

# Homework Tutorial



Data Structure

---



# Requirements

---

- We use g++ 4.8.1 to compile all students' homework
- Your mission is to provide the correct implementation of all functions in the class that we have already defined.
- You need to implement your code in the class "Implement"



# Requirements

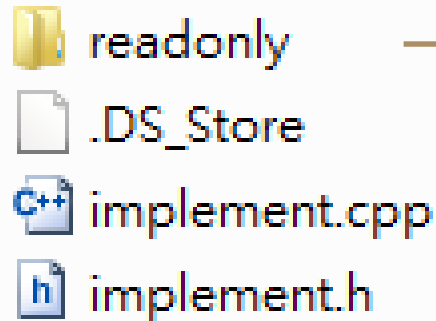
---

- In the different Homework we may asked you to use NTHU Online Judge to test/submit your Homework.
- We will introduce NTHU Online Judge next week.



# File Format

In the file, you can see following things:



The file “adt.h” and our testing lib are put in here, don’t try to modified it.

Project file, open it and write your code.  
The **implement.h** and **implement.cpp** are put in here.



# Methods You Need to Implement

```
implement.cpp  implement.h x  adt.h
(未知的範圍)
#ifndef HW1CPP_IMPLEMENT_H
#define HW1CPP_IMPLEMENT_H

#include "readonly/adt.h"

// Note that you should NOT modify the name of the class "Implement",
class Implement : public HWCPP_ADT::ADT
{
public:

    //write your code in implement.cpp
    const char* getStudentID();

    //write your code in implement.cpp
    int sum(int , int );

    //don't forget to implement the remaining swap functions.
    void swap(int *, int *);

    /**.....*/
    * This function will swap the value of its parameters
    /**.....*/
    void swap(int &, int &);
};

#endif
```

Implement these methods  
in implement.cpp



# How to Test Your Code?

---

- We provide a basic testing file, you can use it to evaluate correctness of your code.
- Make sure your code can pass basic testing.



# How to Test Your Code?

```
*
*  DS HW 1 - testing source file
*
*  YOU SHOULD NOT CHANGE THE CONTENT OF THIS FILE
*
*  if your code is correct implemented, you will see string "pass" in the console
*  if your code is not implemented well,
*      you will see the string "***not pass basic test**" in the console
*
/**-----*/

#include "implement.h"

int main(){

    //instance your implementation of ADT
    Implement implementObj;

    //our testing function, you don't need to care its content.
    hw1ADT::eval(implementObj);

    return 0;
}
```



# Testing Code Example

```
void hw1ADT::eval( ADT & adtInstance )
{
    try{

        int x=10;
        int y=20;

        adtInstance.swap(x,y);
        if (x!=20 || y!=10)
        {
            throw "swap(x,y) is not correct";
        }

        adtInstance.swap(&x,&y);
        if (x!=10 || y!=20)
        {
            throw "swap(x,y) is not correct";
        }

        //....
    }
```





---

Don't try to modified the following file,  
we may replace them and use stricter  
testing.

- `adt.h`
- `main.cpp`
- Name of class "Implement"



- 
- After finishing your code, you can use makefile to help you evaluate your code.
  - We will introduce how to run g++ 4.8.1 , and how to use makefile in Windows and Linux (Ubuntu12.04)



