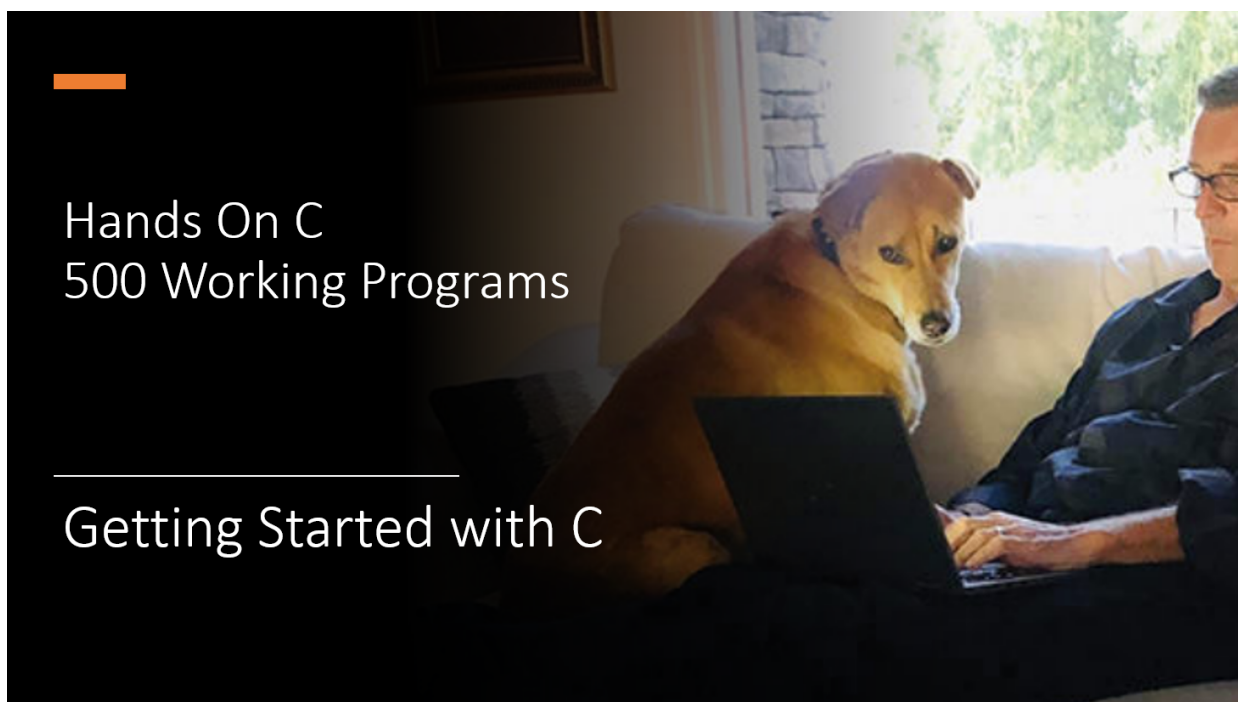




Hands On C
500 Working Programs

Getting Started with C

Why You Need to Learn C



Demand for C Programmers

Knowing C opens the door to JavaScript, Java, C++, and C#

C is ideal for embedded and real-time systems

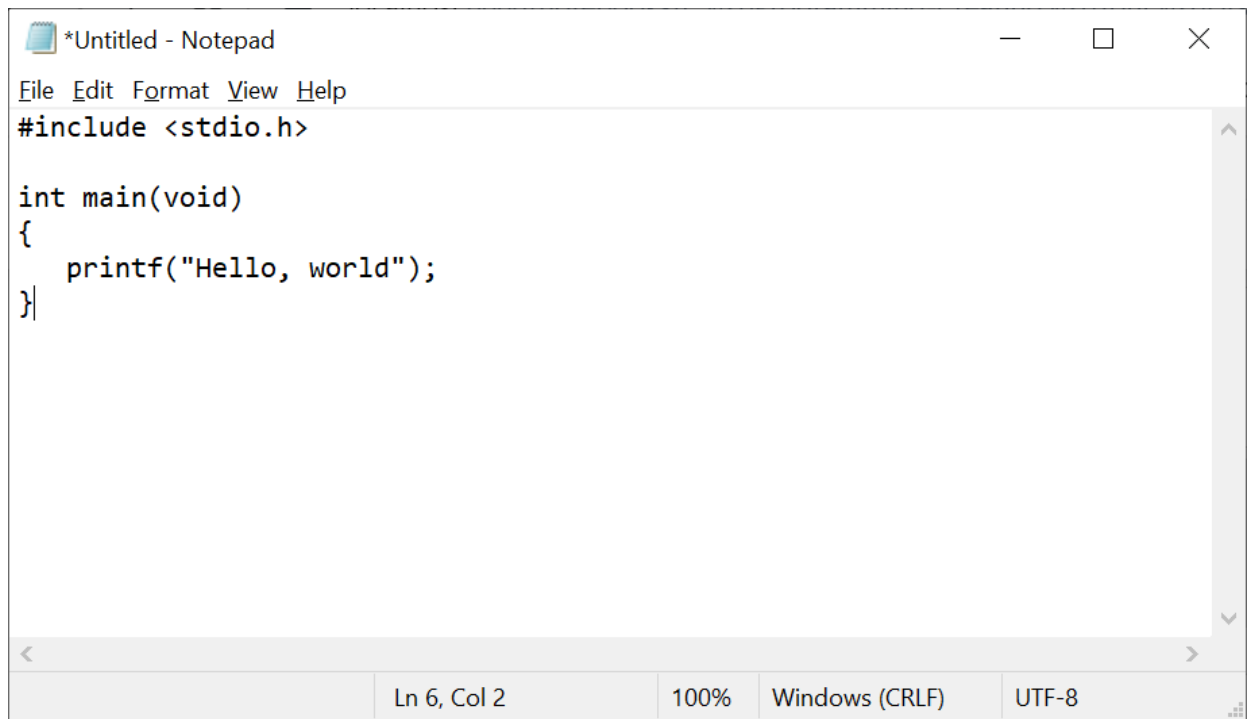
Programmers can learn C quickly

Creating an ASCII Source File

```
In [1]: #include <stdio.h>

int main(void)
{
    printf("Hello, world");
}
```

Hello, world



```
*Untitled - Notepad
File Edit Format View Help
#include <stdio.h>

int main(void)
{
    printf("Hello, world");
}
```

