

C Programming

Practice 0

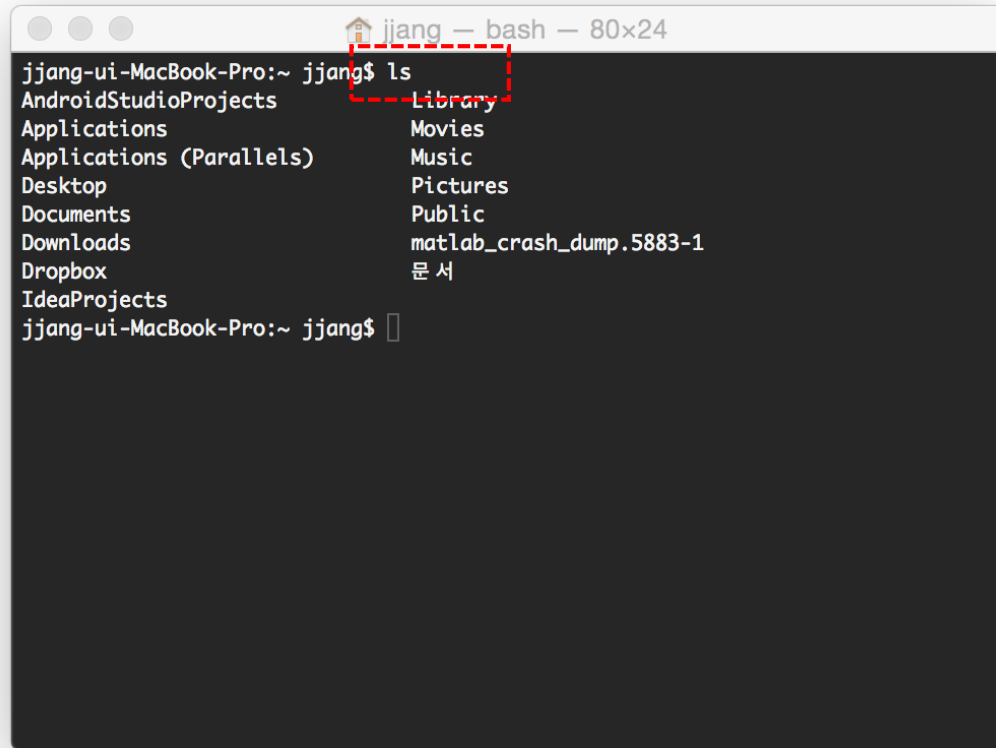
Introduction

- Teaching assistant : Won-cheol Jang (장원철)
- Assignment submission e-mail:
hizorro99@naver.com
- E-mail title: day(Thursday or Friday)_name_#week
 - Ex) Thursday_james_2week
- File title: student id_name_#week.c
 - Ex) 2014123456_james_2week.c
- Weekly assignment: each 1 point / semester's maximum score: 15
- 3 Project: each 5 point / semester's maximum score: 15

Linux brand



Linux terminal - Navigation



A screenshot of a Linux terminal window titled "jjang — bash — 80x24". The terminal shows the command "ls" being executed, which lists the contents of the current directory. The output is as follows:

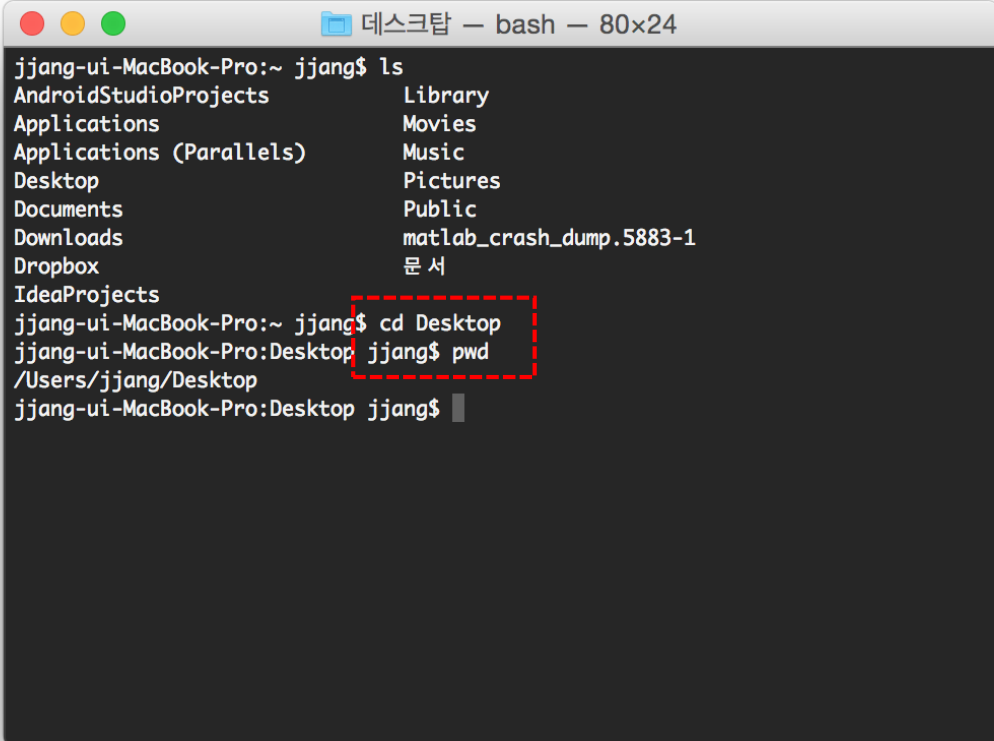
```
jjang-ui-MacBook-Pro:~ jjang$ ls
AndroidStudioProjects  Library
Applications           Movies
Applications (Parallels) Music
Desktop               Pictures
Documents             Public
Downloads             matlab_crash_dump.5883-1
Dropbox               문서
IdeaProjects
```

The terminal prompt "jjang-ui-MacBook-Pro:~ jjang\$" is visible at the bottom of the window.

\$ ls - list files and directories

Reference: <http://linuxcommand.org/index.php>

Linux terminal - Navigation



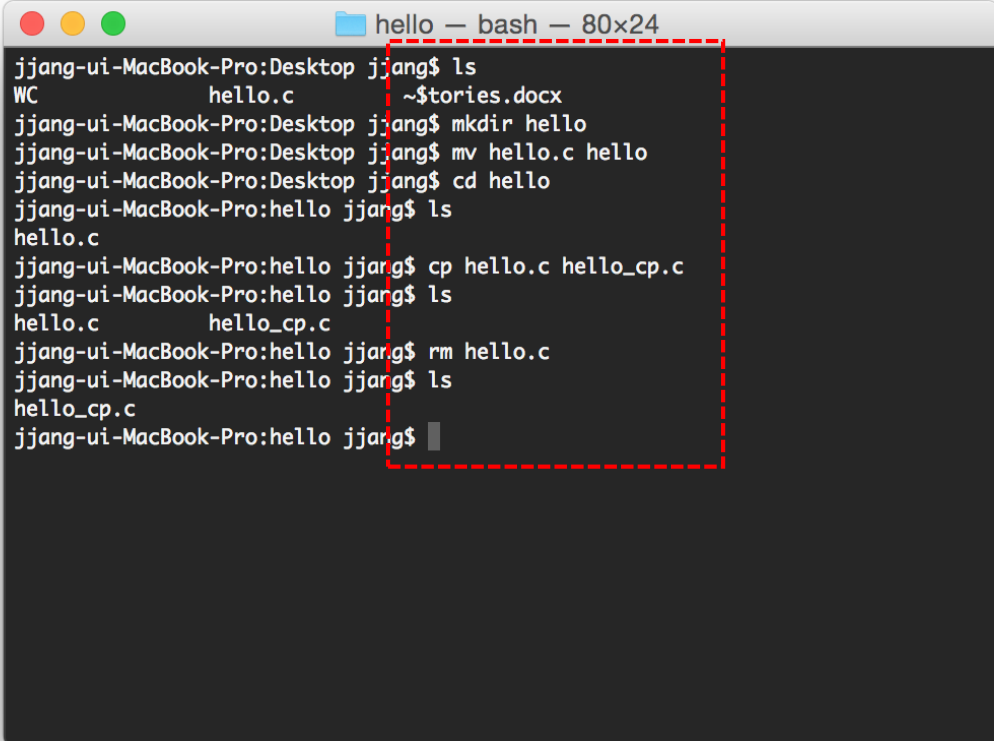
A screenshot of a Linux terminal window. The title bar shows a folder icon, the text '데스크탑' (Desktop), and 'bash — 80x24'. The terminal content shows a user named 'jjang' at a machine named 'jjang-ui-MacBook-Pro'. The user runs 'ls' to list the contents of the home directory, which includes folders like 'AndroidStudioProjects', 'Applications', 'Desktop', 'Downloads', 'IdeaProjects', and files like 'Library', 'Movies', 'Music', 'Pictures', 'Public', 'matlab_crash_dump.5883-1', and '문서'. Then, the user runs 'cd Desktop', and the prompt changes to 'jjang-ui-MacBook-Pro:Desktop'. Finally, the user runs 'pwd', and the output is '/Users/jjang/Desktop'. A red dashed box highlights the 'cd Desktop' and 'pwd' commands and their immediate prompts.

```
jjang-ui-MacBook-Pro:~ jjang$ ls
AndroidStudioProjects  Library
Applications           Movies
Applications (Parallels) Music
Desktop               Pictures
Documents              Public
Downloads              matlab_crash_dump.5883-1
Dropbox                문서
IdeaProjects
jjang-ui-MacBook-Pro:~ jjang$ cd Desktop
jjang-ui-MacBook-Pro:Desktop jjang$ pwd
/Users/jjang/Desktop
jjang-ui-MacBook-Pro:Desktop jjang$
```

