

Compiler Specific Topics

Kate Gregory

www.gregcons.com/kateblog



Command Line Arguments

- The main function should really be:

```
int main(int argc, char* argv[])  
{  
    return 0;  
}
```

- Used when someone passes in parameters from the command prompt

> Small.exe -g Hello 12

> g++ a.cpp b.cpp

- Not a common technique these days
- `argv[]` is an old (C-style) way to do a collection
- `char*` is an old (C-style) way to do a string
- `argc` is the count of how many arguments were passed

Visual Studio

- **Pre compiled headers**
 - stdafx.h is the default name
- **Code that can support Unicode or non-Unicode with the flip of a switch**
 - _t
- **Extra files**
 - targetver.h
 - ReadMe.txt
- **#pragma once**

g++ and Other Compilers

- **Make files**
 - Can save typing a lot of file names and options
 - Can shorten build times by building only changed files
 - You maintain them by hand
- **Make is actually a separate tool that calls your compiler**
 - Therefore it works with any compiler that can be called from a command line
- **You may need to pass an option to specify C++ 11**
 - MinGW 10.2 and up has C++11 as default
 - Check the documentation for the compiler you're using

Summary

- Simple examples from textbooks aren't exactly like the code you meet in the real world
- Visual Studio generates projects for you and adds files you can remove or ignore when you're making a console application
- Many compiler distributions include the `make` utility to keep your build times to the minimum