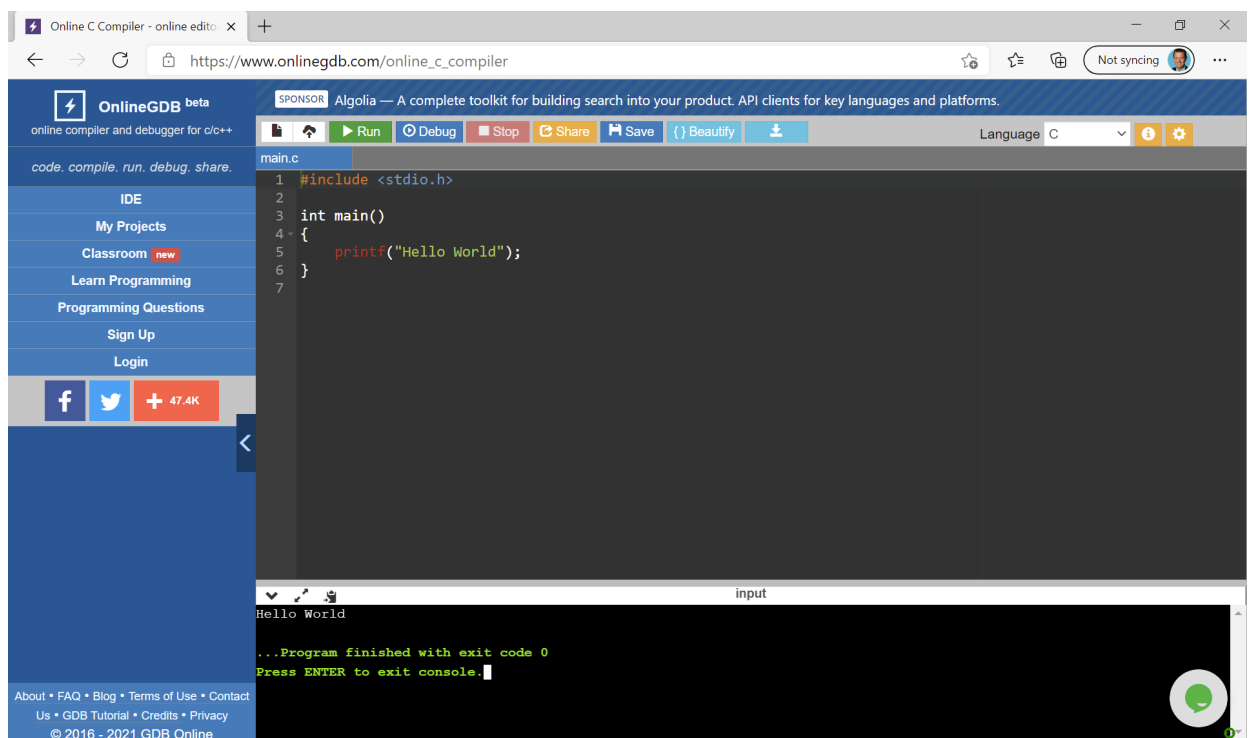
Ln 6, Col 2 100% Windows (CRLF) UTF-8

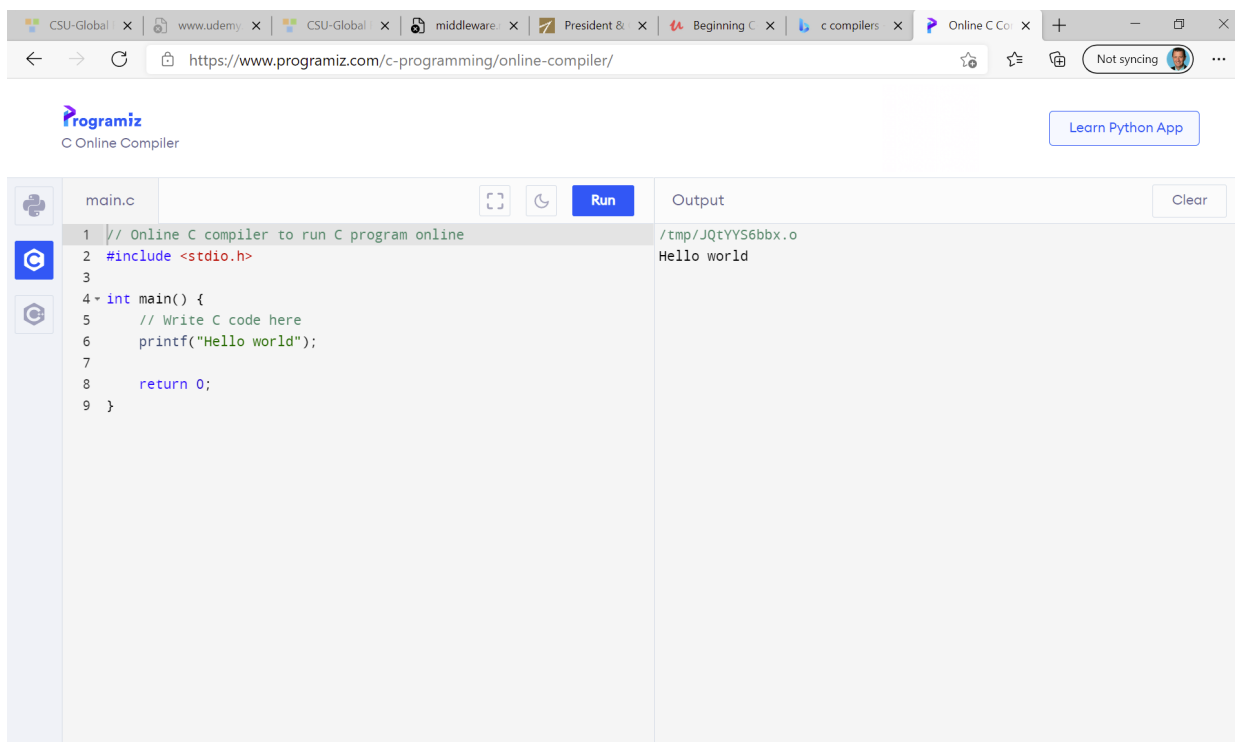
Understanding the Compilation and Program Development Process