\$ cd - change directory

\$ pwd - print working directory

Linux terminal - Manipulating Files

A screenshot of a Linux terminal window titled "hello — bash — 80x24". The terminal shows a series of commands and their outputs. A red dashed rectangle highlights the first five lines of the terminal output. The commands and outputs are as follows:

```
jjang-ui-MacBook-Pro:Desktop jjang$ ls
WC      hello.c      ~$stories.docx
jjang-ui-MacBook-Pro:Desktop jjang$ mkdir hello
jjang-ui-MacBook-Pro:Desktop jjang$ mv hello.c hello
jjang-ui-MacBook-Pro:Desktop jjang$ cd hello
jjang-ui-MacBook-Pro:hello jjang$ ls
hello.c
jjang-ui-MacBook-Pro:hello jjang$ cp hello.c hello_cp.c
jjang-ui-MacBook-Pro:hello jjang$ ls
hello.c      hello_cp.c
jjang-ui-MacBook-Pro:hello jjang$ rm hello.c
jjang-ui-MacBook-Pro:hello jjang$ ls
hello_cp.c
jjang-ui-MacBook-Pro:hello jjang$
```

\$ mkdir - create directories

\$ mv - move or rename files

\$ cp - copy files and directories

\$ rm - remove files and directories

GCC (GNU Compiler Collection) installation

```
howtogeek@ubuntu: ~  
howtogeek@ubuntu:~$ sudo apt-get install build-essential  
[sudo] password for howtogeek:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  dpkg-dev fakeroot g++ g++-4.6 libalgorithm-diff-perl  
  libalgorithm-diff-xs-perl libalgorithm-merge-perl  
  libdpkg-perl libstdc++6-4.6-dev libtimedate-perl patch  
Suggested packages:  
  debian-keyring g++-multilib g++-4.6-multilib gcc-4.6-doc  
  libstdc++6-4.6-dbg libstdc++6-4.6-doc diffutils-doc  
The following NEW packages will be installed:  
  build-essential dpkg-dev fakeroot g++ g++-4.6
```

\$ sudo apt-get install build-essential

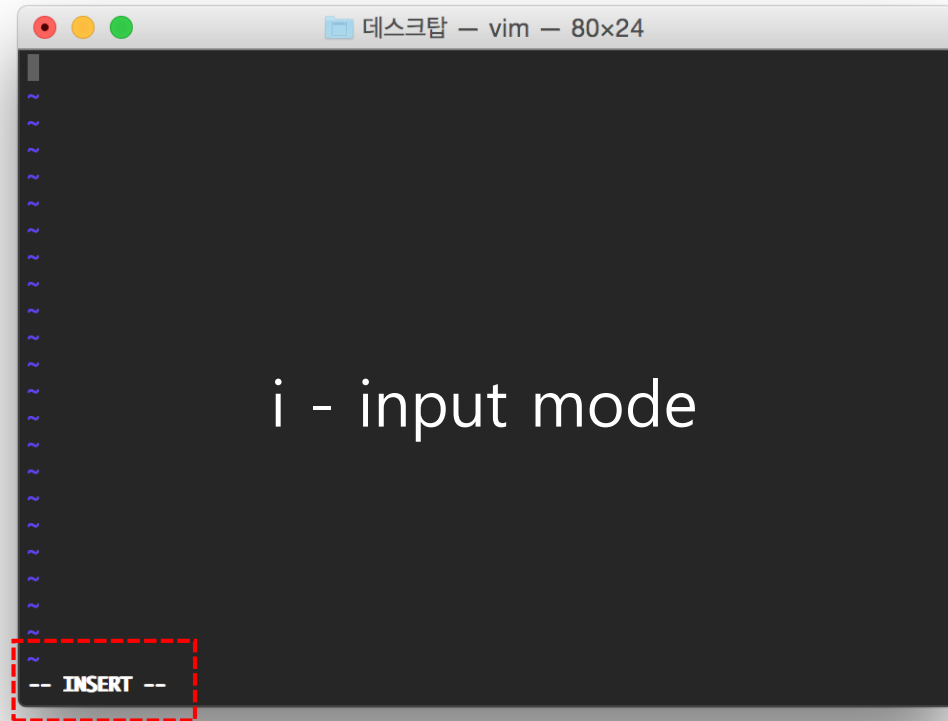
How to use vi editor

```
jjang-ui-MacBook-Pro:~ jjang$ ls
AndroidStudioProjects  Library
Applications           Movies
Applications (Parallels) Music
Desktop                Pictures
Documents              Public
Downloads              matlab_crash_dump.5883-1
Dropbox                문서
IdeaProjects

jjang-ui-MacBook-Pro:~ jjang$ cd Desktop
jjang-ui-MacBook-Pro:Desktop jjang$ pwd
/Users/jjang/Desktop
jjang-ui-MacBook-Pro:Desktop jjang$ vi hello.c
jjang-ui-MacBook-Pro:Desktop jjang$
```

```
"hello.c" [New File]
```


How to use vi editor



A screenshot of the vi editor window titled "데스크탑 — vim — 80x24". The editor is in input mode, displaying "i - input mode" in the center. At the bottom left, a red dashed box highlights the status "-- INSERT --". The left margin shows a vertical column of tilde (~) characters.

```
i - input mode
```

```
-- INSERT --
```



A screenshot of the vi editor window titled "데스크탑 — vim — 80x24". The editor displays a C program. The status at the bottom right indicates the cursor is at line 6, column 76 ("hello.c" 6L, 76C). The left margin shows a vertical column of tilde (~) characters.

```
#include <stdio.h>

int main(void){
    printf("hello world!\n");
    return 0;
}

"hello.c" 6L, 76C
```