# In Windows (Step 1)

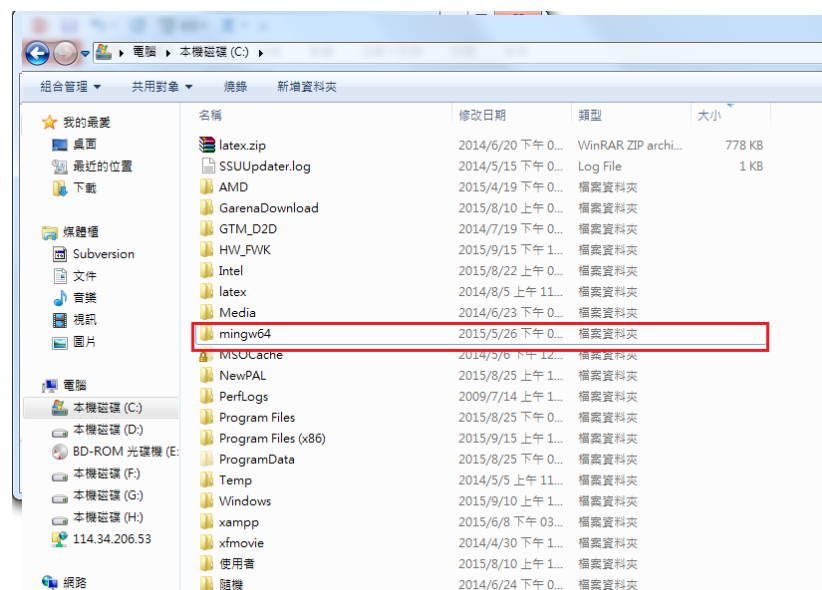
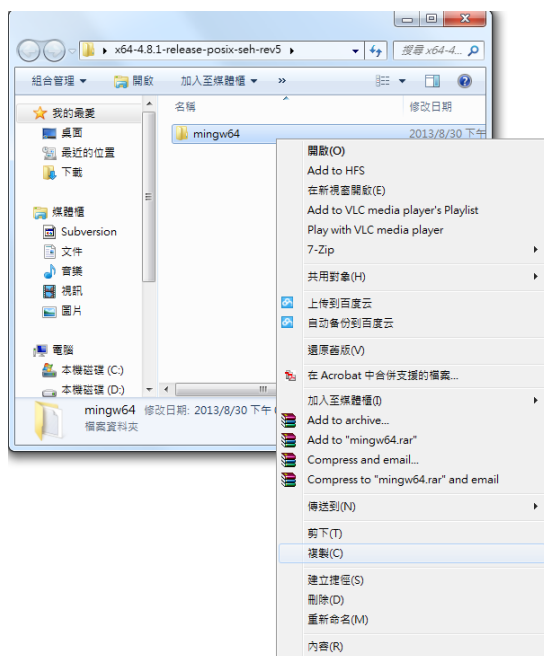
---

- According to your OS version, download the corresponding package
  - 32 bit
    - <http://sourceforge.net/projects/mingwbuilds/files/host-windows/releases/4.8.1/32-bit/threads-posix/dwarf/x32-4.8.1-release-posix-dwarf-rev5.7z/download>
  - 64 bit
    - <http://sourceforge.net/projects/mingwbuilds/files/host-windows/releases/4.8.1/64-bit/threads-posix/seh/x64-4.8.1-release-posix-seh-rev5.7z/download>



# In Windows (Step 2)

- After decompressing the downloaded file, copy the mingw32 (or mingw64) folder to C:\





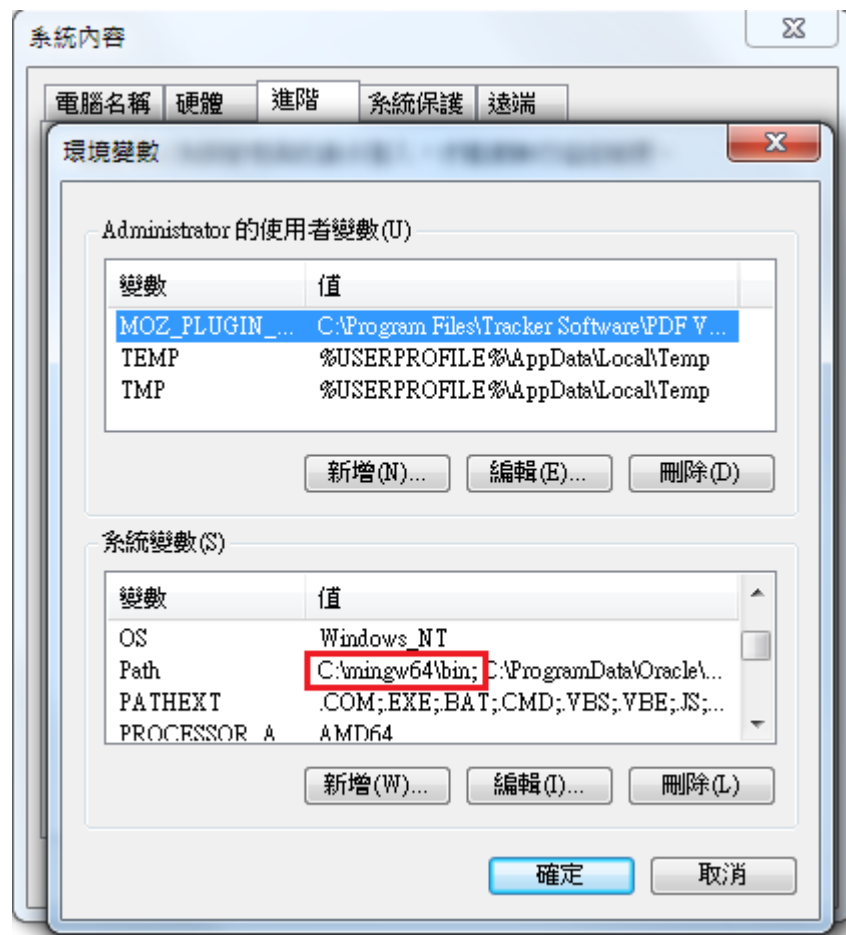
## In Windows (Step 3)

