

# 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

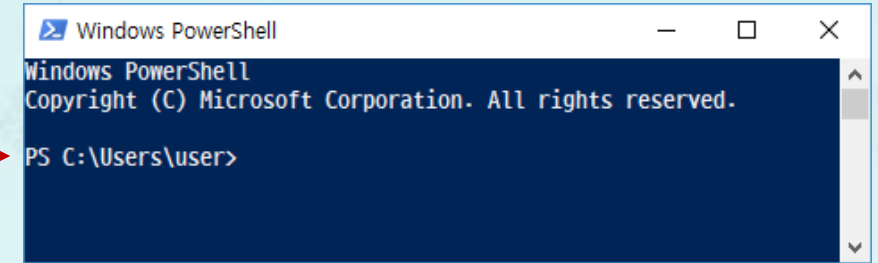
# C++ for C Coders

---

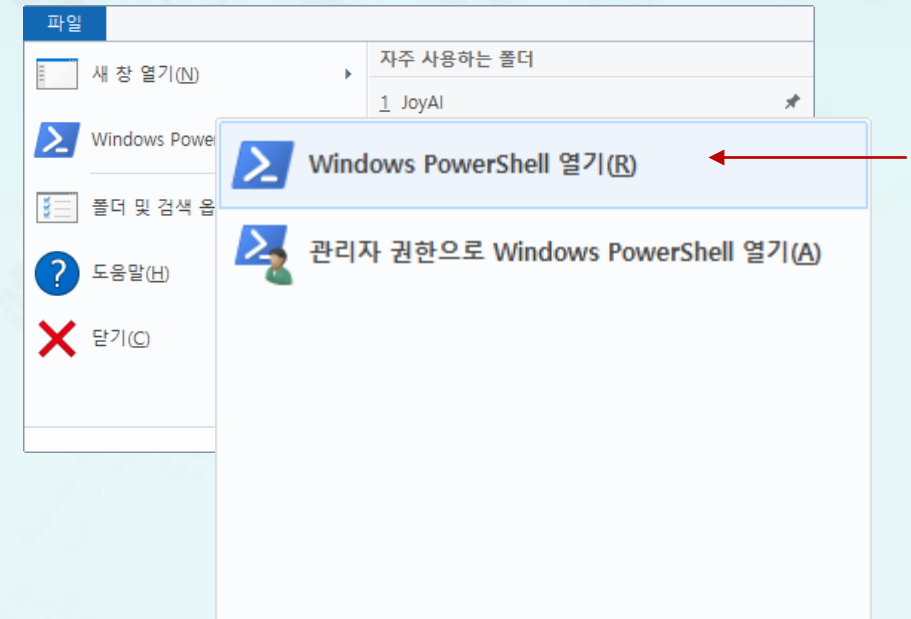
- Don't forget registering Piazza.com
- Reading material available in [github.com/idebtor/nowic](https://github.com/idebtor/nowic)
  - **README**
  - **ECE20010 Syllabus**
  - **GettingStarted**

# Start a console

1. Start PowerShell through "Start" menu

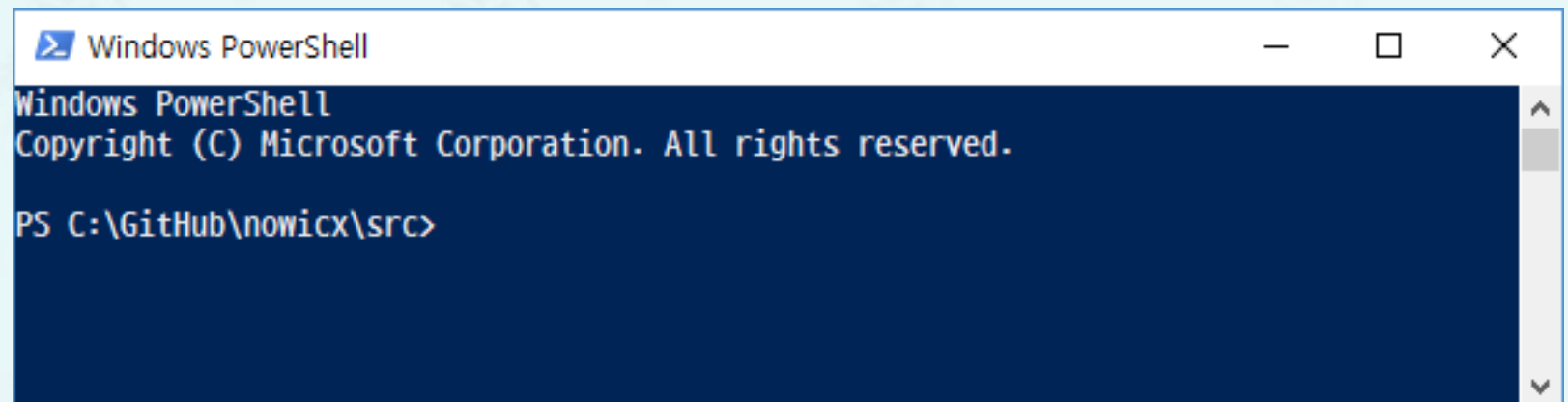
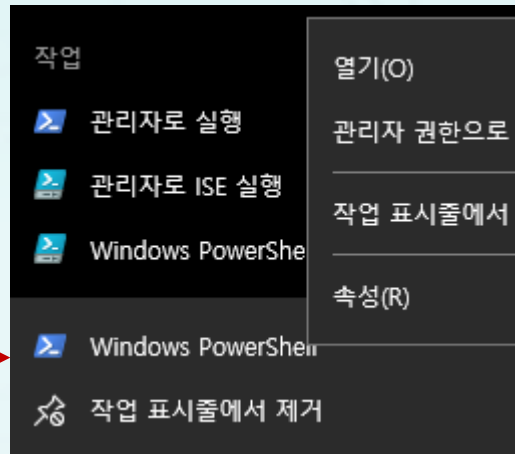