How to use vi editor

A screenshot of a terminal window running the Vim text editor. The title bar at the top shows three colored circles (red, yellow, green) on the left and the text "데스크탑 — vim — 80x24" on the right. The main area has a dark background with light-colored text. It contains a C program snippet:

```
#include <stdio.h>

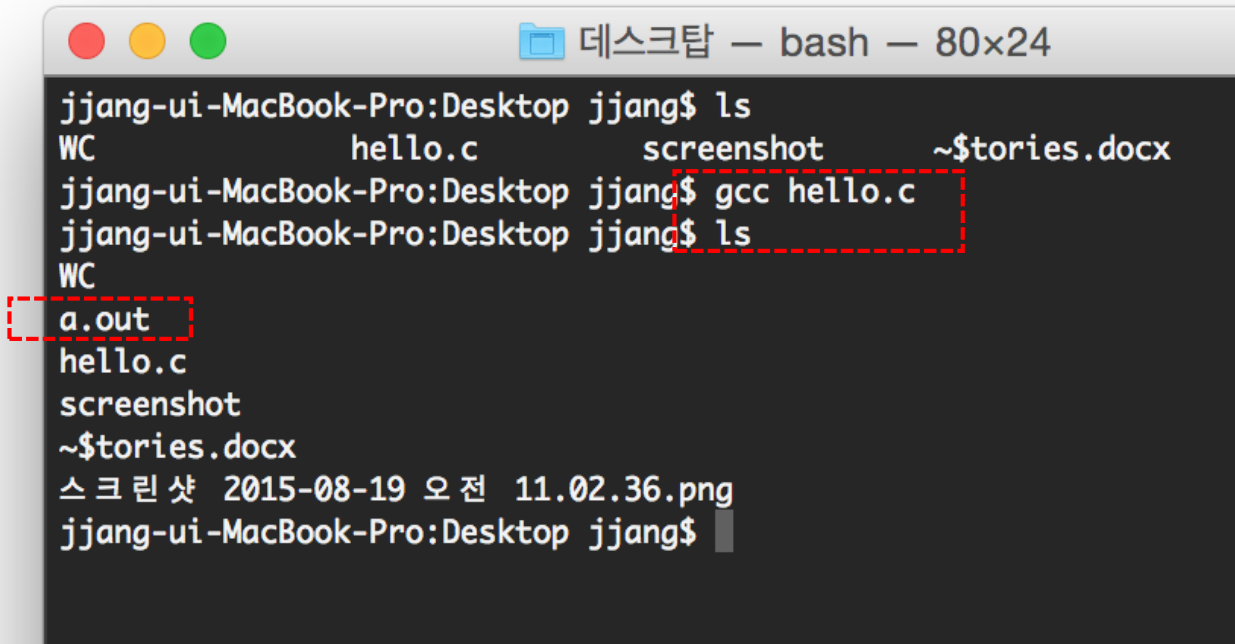
int main(void){
    printf("hello world!\n");
    return 0;
}
```

The first line of the code is highlighted in blue. Below the closing brace of the function, there are several lines, each starting with a tilde (~). At the bottom left, a red dashed rectangle highlights the command prompt ":wq".

Esc – command mode

:wq – save and quit

Compile the program

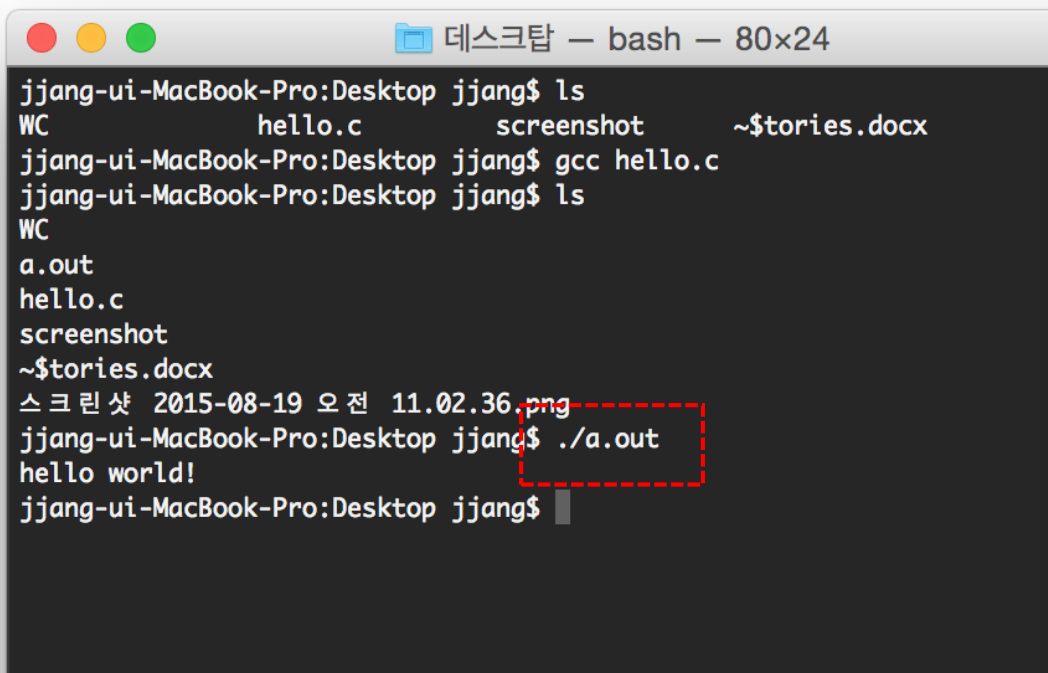


```
jjang-ui-MacBook-Pro:Desktop jjang$ ls
WC          hello.c          screenshot    ~$stories.docx
jjang-ui-MacBook-Pro:Desktop jjang$ gcc hello.c
jjang-ui-MacBook-Pro:Desktop jjang$ ls
WC
a.out
hello.c
screenshot
~$stories.docx
스크린샷 2015-08-19 오전 11.02.36.png
jjang-ui-MacBook-Pro:Desktop jjang$
```

The image shows a terminal window titled '데스크탑 — bash — 80x24'. The user runs 'ls' and lists files: 'WC', 'hello.c', 'screenshot', and '~\$stories.docx'. Then, they run 'gcc hello.c', which is highlighted with a red dashed box. After another 'ls' command, the output now includes 'a.out', which is also highlighted with a red dashed box. The terminal shows the successful compilation of 'hello.c' into 'a.out'.

\$ gcc filename(.c) - compile c code

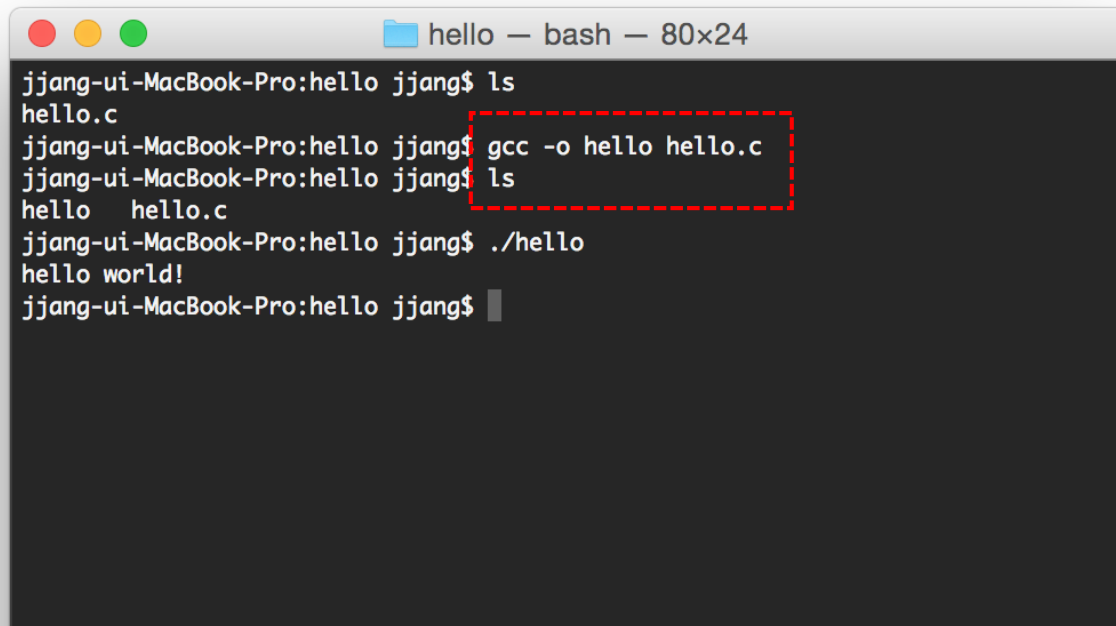
Execute the program



```
jjang-ui-MacBook-Pro:Desktop jjang$ ls
WC          hello.c          screenshot    ~$stories.docx
jjang-ui-MacBook-Pro:Desktop jjang$ gcc hello.c
jjang-ui-MacBook-Pro:Desktop jjang$ ls
WC
a.out
hello.c
screenshot
~$stories.docx
스크린샷 2015-08-19 오전 11.02.36.png
jjang-ui-MacBook-Pro:Desktop jjang$ ./a.out
hello world!
jjang-ui-MacBook-Pro:Desktop jjang$
```

