C++ For C Coders 1

Data Structures
C++ for C Coders

한동대학교 김영섭교수
idebtor@gmail.com

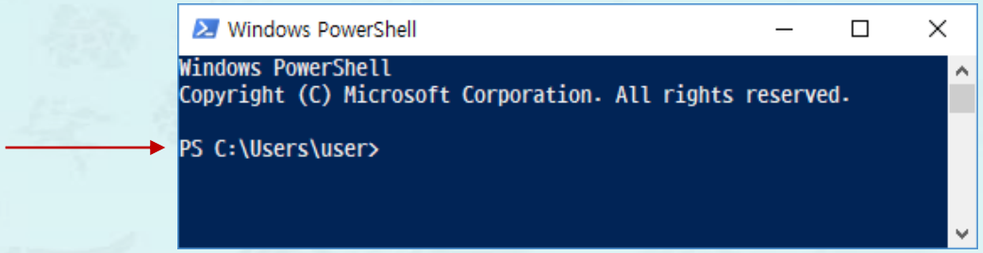
Start a console
PS, cmd, bash
Revisiting installation of g++
atom, git, github, github Desktop
C vs C++
In-house programming principles
Hello World

C++ for C Coders

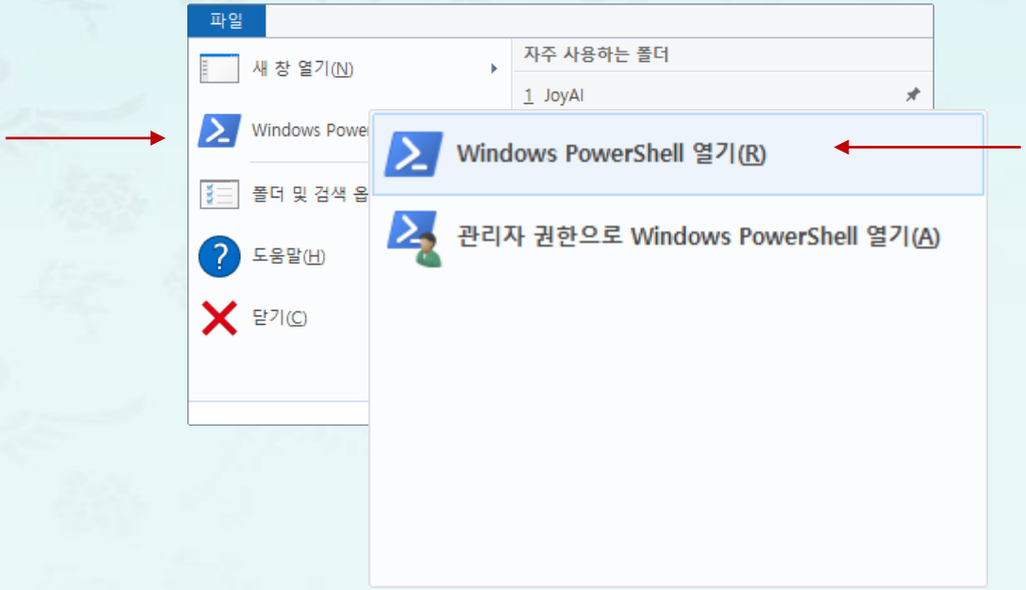
- Don't forget registering Piazza.com
- Reading material available in github.com/idebtor/nowic
  - **README**
  - **Syllabus**
  - **GettingStarted**