---

- Computer [right button]->Properties->Advanced system settings->Advanced->Environment Variables->System Variables->Path->add the instruction `C:\mingw32\bin;` or `(C:\mingw64\bin;)`
- Now you can use `g++ 4.8.1` in cmd



# For 64 bit as an example(Win7)





# For 64 bit as an example(Win10)

環境變數

Stan Lu 的使用者變數(U)

變數	值
OneDrive	C:\Users\Stan Lu\OneDrive
Path	C:\Users\Stan Lu\Anaconda2;C:\Users\Sta...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

新增(N)... 編輯(E)... 刪除(D)

系統變數(S)

變數	值
Path	C:\ProgramData\Oracle\Java\javapath;C:...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;...
PROCESSOR_AR...	AMD64
PROCESSOR_IDE...	Intel64 Family 6 Model 94 Stepping 3, Ge...

新增(W)... 編輯(I)... 刪除(L)

確定 取消

編輯環境變數

Stan Lu 的使用者變數(U)

系統變數(S)

變數名稱

變數值

新增(N) 編輯(E) 瀏覽(B)... 刪除(D) 上移(U) 下移(O) 編輯文字(T)...

確定 取消



# For 64 bit as an example

- Start Menu->search "cmd"->enter "gcc -v" or "g++ -v" (If it is successful, the cmd would show gcc version

4.

```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=c:/mingw64/bin/./libexec/gcc/mingw32/4.8.1/lto-wrapper.exe
Target: mingw32
Configured with: ../gcc-4.8.1/configure --prefix=/mingw --host=mingw32 --build=m
ingw32 --without-pic --enable-shared --enable-static --with-gnu-ld --enable-lto
--enable-libssp --disable-multilib --enable-languages=c,c++,fortran,objc,obj-c++
,ada --disable-sjlj-exceptions --with-dwarf2 --disable-win32-registry --enable-l
ibstdcxx-debug --enable-version-specific-runtime-libs --with-gmp=/usr/src/pkg/gm
p-5.1.2-1-mingw32-src/bld --with-mpc=/usr/src/pkg/mpc-1.0.1-1-mingw32-src/bld --
with-mpfr= --with-system-zlib --with-gnu-as --enable-decimal-float=yes --enable-
libgomp --enable-threads --with-libiconv-prefix=/mingw32 --with-libintl-prefix=/
mingw --disable-bootstrap LDFLAGS=-s CFLAGS=-D_USE_32BIT_TIME_T
Thread model: win32
gcc version 4.8.1 (GCC)

C:\Users\Administrator>
```





# In Windows (makefile)

- Cmd->goto the directory of makefile  
(*Note:* cd: change directory  
dir: for file and directory listing)

```
系統管理員: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\

C:\>cd HW_FWK\readonly

C:\HW_FWK\readonly>dir
磁碟區 C 中的磁碟沒有標籤。
磁碟區序號: C024-9FBE

C:\HW_FWK\readonly 的目錄
2015/09/17 下午 04:39 <DIR> .
2015/09/17 下午 04:39 <DIR> ..
2015/08/26 上午 01:01          1,288 adt.h
2015/09/15 下午 10:43       4,982 libmylib.a
2015/08/26 上午 12:57          118 main.cpp
2015/09/15 下午 10:42          246 makefile
                4 個檔案          6,634 位元組
                2 個目錄      14,244,421,632 位元組可用

C:\HW_FWK\readonly>
```