\$./filename - execute program

How to use gcc options (-o)

A terminal window titled 'hello — bash — 80x24' with a dark background and light text. The window shows a series of commands and their outputs. The command 'gcc -o hello hello.c' is highlighted with a red dashed rectangle. The output of the program is 'hello world!'.

```
jjang-ui-MacBook-Pro:hello jjang$ ls
hello.c
jjang-ui-MacBook-Pro:hello jjang$ gcc -o hello hello.c
jjang-ui-MacBook-Pro:hello jjang$ ls
hello  hello.c
jjang-ui-MacBook-Pro:hello jjang$ ./hello
hello world!
jjang-ui-MacBook-Pro:hello jjang$
```

\$ gcc -o filename filename(.c)
- execute program

How to use gcc options (-o)

```
#include <stdio.h>

void func1();
void func2();
~
~
~
```

test.h

```
#include "test.h"

int main(void){
    func1();
    func2();
    return 0;
}
~
~
```

main.c

```
#include <stdio.h>

void func1(){
    print("function1\n");
}
~
~
~
~
~
```

func1.c

```
#include <stdio.h>

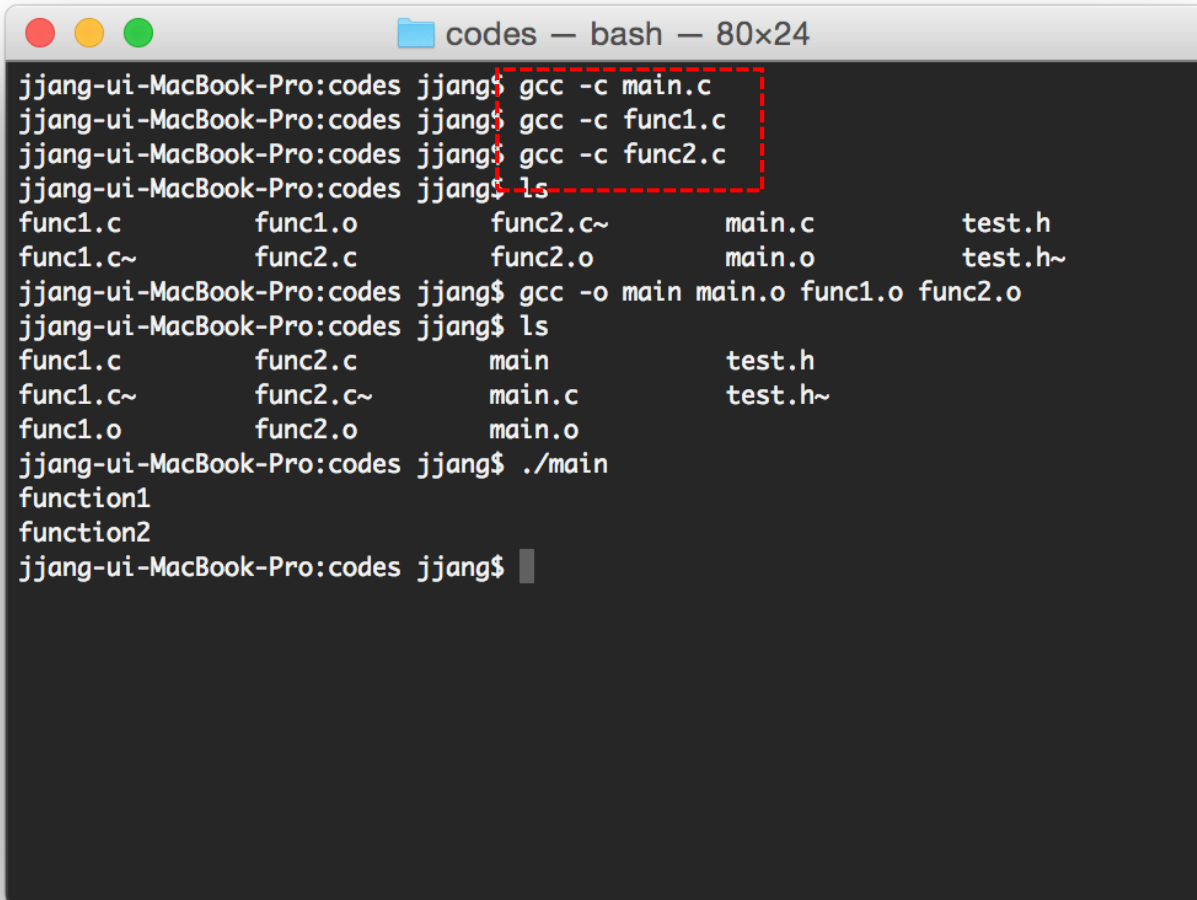
void func2(){
    print("function2\n");
}
~
~
~
~
~
```

func2.c

```
jjang-ui-MacBook-Pro:codes jjang$ ls
func1.c      func2.c      main.c      test.h~
func1.c~     func2.c~     test.h
jjang-ui-MacBook-Pro:codes jjang$ gcc -o main main.c func1.c func2.c
jjang-ui-MacBook-Pro:codes jjang$ ls
func1.c      func2.c      main      test.h
func1.c~     func2.c~     main.c    test.h~
jjang-ui-MacBook-Pro:codes jjang$ ./main
function1
function2
jjang-ui-MacBook-Pro:codes jjang$
```

\$ gcc -o filename filename(.c) filename(.c) filename(.c)
- execute program

How to use gcc options (-c)

A terminal window titled 'codes — bash — 80x24' showing a series of gcc commands and their outputs. The commands are: 'gcc -c main.c', 'gcc -c func1.c', 'gcc -c func2.c', 'ls', 'gcc -o main main.o func1.o func2.o', 'ls', and './main'. The first three gcc commands are enclosed in a red dashed box. The outputs show the creation of object files (main.o, func1.o, func2.o) and the execution of the main program, which prints 'function1' and 'function2'.

```
jjang-ui-MacBook-Pro:codes jjang$ gcc -c main.c
jjang-ui-MacBook-Pro:codes jjang$ gcc -c func1.c
jjang-ui-MacBook-Pro:codes jjang$ gcc -c func2.c
jjang-ui-MacBook-Pro:codes jjang$ ls
func1.c      func1.o      func2.c~    main.c      test.h
func1.c~    func2.c      func2.o     main.o      test.h~
jjang-ui-MacBook-Pro:codes jjang$ gcc -o main main.o func1.o func2.o
jjang-ui-MacBook-Pro:codes jjang$ ls
func1.c      func2.c      main        test.h
func1.c~    func2.c~    main.c      test.h~
func1.o      func2.o      main.o
jjang-ui-MacBook-Pro:codes jjang$ ./main
function1
function2
jjang-ui-MacBook-Pro:codes jjang$
```