Start a console

1. Start PowerShell through "Start" menu

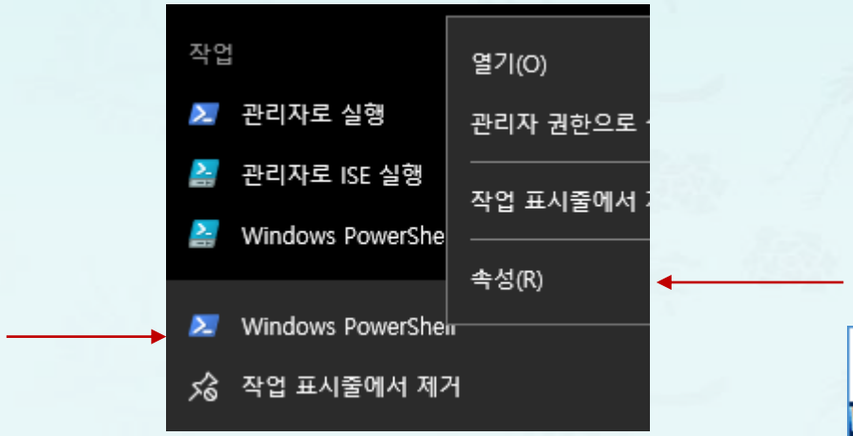


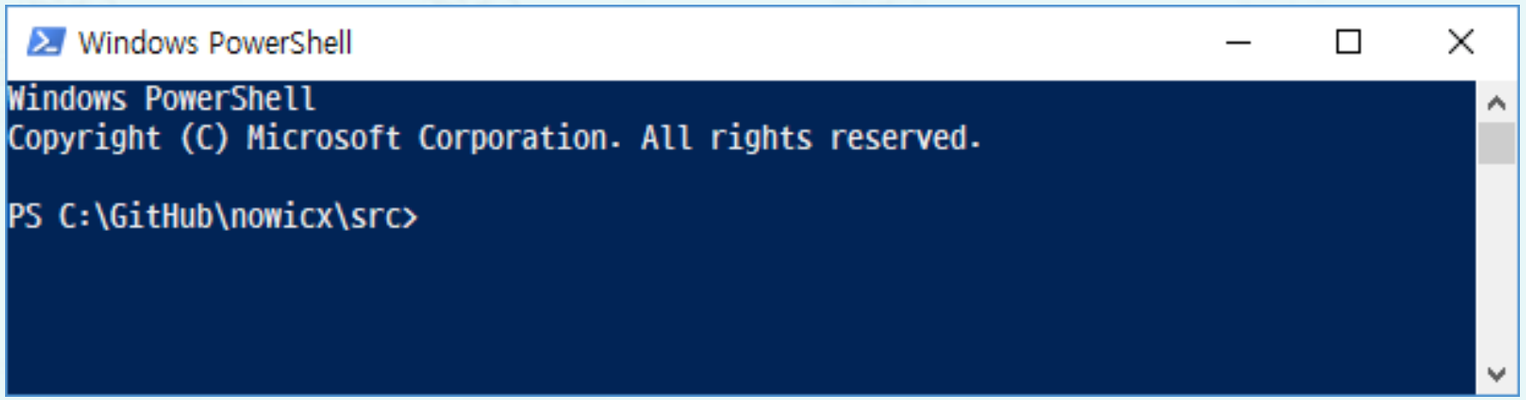
2. Move to the folder you want to start.
   - Click the "File" menu at the top left corner o
   - PowerShell or cmd windows
   - Enter a command at console. For example, **atom myfile.txt**



Start a console at your favorite folder

1. Start PowerShell if you don't have a PowerShell icon in Taskbar.
2. In the Taskbar right-click and pin to keep a link there.
3. **Again right click the icon in taskbar** and then right-click Windows PowerShell and choose Properties
4. Enter your preferred directory in the Start in: input field and press OK.
5. Start from the taskbar icon





PowerShell, Cmd, vs Bash (Borne-again shell), sh, ksh, csh

PowerShell (Cmdlet)	PowerShell (Alias)	Windows Cmd	Unix shell	Description
Get-ChildItem	gci, dir, ls	dir	ls	Lists all files and folders in the current or given folder
Get-Content	gc, type, cat	type	cat	Gets the content of a file
Get-Command	gcm	help	type, which	Lists available commands
Get-Help	help, man	help	man	Prints a command's documentation on the console
Clear-Host	cls, clear	cls	clear	Clears the screen
Copy-Item	cpi, copy, cp	copy, xcopy	cp	Copies files and folders to another location
Move-Item	mi, move, mv	move	mv	Moves files and folders to a new location
Remove-Item	ri, del, rmdir, rd, rm	del, rmdir, rd	rm, rmdir	Deletes files or folders
Rename-Item	rni, ren, mv	ren, rename	mv	Renames a single file, folder, hard link or symbolic link
Get-Location	gl, cd, pwd	cd	pwd	Displays the working path (current folder)
Pop-Location	popd	popd	popd	Changes the working path to the location most recently pushed onto the stack
Push-Location	pushd	pushd	pushd	Stores the working path onto the stack
Set-Location	sl, cd, chdir	cd, chdir	cd	Changes the working path, cd .. (move to parent folder), cd ~ (move to HOME folder)
Write-Output	echo, write	echo	echo	Prints strings or other objects to the standard output
Get-Process	gps, ps	tlist, tasklist	ps	Lists all running processes
Stop-Process	spps, kill	kill, taskkill	kill	Stops a running process
Select-String	sls	findstr	find, grep	Prints lines matching a pattern
Set-Variable	sv, set	set	env, export, set, setenv	Creates or alters the contents of an environment variable