2. Move to the folder you want to start.
  - Click the "File" menu at the top left corner of PowerShell or cmd windows
  - Enter a command at console. For example, **atom myfile.txt**



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

↙ This is the folder Atom is installed.

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

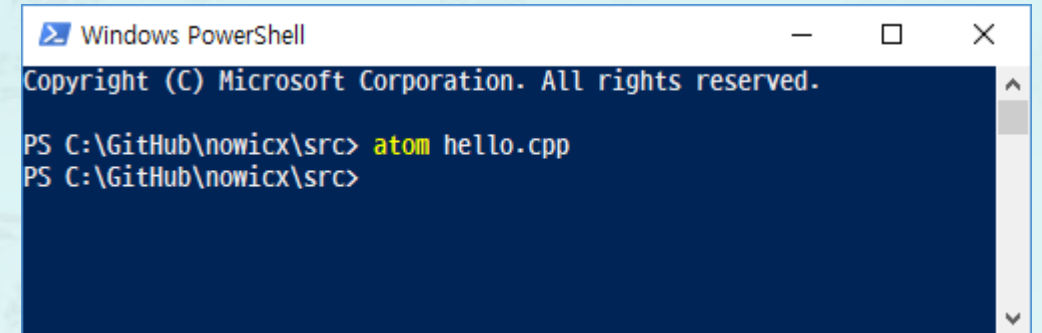
↙ check your github folder

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



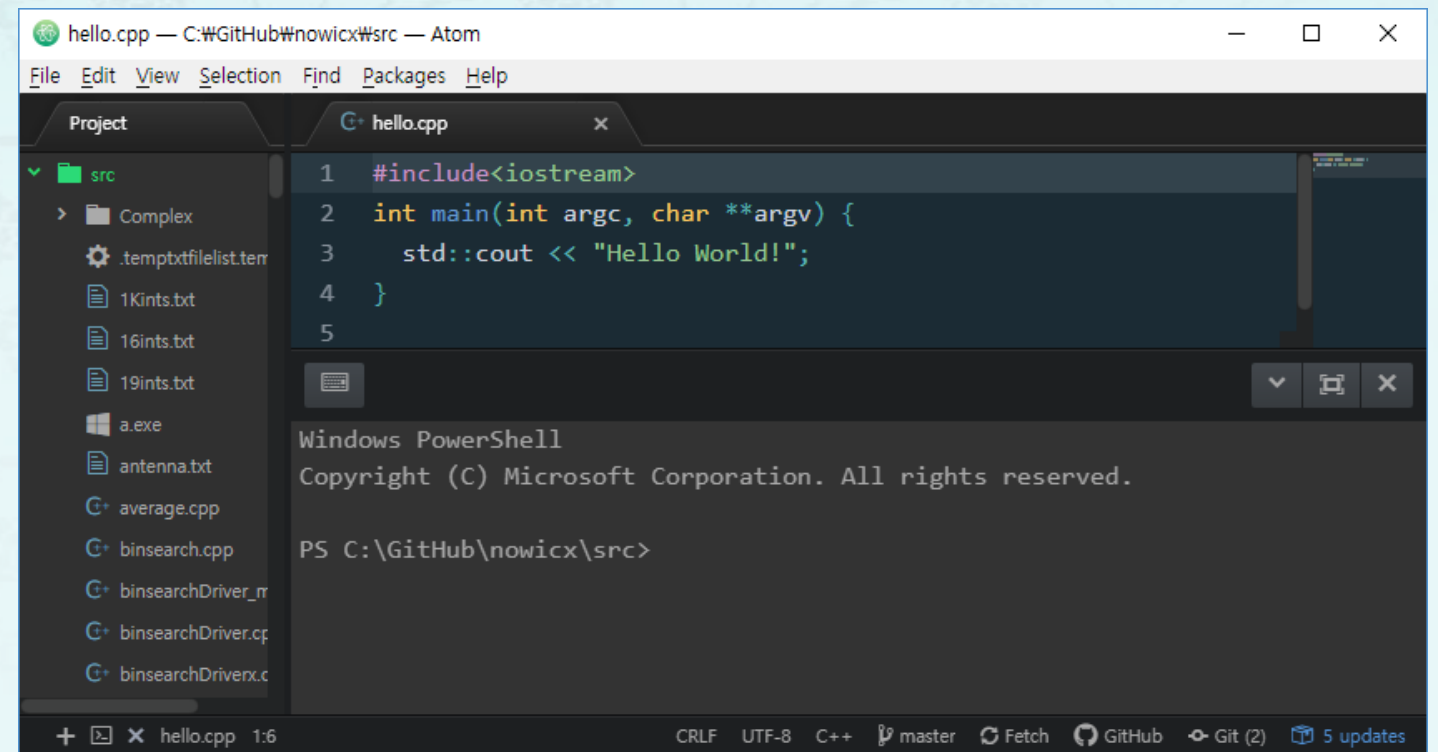
# Install Atom

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**