1. Create a C source file
2. Compile and link your program's source file
3. Correct syntax errors
4. Compile your updated source file
5. Test your program logic

6. Correct any logic errors
7. Compile your updated source file

Choosing a C Compiler





The screenshot shows a web browser window with the URL <https://www.programiz.com/c-programming/online-compiler/>. The page features the Programiz logo and a "Learn Python App" button. The main area is divided into two panels: a code editor on the left and an output panel on the right. The code editor contains a C program named "main.c" with the following code:

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Hello world");
7
8     return 0;
9 }
```

The output panel shows the result of running the program: `/tmp/JQtYYS6bbx.o` and `Hello world`. A "Run" button is visible above the code editor, and a "Clear" button is visible above the output panel.

Typical Structure of a C Program

In [2]: `#include <stdio.h> // header files`

`// function definitions`

```
int main(void)
{
    // variable declarations
    // statements
}
```

Your First C Program -- Hello, World

In [3]: `#include <stdio.h>`

```
int main(void)
{
    printf("Hello, world");
}
```

Hello, world

Understanding Syntax Errors

In [4]: `#include <stdio.h>`

```
int main(void)
{
    printf("Hello, world!");
}
```

```
/tmp/tmpky0ce5ws.c: In function 'main':
/tmp/tmpky0ce5ws.c:5:6: warning: implicit declaration of function 'printf'; did
you mean 'printf'? [-Wimplicit-function-declaration]
    printf("Hello, world!");
    ^~~~~
    printf
/tmp/tmpky0ce5ws.c:5:12: warning: missing terminating " character
    printf("Hello, world!");
            ^
/tmp/tmpky0ce5ws.c:5:12: error: missing terminating " character
    printf("Hello, world!");
            ^~~~~~
/tmp/tmpky0ce5ws.c:6:1: error: expected expression before '}' token
}
^
/tmp/tmpky0ce5ws.c:6:1: error: expected ';' before '}' token
[C kernel] GCC exited with code 1, the executable will not be executed
```

In [5]: `#include <stdio.h>`

```
int main(void)
{
    printf("Hello, world")
}
```

/tmp/tmp2va5brwy.c: In function 'main':

/tmp/tmp2va5brwy.c:6:1: error: expected ';' before '}' token

```
}
^
```

[C kernel] GCC exited with code 1, the executable will not be executed

Understanding Logic Errors -- Bugs

In [6]: `#include <stdio.h>`

```
int main(void)
{
    printf("This is line one");
    printf("This is line two");
}
```

This is line oneThis is line two

```
In [7]: #include <stdio.h>

int main(void)
{
    printf("This is line one\n");
    printf("This is line two");
}
```

This is line one
This is line two

Adding Statements to Your Program