Install GNU C/C++ Compiler (MSYS2, MinGW-w64)

- MSYS = "**M**inimal **SYS**tem", is a Bourne Shell command line interpreter system
  MinGW = "**Min**imalist **G**NU for **W**indows“

- Install MSYS2 first from **http://www.msys2.org/**
  Then use a command at the msys2 window.

```
$ pacman -Syu
$ pacman -S mingw-w64-x86_64-gcc
```

- Add the following path to the PATH environment variable

```
C:/msys64/mingw64/bin                    # for g++
C:/msys64/usr/bin                        # for commands such as ls, cat, rm
```

be cautious, not 'user'

- Add .bash_profile

```
C:/msys64/home/user/.bash_profile        # a sample is provided
```

check your home folder

.bash_profile (an example) in C:\msys64\home\user

```
alias ls='ls -Gp --color=auto'
alias ll='ls -alkF'
alias rm='rm -i'
alias h='history'
LS_COLORS=$LS_COLORS':no=00:di=36;01'
LS_COLORS=$LS_COLORS':*.h=1;33:*.exe=31:*.o=1;32:*.md=1;33'
export LS_COLORS

export PATH=$PATH:"C:\msys64\mingw64\bin"
export PATH=$PATH:"C:\msys64\usr\bin"
export PATH=$PATH:"C:\Users\$USER\AppData\Local\atom\bin"
echo $PWD

# Setting my dev folder as a startup folder of msys.
HOME="/c/GitHub/nowic"
cd $HOME
echo $PWD

# @$(hostname) may be added, if necessary, after $(whoami)
GREEN="$(tput setaf 2)"
RESET="$(tput sgr0)"
PS1='${GREEN}$(pwd -W)> ${RESET}'
```

This is the folder Atom is installed.

check your github folder

Mac OSX uses .zshrc instead of .bash_profile

Open a terminal & run the following commands
참고: https://qastack.kr/superuser/886132/where-is-the-zshrc-file-on-mac
> **touch ~/.zshrc** # 폴더와 파일 생성
> **open ~/.zshrc** # 파일 열고, .bash_profile내용 입력
Close the file and reopen a terminal.

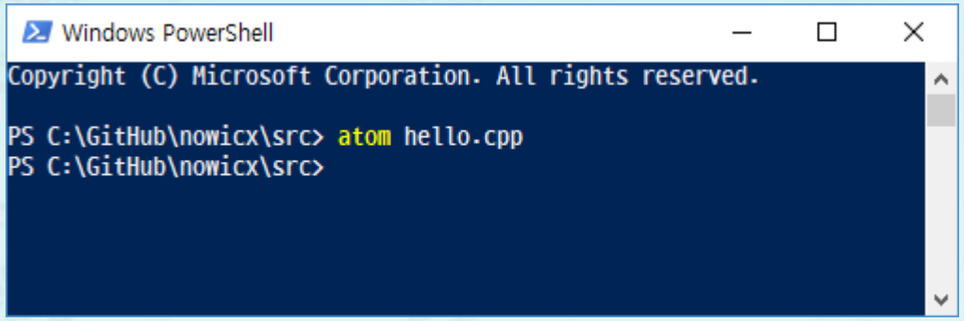
```
alias ls='ls -Gp --color=auto'
alias ll='ls -alkF'
alias rm='rm -i'
alias h='history'
LS_COLORS=$LS_COLORS':no=00:di=36;01'
LS_COLORS=$LS_COLORS':*.h=1;33:*.exe=31:*.o=1;32:*.md=1;33'
export LS_COLORS

export PATH=$PATH:"C:\msys64\mingw64\bin"
export PATH=$PATH:"C:\msys64\usr\bin"
export PATH=$PATH:"C:\Users\$USER\AppData\Local\atom\bin"
echo $PWD

# Setting my dev folder as a startup folder of msys.
HOME="/c/GitHub/nowic"
cd $HOME
echo $PWD

# @$(hostname) may be added, if necessary, after $(whoami)
GREEN="$(tput setaf 2)"
RESET="$(tput sgr0)"
PS1='${GREEN}$(pwd -W)> ${RESET}'
```