# In Windows

- Enter “**mingw32-make -f makefile**”  
and it will create a executable file of  
main.cpp.

```
C:\Windows\system32\cmd.exe
C:\Users\Administrator>cd C:\
C:\>cd HW_FWK\readonly
C:\HW_FWK\readonly>dir
磁碟區 C 中的磁碟沒有標籤。
磁碟區序號: C024-9FBE

C:\HW_FWK\readonly 的目錄

2015/09/17 下午 04:39 <DIR>      .
2015/09/17 下午 04:39 <DIR>      ..
2015/08/26 上午 01:01           1,288 adt.h
2015/09/15 下午 10:43           4,982 libmylib.a
2015/08/26 上午 12:57             118 main.cpp
2015/09/15 下午 10:42             246 makefile
                4 個檔案          6,634 位元組
                2 個目錄      14,173,978,624 位元組可用

C:\HW_FWK\readonly>mingw32-make -f makefile
g++ -ln -std=c++11 -c ../*.cpp
g++ -ln -std=c++11 -c ../*.cpp
g++ -ln -std=c++11 ../*.cpp ../*.cpp ../*.a -o main

C:\HW_FWK\readonly>
```



# In Windows

- Finally, run the executable file (**main**) in this case, cmd will show the result

```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\

C:\>cd HW_FWK\readonly

C:\HW_FWK\readonly>mingw32-make -f makefile
g++ -lm -std=c++11 -c ./*.cpp
g++ -lm -std=c++11 -c ../*.cpp
g++ -lm -std=c++11 ../*.cpp ./*.cpp ./*.a -o main

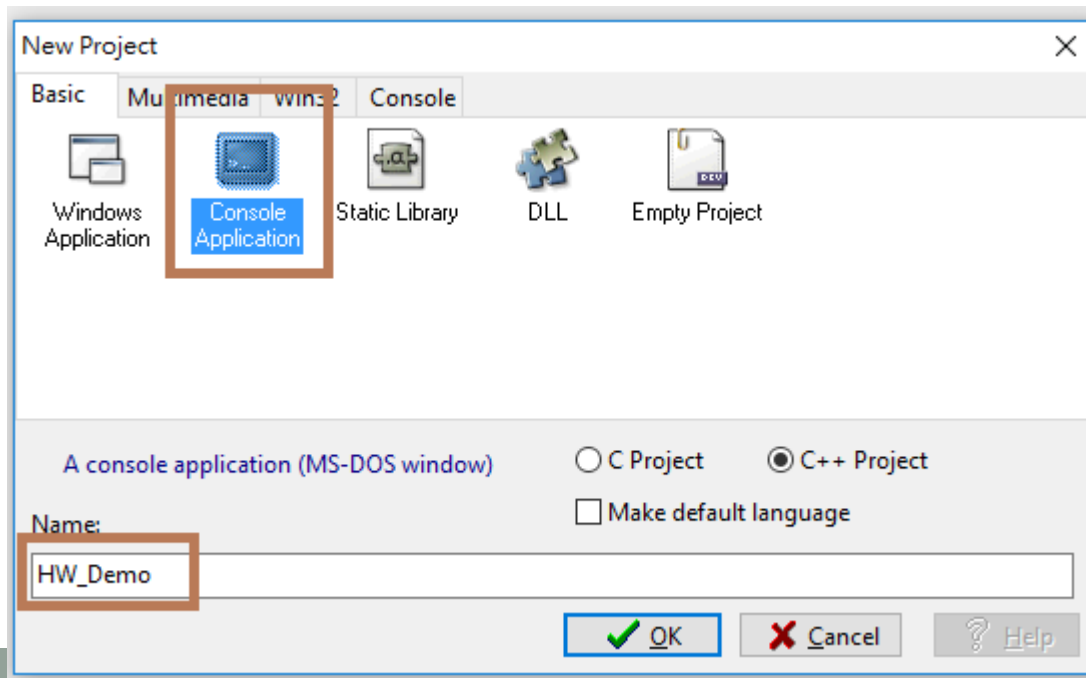
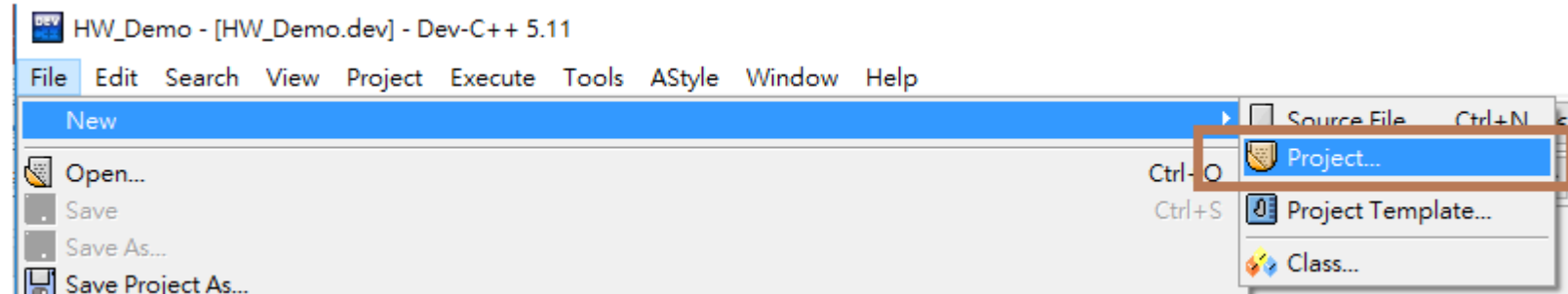
C:\HW_FWK\readonly>main
-- PASS --

C:\HW_FWK\readonly>