```
In [8]: #include <stdio.h>

int main(void)
{
    printf("Hello, world\n");
    printf("Hello, Kris\n");
    printf("C is easy!\n");
}
```

Hello, world
Hello, Kris
C is easy!

To C, Upper and Lowercase Letters Matter

In [9]: `#include <stdio.h>`

```
int main(void)
{
    Printf("Hello, world");
}
```

```
/tmp/tmpsvm_jrh1.c: In function 'main':
/tmp/tmpsvm_jrh1.c:5:5: warning: implicit declaration of function 'Printf'; did
you mean 'printf'? [-Wimplicit-function-declaration]
    Printf("Hello, world");
    ^~~~~~
    printf
/tmp/tmp6hje2p80.out: symbol lookup error: /tmp/tmpmvuh0pgd.out: undefined symb
ol: Printf
[C kernel] Executable exited with code 127
```

Understanding C Program File Types

Header
C Source Files
Linker object Files
Library files
Executable files

Understanding How C Uses Header Files

```
In [10]: #include <stdio.h>

int main(void)
{
    printf("Hello, world");
}
```

Hello, world

```
In [11]: int main(void)
{
    printf("Hello, world");
}
```

```
/tmp/tmpnjzpdri3.c: In function 'main':
/tmp/tmpnjzpdri3.c:3:3: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
    printf("Hello, world");
    ^~~~~~
/tmp/tmpnjzpdri3.c:3:3: warning: incompatible implicit declaration of built-in function 'printf'
/tmp/tmpnjzpdri3.c:3:3: note: include '<stdio.h>' or provide a declaration of 'printf'
```

Hello, world

Comments Describe Your Programs

```
// Hello.c
```

```
// Written by: Kris Jamsa
// Date Written: 1-31-21
// Purpose: Display the Hello, World message

#include <stdio.h> // needed for printf

int main(void)
{
    printf("Hello, world");
}
```

In [12]:

```
/* Hello.c
   Written by: Kris Jamsa
   Date Written: 1-31-21
   Purpose: Display the Hello, World message */

#include <stdio.h> // needed for printf

int main(void)
{
    printf("Hello, world");
}
```

Hello, world

Using Comments to Turn Off Program Statements

```
In [13]: #include <stdio.h>

int main(void)
{
    printf("Message 1\n");
    printf("Message 2\n");
    printf("Message 3\n");
}
```

Message 1

Message 2

Message 3

The Semicolon in C

In [14]: `#include <stdio.h>`

```
int main(void)
{
    printf("This is line 1\n")
    printf("This is line 2\n");
}
```

/tmp/tmpbph97qtb.c: In function 'main':

/tmp/tmpbph97qtb.c:6:5: error: expected ';' before 'printf'

```
    printf("This is line 2\n");
    ^~~~~~
```

[C kernel] GCC exited with code 1, the executable will not be executed

Pay Attention to Compiler Warnings

In [15]: `#include <stdio.h>`

```
main(void)
{
    printf("Hello, world");
}
```

/tmp/tmpdksbfzw8.c:3:1: warning: return type defaults to 'int' [-Wimplicit-int]

```
main(void)
^~~~~
```

Hello, world

Ways You May Encounter main

```
In [16]: #include <stdio.h>

int main(void)
{
    printf("Hello, world");
}
```

Hello, world

```
In [17]: #include <stdio.h>

void main(void)
{
    printf("Hello, world");
}
```

Hello, world

[C kernel] Executable exited with code 12

In [18]: `#include <stdio.h>`

```
main()
{
    printf("Hello, world");
}
```

/tmp/tmp098yiwv.c:3:1: warning: return type defaults to 'int' [-Wimplicit-int]

```
main()
^~~~
```

Hello, world

In [19]: `#include <stdio.h>`

```
int main(int argc, char *argv[])
{
    printf("Hello, world");
}
```

Hello, world

What You will Learn Next

Programs store data in variables as they execute

Each variable is a specific type, such as int, which defines a set of values the variable can store and a set of operations the program can perform on the variable