A screenshot of a Windows PowerShell window. The title bar says "Windows PowerShell". The text inside shows the copyright notice for Microsoft Corporation and the command "atom hello.cpp" being executed in the directory "C:\GitHub\nowicx\src".

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\GitHub\nowicx\src> atom hello.cpp
PS C:\GitHub\nowicx\src>
```





# 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](https://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”

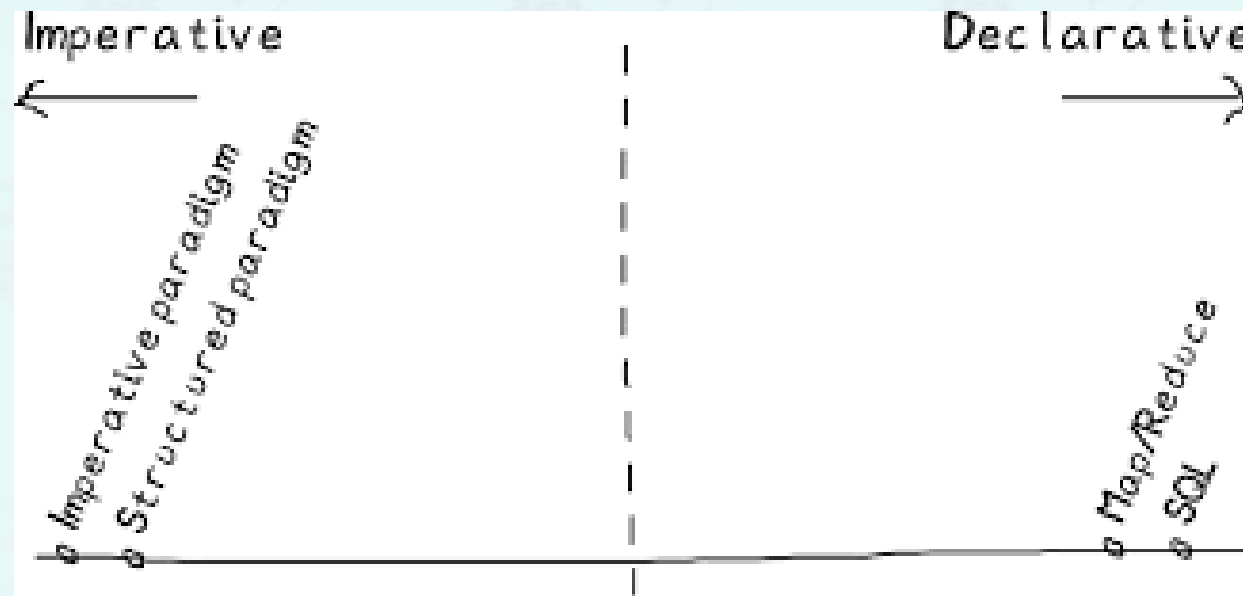
Imperative

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

---

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

## 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

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