```

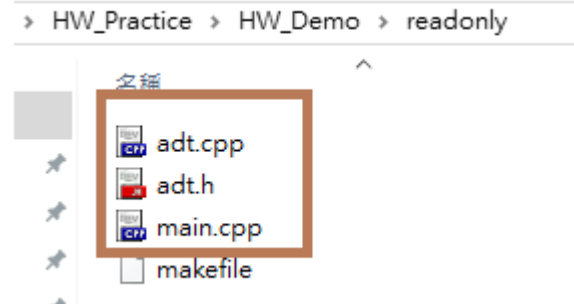
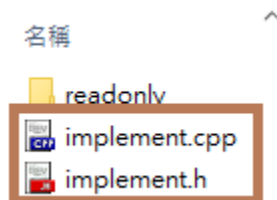
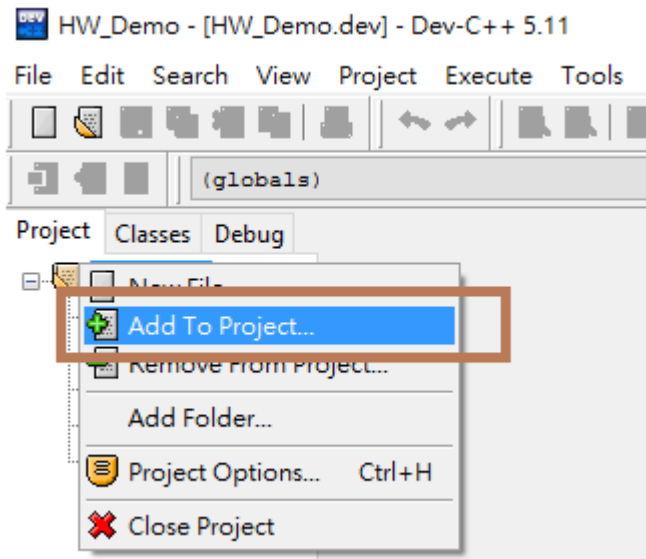


# Dev C++ ()





# Dev C++





# NotePad++

```
案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?
new 1 x index.php x new 2 x new 3 x new 4 x new 5 x makefile x implement.h x implement.cpp x
1 #ifndef HW1CPP_IMPLEMENT_H
2 #define HW1CPP_IMPLEMENT_H
3
4 #include "adt.h"
5
6 // Note that you should NOT modify the name of the class "Implement",
7 class Implement : public HWCPP_ADT::ADT
8 {
9
10 public:
11
12     //write your code in implement.cpp
13     const char* getStudentID();
14
15     //write your code in implement.cpp
16     int sum(int , int );
17
18     //don't forget to implement the remaining swap functions.
19     void swap(int *, int *);
20
21     /**-----*/
22     * This function will swap the value of its parameters
23     **-----*/
24     void swap(int &, int &);
25
26 private:
27     int id;
28
29
30 };
31
32 #endif
```



## In Ubuntu12.04

---

- If you want to use Linux (Ubuntu12.04), you can use following way to install the g++ 4.8.1.
- Open the terminal and run the instructions step by step as follows

```
sudo apt-get install libgmp-dev
sudo apt-get install libmpfr4 libmpfr-dev
sudo apt-get install libmpc-dev libmpc2
sudo apt-get install libtool
sudo apt-get install m4
sudo apt-get install bison
sudo apt-get install flex
sudo apt-get install autoconf
```



# In Ubuntu12.04

---

- `sudo add-apt-repository ppa:ubuntu-toolchain-r/test`  
`sudo apt-get update`

```
sudo apt-get install gcc-4.8
sudo apt-get install g++-4.8
sudo apt-get install gcc-4.8-multilib
sudo apt-get install g++-4.8-multilib
sudo apt-get install gcc-4.8-doc
```