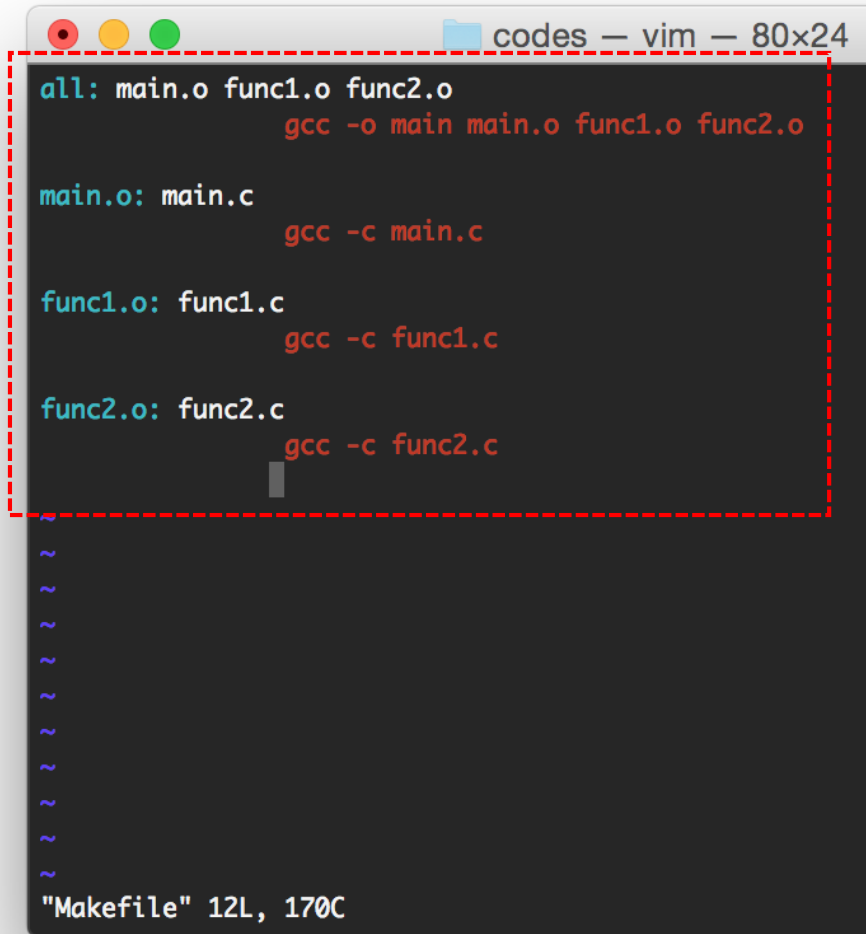
\$ gcc -c filename(.c) – create object file

How to use gcc options

```
codes — bash — 80x24
jjang-ui-MacBook-Pro:codes jjang$ gcc -c main.c
jjang-ui-MacBook-Pro:codes jjang$ gcc -c func1.c
jjang-ui-MacBook-Pro:codes jjang$ gcc -c func2.c
jjang-ui-MacBook-Pro:codes jjang$ ls
func1.c      func1.o      func2.c~     main.c       test.h
func1.c~     func2.c      func2.o      main.o       test.h~
jjang-ui-MacBook-Pro:codes jjang$ gcc -o main main.o func1.o func2.o
jjang-ui-MacBook-Pro:codes jjang$ ls
func1.c      func2.c      main         test.h
func1.c~     func2.c~     main.c       test.h~
func1.o      func2.o      main.o
jjang-ui-MacBook-Pro:codes jjang$ ./main
function1
function2
jjang-ui-MacBook-Pro:codes jjang$ vi func2.c
jjang-ui-MacBook-Pro:codes jjang$ gcc -c func2.c
jjang-ui-MacBook-Pro:codes jjang$ gcc -o main main.o func1.o func2.o
jjang-ui-MacBook-Pro:codes jjang$ ./main
function1
function3
jjang-ui-MacBook-Pro:codes jjang$
```

\$ gcc -c filename(.c) – create object file

Create a Makefile



```
codes — vim — 80x24
all: main.o func1.o func2.o
    gcc -o main main.o func1.o func2.o

main.o: main.c
    gcc -c main.c

func1.o: func1.c
    gcc -c func1.c

func2.o: func2.c
    gcc -c func2.c

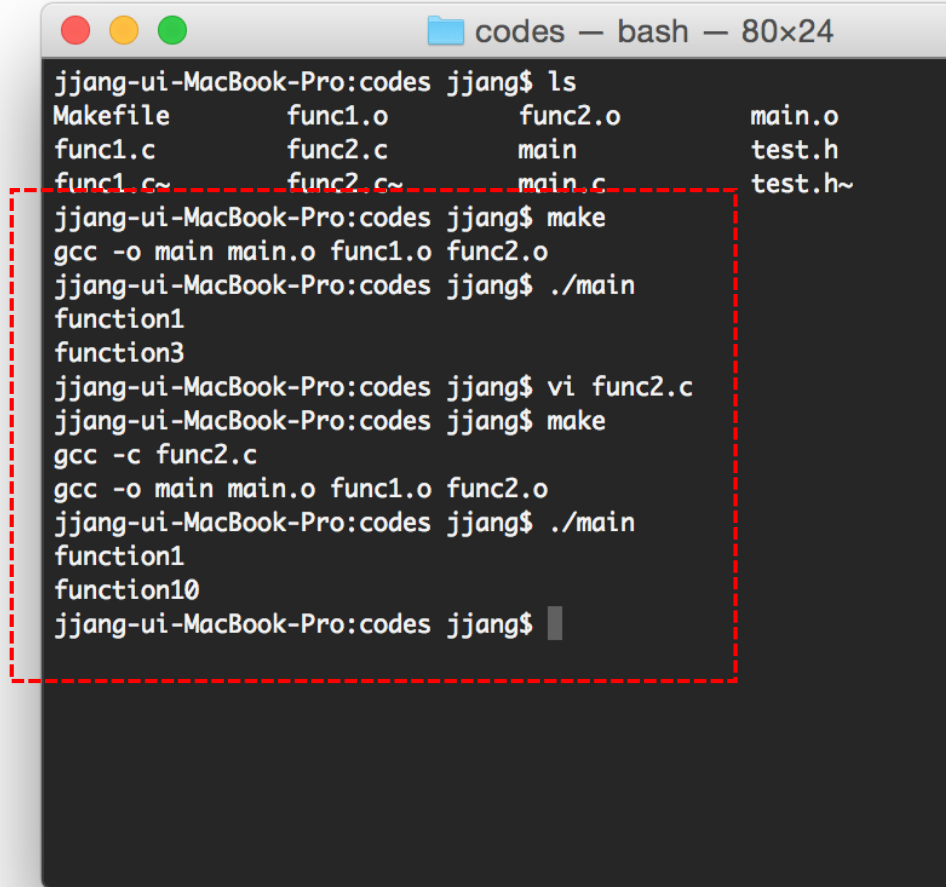
~
~
~
~
~
~
~
~
~
~
"Makefile" 12L, 170C
```

\$ vi Makefile

Reference:

<http://powergi.tistory.com/entry/%ED%8E%8C-Makefile%EC%9D%84-%EB%A7%8C%EB%93%A4%EC%96%B4%EB%B3%B4%EC%9E%90-1>

How to use the Makefile

A terminal window titled 'codes — bash — 80x24' showing a series of commands and their outputs. The window has a red dashed box highlighting the 'make' and './main' commands and their outputs. The terminal shows the following sequence of commands and outputs:

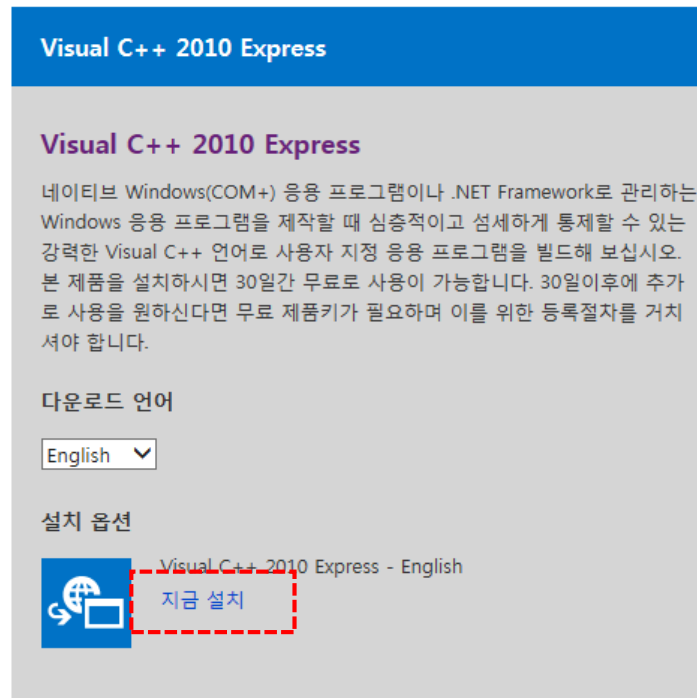
```
jjang-ui-MacBook-Pro:codes jjang$ ls
Makefile      func1.o      func2.o      main.o
func1.c       func2.c      main         test.h
func1.c~      func2.c~     main.c       test.h~
jjang-ui-MacBook-Pro:codes jjang$ make
gcc -o main main.o func1.o func2.o
jjang-ui-MacBook-Pro:codes jjang$ ./main
function1
function3
jjang-ui-MacBook-Pro:codes jjang$ vi func2.c
jjang-ui-MacBook-Pro:codes jjang$ make
gcc -c func2.c
gcc -o main main.o func1.o func2.o
jjang-ui-MacBook-Pro:codes jjang$ ./main
function1
function10
jjang-ui-MacBook-Pro:codes jjang$
```