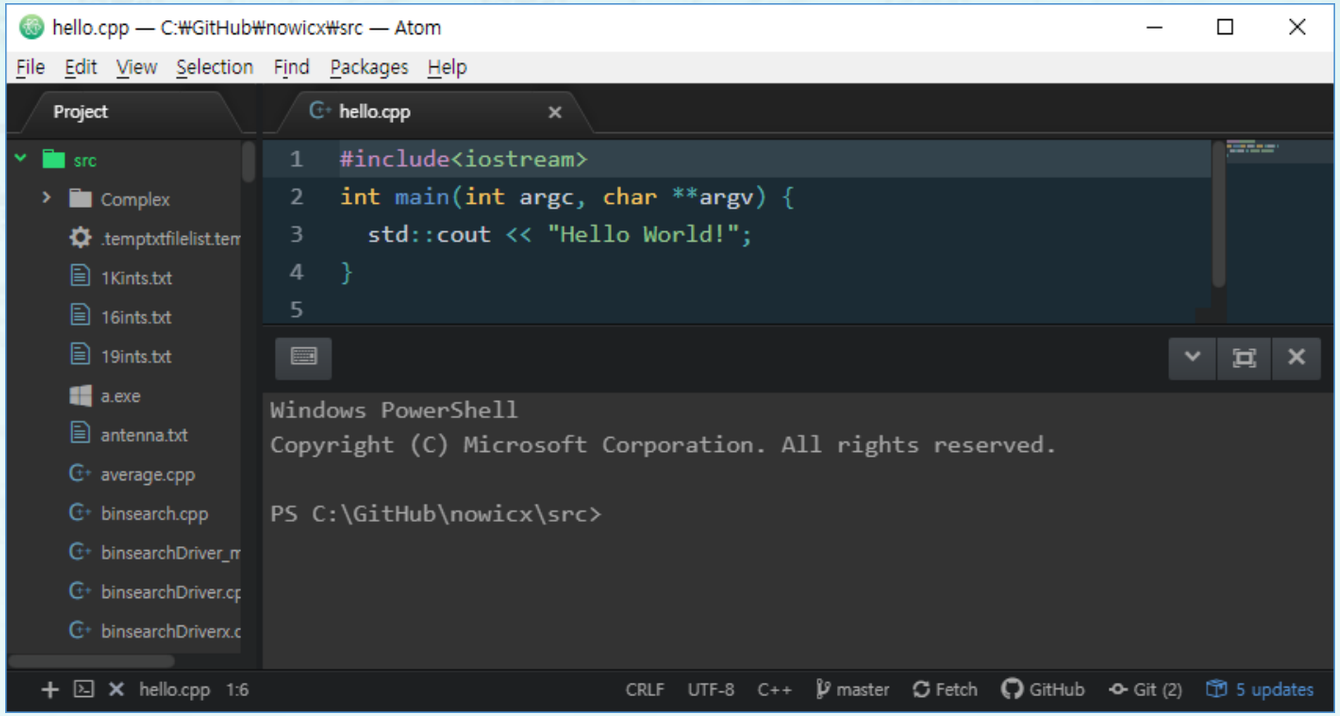
This is the folder Atom is installed.

check your github folder

Install Atom

1. Install **Atom.**
2. Start Atom to edit a text file at a console.
   - start a console (or console window)
   - select Windows PowerShell
   - start the following command at console
     atom hello.cpp
3. Add the following packages
   - platformio-ide-terminal
   - file-icons
   - minimap
   - **autocomplete-clang**





Install Git and GitHub Desktop

- **Git**
  - **a revision control system,**
  - **a tool to manage your source code history**
- **Github**
  - **a hosting service for Git repositories.**
- **Github Desktop**
  - **a Windows interface for Git/GitHub,**
  - **a subset of Git commands supported**

1. Install Git.
2. Install Github Desktop
3. Clone the following repository
   - github.com/idebtor/nowic

C++ for C Coders

- C
- Dennis Ritchie
- 1972
- 29 Keywords
  - Imperative programming
  - Procedural programming

- C++
- Stroustrup
- 1985
- 63 Keywords by 1996
- C++ as a better C
  - Imperative programming
  - Object-oriented programming
  - **Generic programming**

"Swiss Army Knife"