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 20
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.8 20
```

```
sudo update-alternatives --config gcc
sudo update-alternatives --config g++
```

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get dist-upgrade
```





# In Ubuntu12.04

Ubuntu - VMware Player (Non-commercial use only)

Player

终端機

```
hello@hello-VirtualBox: ~  
hello@hello-VirtualBox:~$ gcc -v  
Using built-in specs.  
COLLECT_GCC=gcc  
COLLECT_LTO_WRAPPER=/usr/lib/gcc/i686-linux-gnu/4.8/lto-wrapper  
Target: i686-linux-gnu  
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 4.8.1-2ubuntu1~12.04' --with-bugurl=file:///usr/share/doc/gcc-4.8/README.Bugs --enable-languages=c,c++,java,go,d,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-4.8 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/include/c++/4.8 --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-gnu-unique-object --enable-plugin --with-system-zlib --disable-browser-plugin --enable-java-awt=gtk --enable-gtk-cairo --with-java-home=/usr/lib/jvm/java-1.5.0-gcj-4.8-i386/jre --enable-java-home --with-jvm-root-dir=/usr/lib/jvm/java-1.5.0-gcj-4.8-i386 --with-jvm-jar-dir=/usr/lib/jvm-exports/java-1.5.0-gcj-4.8-i386 --with-arch-directory=i386 --with-ecj-jar=/usr/share/java/eclipse-ecj.jar --enable-objc-gc --enable-targets=all --enable-multiarch --disable-werror --with-arch-32=i686 --with-multilib-list=m32,m64 --with-tune=generic --enable-checking=release --build=i686-linux-gnu --host=i686-linux-gnu --target=i686-linux-gnu  
Thread model: posix  
gcc version 4.8.1 (Ubuntu 4.8.1-2ubuntu1~12.04)  
hello@hello-VirtualBox:~$
```



# In Ubuntu12.04

- Terminal->enter the directory of makefile  
(*Note:* cd: change directory  
ls: for file and directory listing)

```
hello@hello-VirtualBox: ~/HW_FWK/readonly
hello@hello-VirtualBox:~$ cd HW_FWK/readonly/
hello@hello-VirtualBox:~/HW_FWK/readonly$ ls
adt.h  libmylib.a  main.cpp  main.cpp~  makefile  makefile~
hello@hello-VirtualBox:~/HW_FWK/readonly$
```



# In Ubuntu12.04

- Enter “**make -f makefile**”  
and it will create a executable file of  
main.cpp.

```
hello@hello-VirtualBox: ~/HW_FWK/readonly
hello@hello-VirtualBox:~$ ls
123      eclipse                               tarb
123~     eclipse-cpp-luna-R-linux-gtk.tar.gz  tarballs
adt.cpp  example                             vmware-tools-distrib
adt.h    examples.desktop                   workspace
adt.o    HW_FWK                             下載
hello@hello-VirtualBox:~$ cd HW_FWK/readonly/
hello@hello-VirtualBox:~/HW_FWK/readonly$ make -f makefile
g++-4.8 -lm -std=c++11 -c ./*.cpp
g++-4.8 -lm -std=c++11 -c ../*.cpp
g++-4.8 -lm -std=c++11 ../*.cpp ./*.cpp ./*.a -o main
hello@hello-VirtualBox:~/HW_FWK/readonly$
```



# In Ubuntu12.04

- Finally, run the executable file (**./main**) in this case terminal will show the result

```
hello@hello-VirtualBox: ~/HW_FWK/readonly
hello@hello-VirtualBox:~$ ls
123      eclipse          tarb
123~     eclipse-cpp-luna-R-linux-gtk.tar.gz tarballs
adt.cpp  example          vmware-tools-distrib
adt.h    examples.desktop workspace
adt.o    HW_FWK           下載
hello@hello-VirtualBox:~$ cd HW_FWK/readonly/
hello@hello-VirtualBox:~/HW_FWK/readonly$ make -f makefile
g++-4.8 -lm -std=c++11 -c ./*.cpp
g++-4.8 -lm -std=c++11 -c ../*.cpp
g++-4.8 -lm -std=c++11 ../*.cpp ./*.cpp ./*.a -o main
hello@hello-VirtualBox:~/HW_FWK/readonly$ ./main
-- PASS --
hello@hello-VirtualBox:~/HW_FWK/readonly$
```



# Demo

---

1. Install g++ 4.8.1.
2. Compile used Console.
3. Compile used Dev C++.