\$ make or make all – run makefile

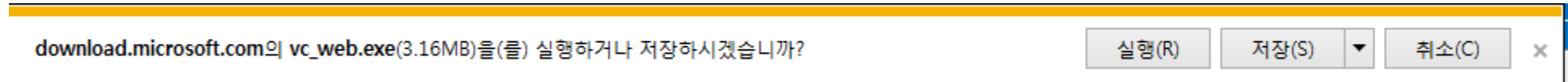
How to use Visual Studio

- <http://www.visualstudio.com/downloads/download-visual-studio-vs>

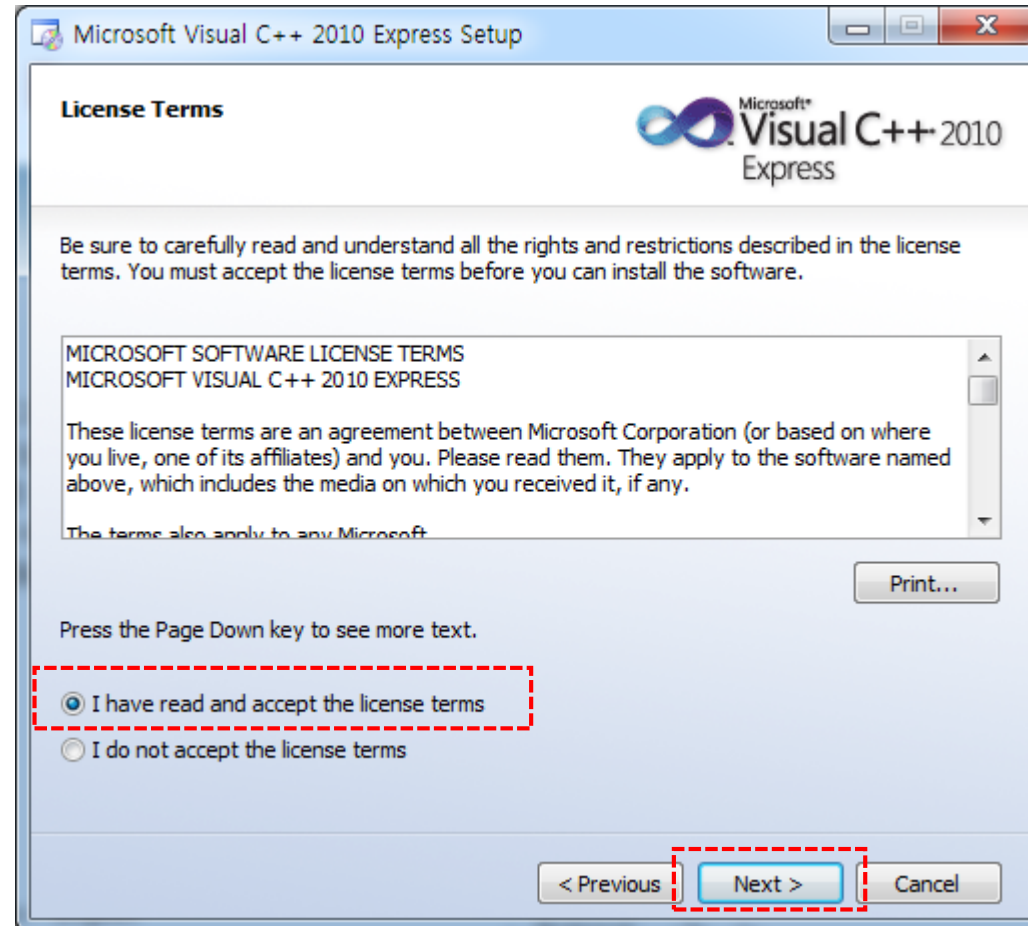
Visual Studio 2010 Express