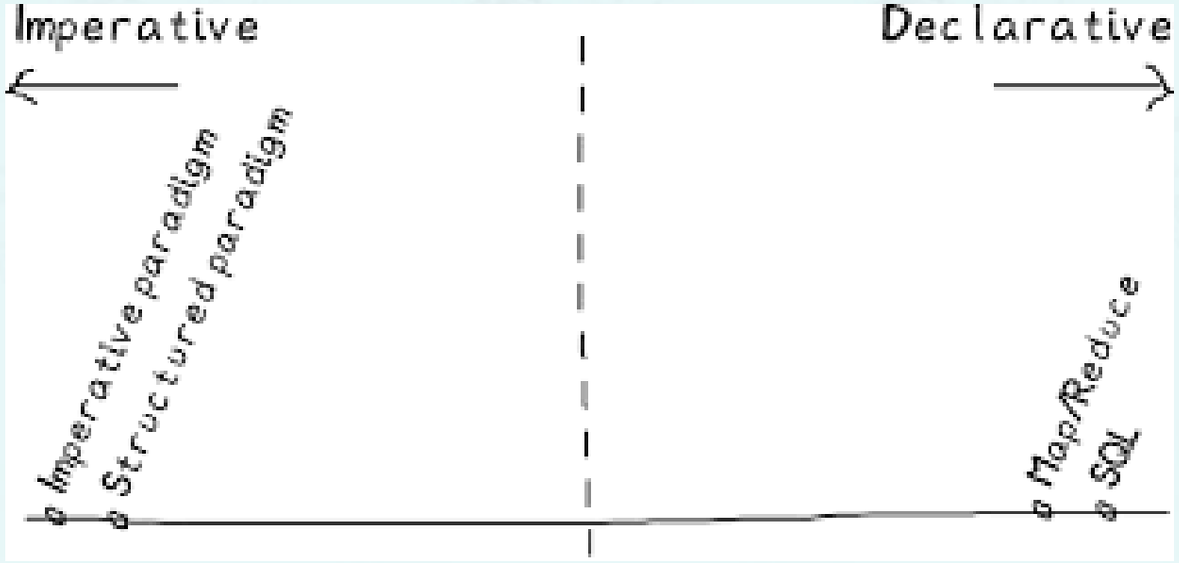
Imperative Declarative programming

명령형(C,C++,Java) vs 선언형(Prolog, Haskell, SQL)

Imperative(명령형) vs Declarative(선언형) Programming

- The difference between these is generally described as such: declarative languages **tell the computer what to do**, while imperative languages **tell the computer how to do it**.
- Obviously, no languages completely fits into one side of this spectrum, but traditional Object-Oriented languages are generally considered to be more imperative than functional languages

- C,C++,Java



- Prolog, Haskell, SQL

In-house Programming Principles

- DRY or DIE
  - Don't repeat yourself, or Duplication is Evil.
  - To make you professional coders and save time.
- KISS
  - Keep it simple, stupid!, Keep it simple and short.
  - To make you professional coders – code maintainable
- NMN
  - No magic number
  - To make you professional coders - code robust.
- NSE
  - No side effects
  - To make you professional coders - code clean.

C++ specific keywords

```
asm          dynamic_cast  namespace  reinterpret_cast  try
bool         explicit      new        static_cast       typeid
catch        false         operator   template          typename
class        friend        private    this              using
const_cast   inline        public     throw             virtual
delete       mutable       protected  true              wchar_t
```

From C to C++

- Even if C++ is slanted toward object-oriented programming (OOP), you can nevertheless use any C++ compiler to compile C code and benefits from some C++ goodies.

- C++ as a better C

Write "Hello World" program in C++

- Open Atom editor

```
$ atom hello.cpp
```

  - Add the source code.
  - Save the file.

- Compile and Execute

```
$ g++ hello.cpp
$ ./a.exe
$ ./a
$ a
```

- Compile and Execute

```
$ g++ hello.cpp -o hello
$ ./hello.exe
$ ./hello
$ hello
```

```
// file: hello.cpp
#include <iostream>

int main() {
  std::cout << "Hello World!" << std::endl;
}
```

```
// file: hello.cpp
#include <iostream>
using namespace std;

int main() {
  cout << "Hello World!" << endl;
}
```

(1)

C++ For C Coders 1

Data Structures
C++ for C Coders

한동대학교 김영섭교수
idebtor@gmail.com

Start a console
PS, cmd, bash
Revisiting installation of g++
atom, git, github, github Desktop
C vs C++
In-house programming principles
Hello World