Come back to our terminal.</p> <p>You can recall the previously entered command by using up arrow key.</p> <p>That is what I did now.</p> <p>We see that the second statement <b>Welcome</b> is not executed.</p>  |
|  | <p>Now come back to our program.</p> <p>Let us write the Welcome statement above the return statement.</p>   |
|  | <p>Click on Save</p> <p>Let us compile and execute.</p>  |
|  | <p>We see that the second printf statement <b>Welcome</b> has also been executed.</p>  |
| <p>Errors</p> <p>Type:</p> <p>&lt;stdioh&gt;</p> | <p>Now,let us see the common errors which we can come across</p> <p>Come back to our program.</p> <p>Now suppose here I will miss the dot '.' in</p> <p><b>&lt;stdio.h&gt;</b></p> <p>Click on <b>Save</b></p> <p>Let us compile and execute.</p> <p>we see that there is an fatal error at line no.2 in our <b>talk.c</b> file.</p> <p>The compiler cannot find a header file with the name “<b>stdioh</b>”</p> <p>Hence it is giving an error "No such file or directory" and the compilation is terminated.</p> |
|  | <p>Let us now fix the error.</p> <p>Come back to our program.</p> <p>Reinsert the “.”</p>  |
|  | <p><b>Click on Save</b></p> <p>Let us compile and execute.</p> <p>Yes it is working.</p>   |

|   |  |
|---|--|
| <p>Error 2</p> <p>Type:</p> <pre>printf("Hello World \n")</pre> | <p>I will show you another common error.</p> <p>Let us switch back to the program.</p> <p>Now, suppose here I will miss the semicolon at the end of the line.</p> <p>Click on <b>Save</b></p> <p>Let us compile and execute.</p> <p>we see that there is an error at line no.6 in our <b>talk.c</b> file.</p> <p>That expected semicolon before <b>printf</b>.</p> <p>Come back to our program.</p> <p>As I said before, semicolon acts as a statement terminator.</p> <p>So it will search for it at the end of the line 5 and at the begning of the line 6.</p> <p>This is line 6.</p> <p>This is the last place where you can put the semicolon.</p> <p>Recall that compiler also gave the error message on line 6.</p> |
|   | <p>Let us try what happens if you put the semicolon here.</p> <p>Click on <b>Save</b>.</p>   |
|   | <p>Let us compile and execute.</p> <p>Yes it is working.</p>   |
|   | <p>Now come back to our program.</p> <p>Let us type the semicolon here at the end of this line.</p> <p>As it is the conventional practice to type the semicolon at the end of the line.</p>  |
|   | <p>Now click on <b>Save</b>.</p> <p>Let us compile and execute.</p> <p>Yes it is working.</p> <p>Now let us move back to our slides.</p>   |

|  |   |
|--|---|
| Slide 7<br>Assignment                        | <p>As an Assignment</p> <p>Write a program to print "<b>Welcome to the World of C</b>"</p> <p>See what happens if \n is not included in the <b>printf</b> statement.</p>  |
|  | This brings us to the end of this tutorial  |
| Slide 8<br>About the Spoken Tutorial Project | <p>Watch the video available at the link shown below</p> <p><a href="http://spoken-tutorial.org">http://spoken-tutorial.org</a> (<a href="http://spoken-tutorial.org/">http://spoken-tutorial.org/</a>) /<br/>What\_is\_a\_Spoken\_Tutorial</p> <p>It summarises the Spoken Tutorial project</p> <p>If you do not have good bandwidth, you can download and watch it.</p> |
| Slide Number 9<br>Spoken Tutorial Workshops  | <p>The Spoken Tutorial Project Team</p> <p>Conducts workshops using spoken tutorials</p> <p>Gives certificates to those who pass an online test</p> <p>For more details, please write to <a href="mailto:contact@spoken-tutorial.org">contact@spoken-tutorial.org</a></p>   |
| Slide Number 10<br>Acknowledgement           | <p>Spoken Tutorial Project is a part of the Talk to a Teacher project</p> <p>It is supported by the National Mission on Education through ICT, MHRD, Government of India</p> <p>More information on this Mission is available at the link shown below:<br/><a href="http://spoken-tutorial.org/NMEICT-Intro">http://spoken-tutorial.org/NMEICT-Intro</a></p>              |
| Slide Number 11<br>About the contributor     | <p>This is Ashwini Patil from IIT Bombay signing off</p> <p>Thank You for joining.</p>  |

## Contributors and Content Editors

Ashwini, PoojaMoolya

Retrieved from "<https://script.spoken-tutorial.org/index.php?title=C-and-C%2B%2B/C2/First-C-Program/English&oldid=14018>"

- This page was last modified on 14 July 2014, at 16:39.
- This page has been accessed 18,720 times.