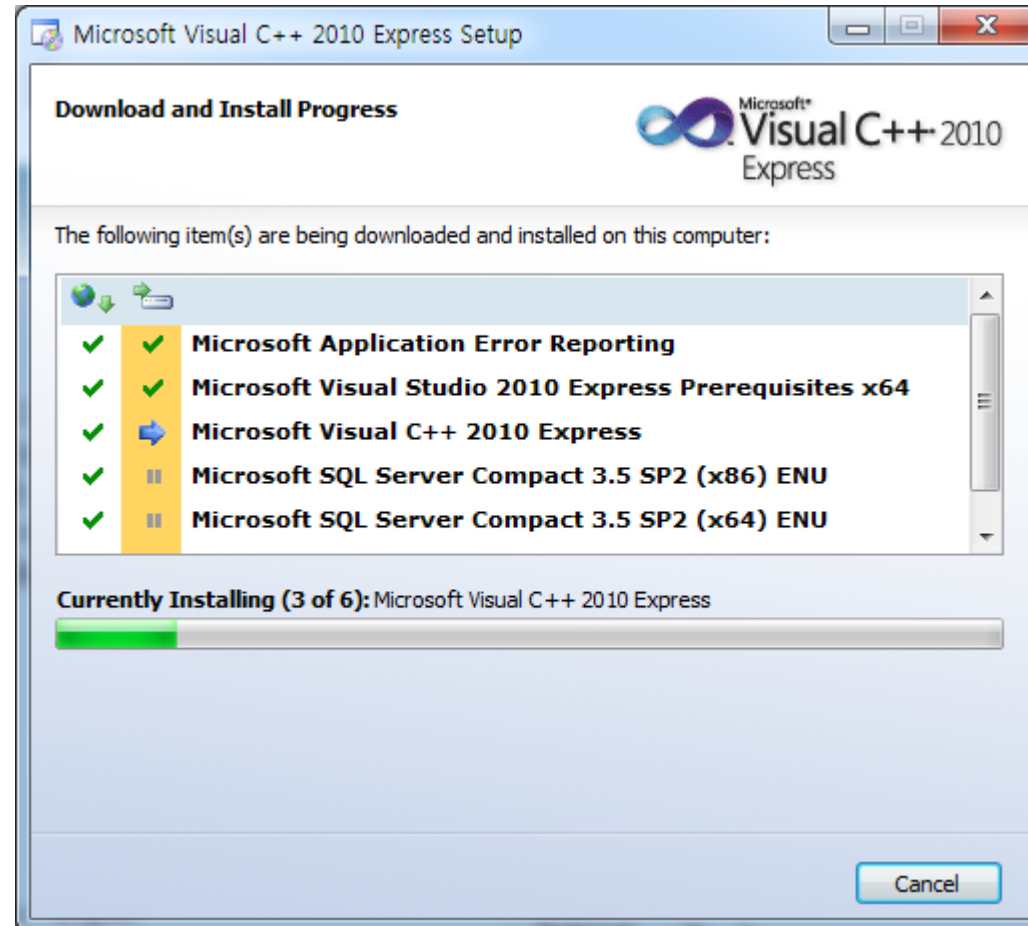
How to use Visual Studio



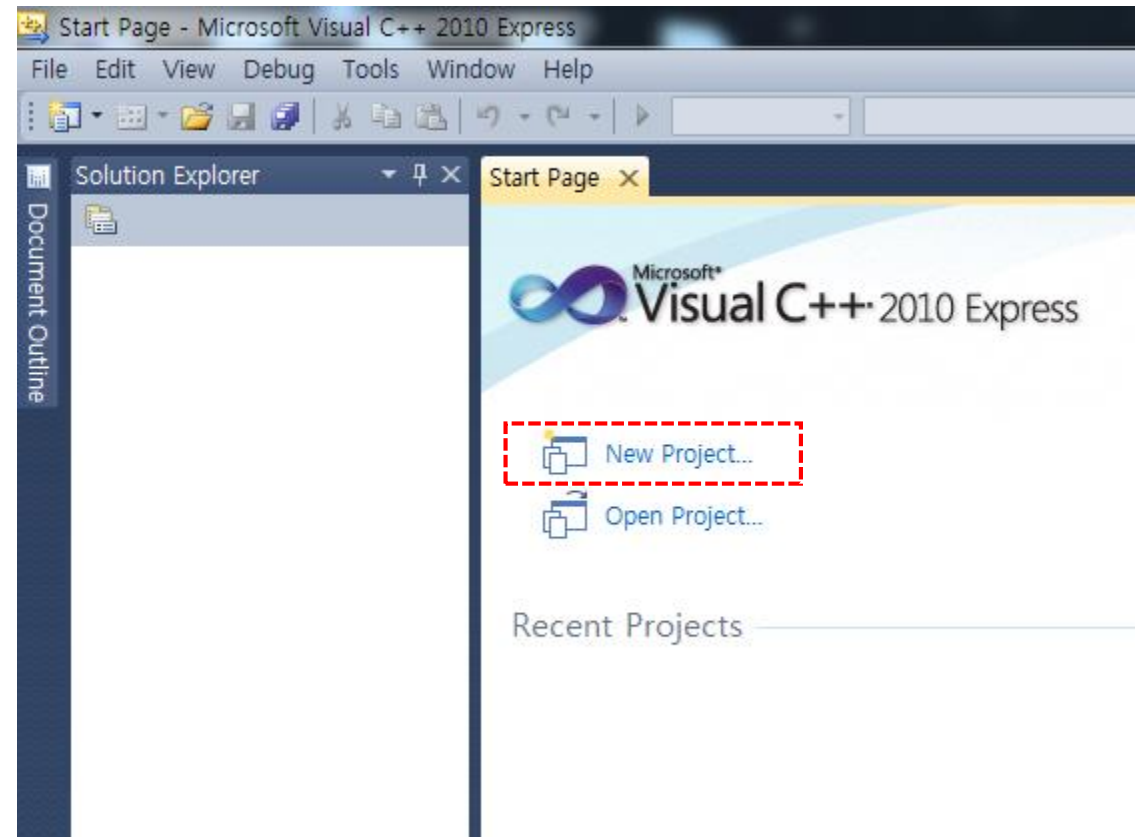
How to use Visual Studio



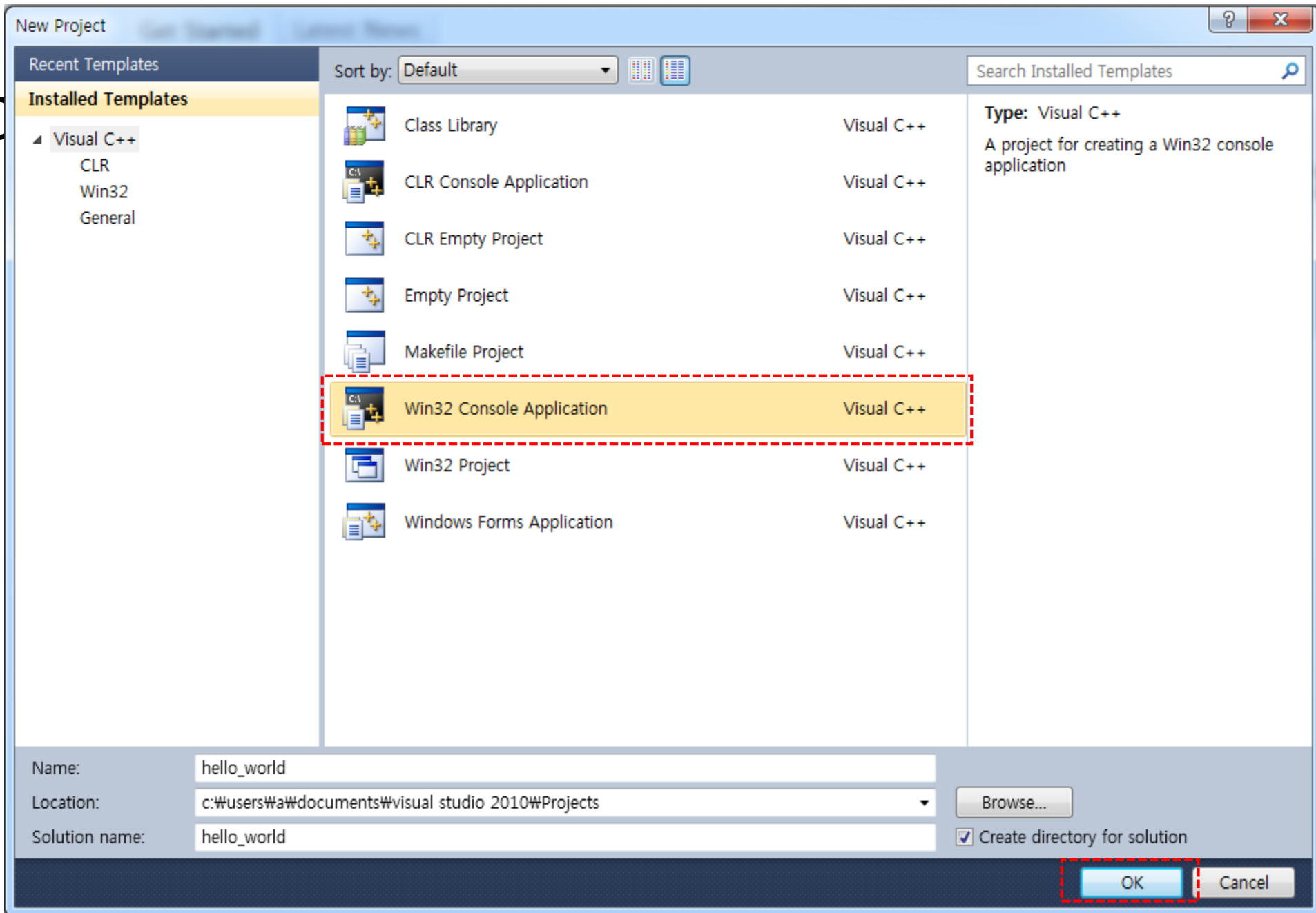
How to use Visual Studio



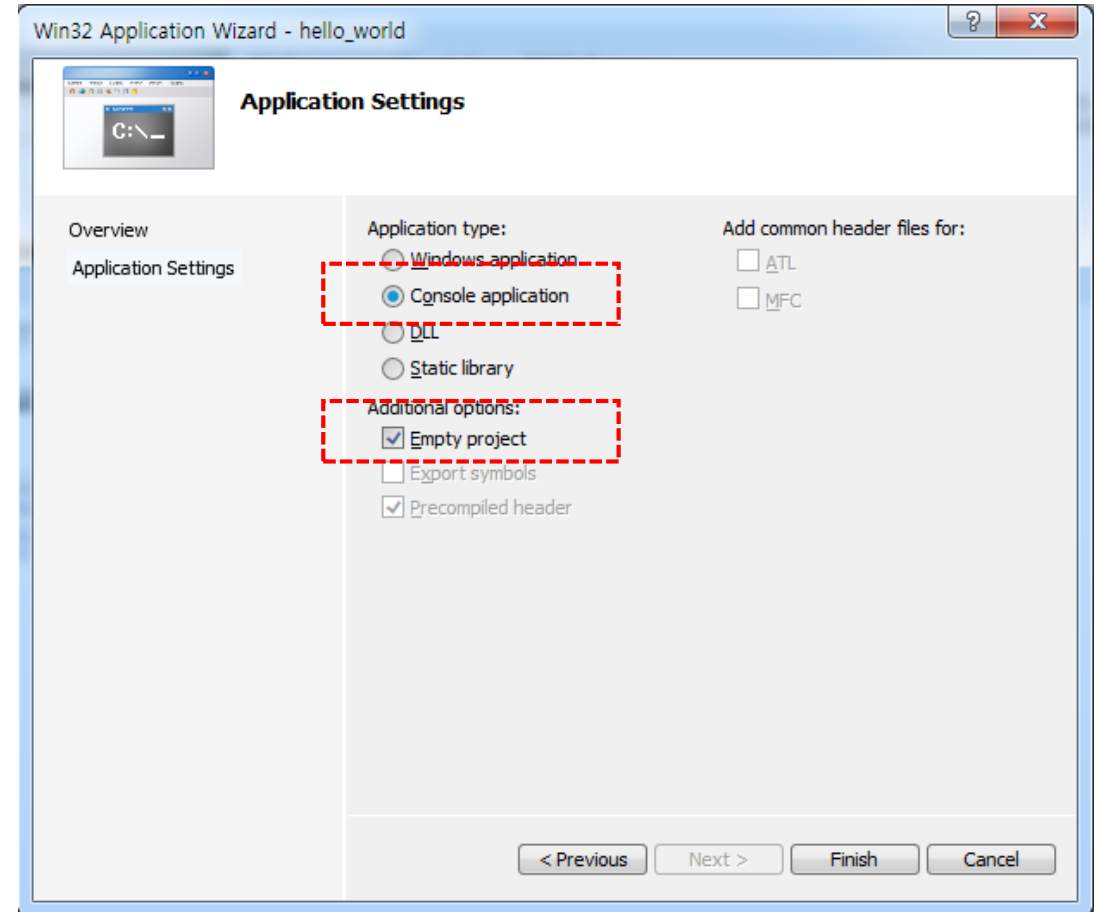
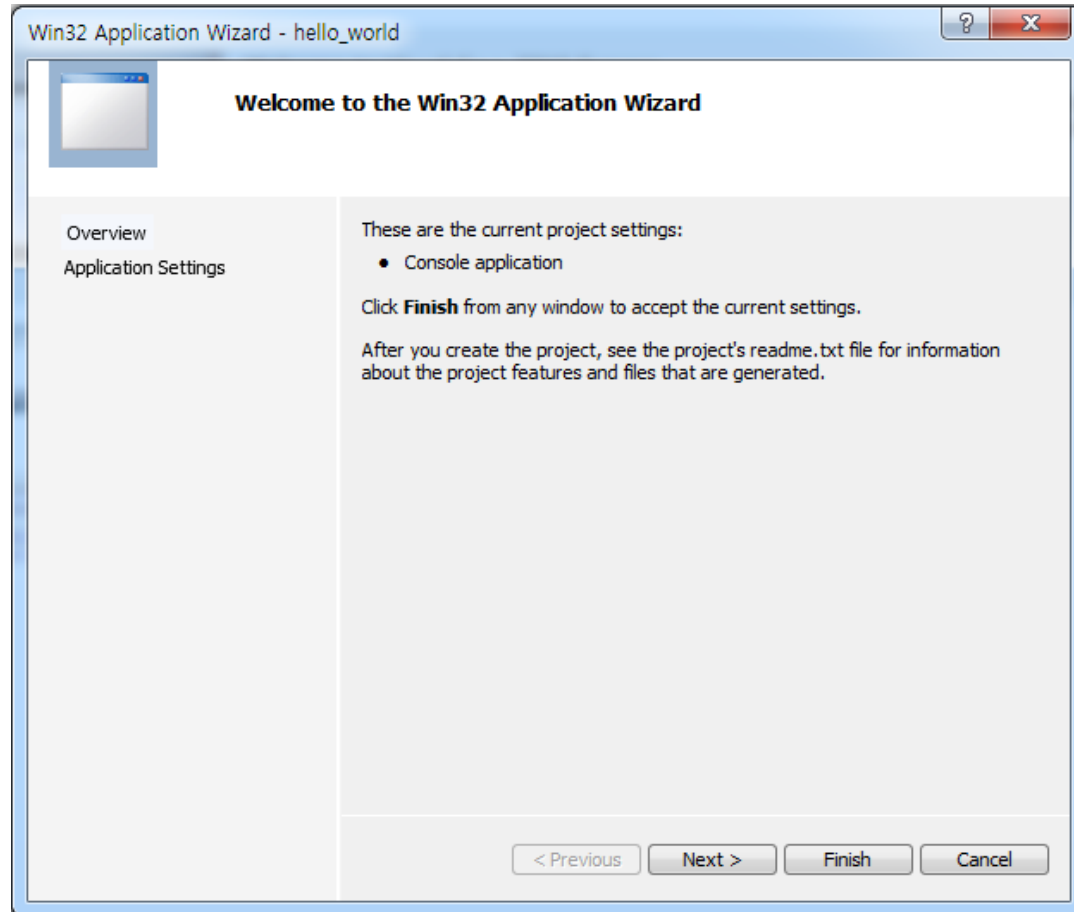
How to use Visual Studio



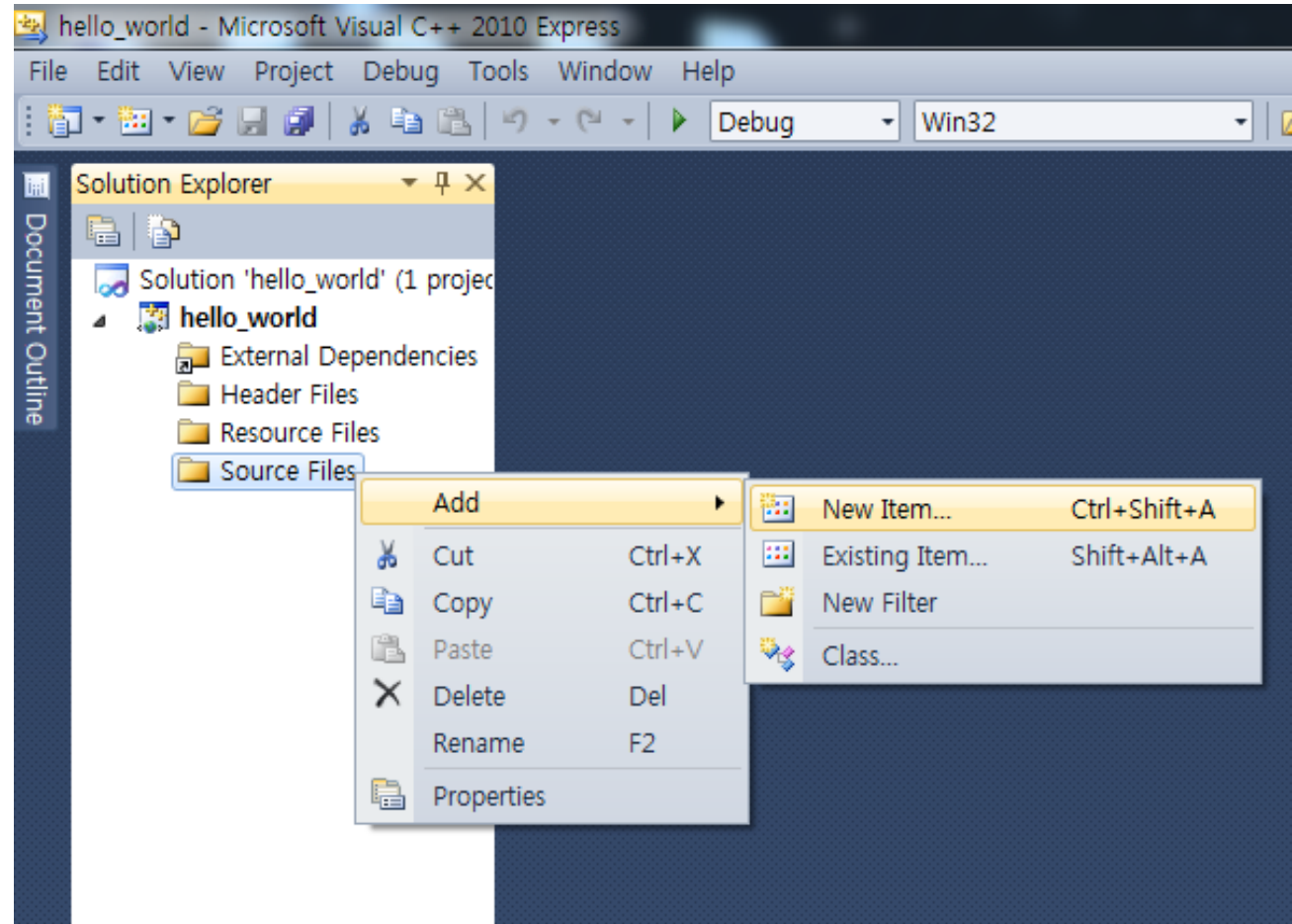
How



How to use Visual Studio



How to use Visual Studio



How to use Visual Studio

