

Help for dotnet Command

Jim Fawcett

CSE686 – Internet Programming

Spring 2019

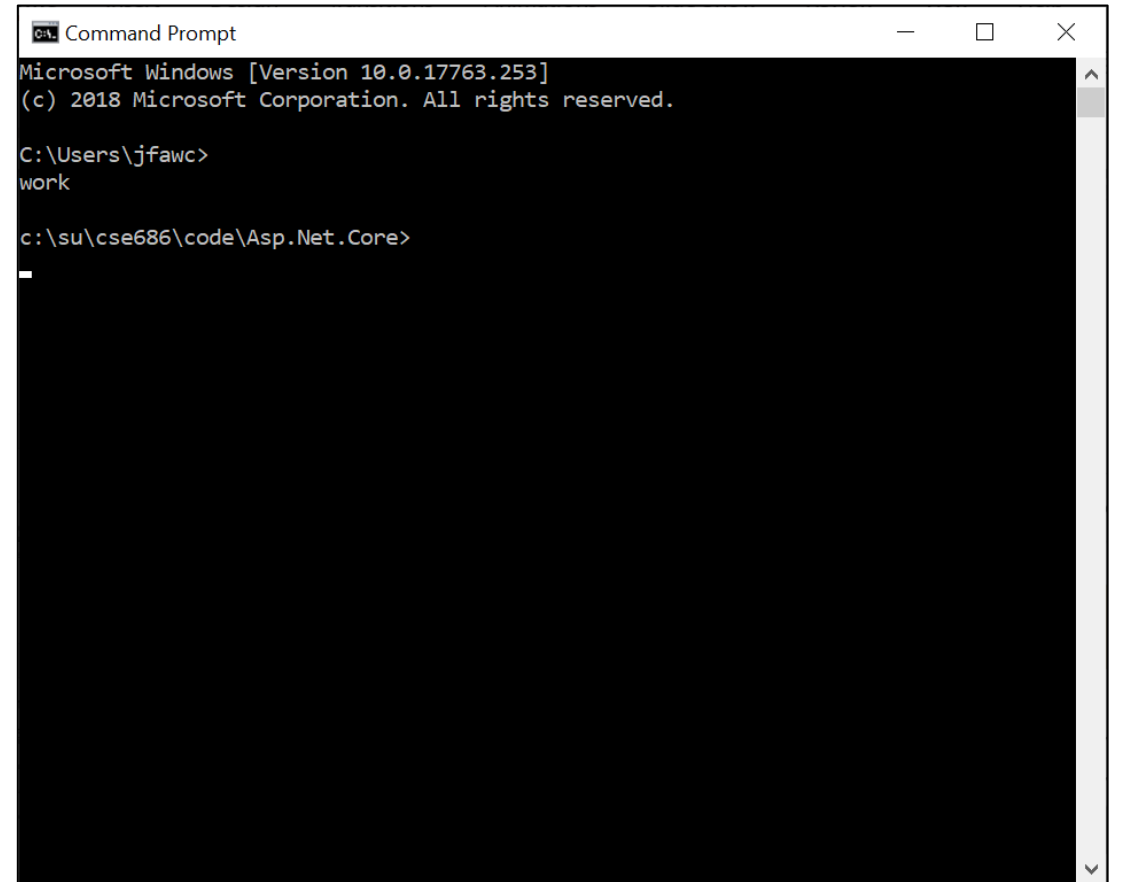
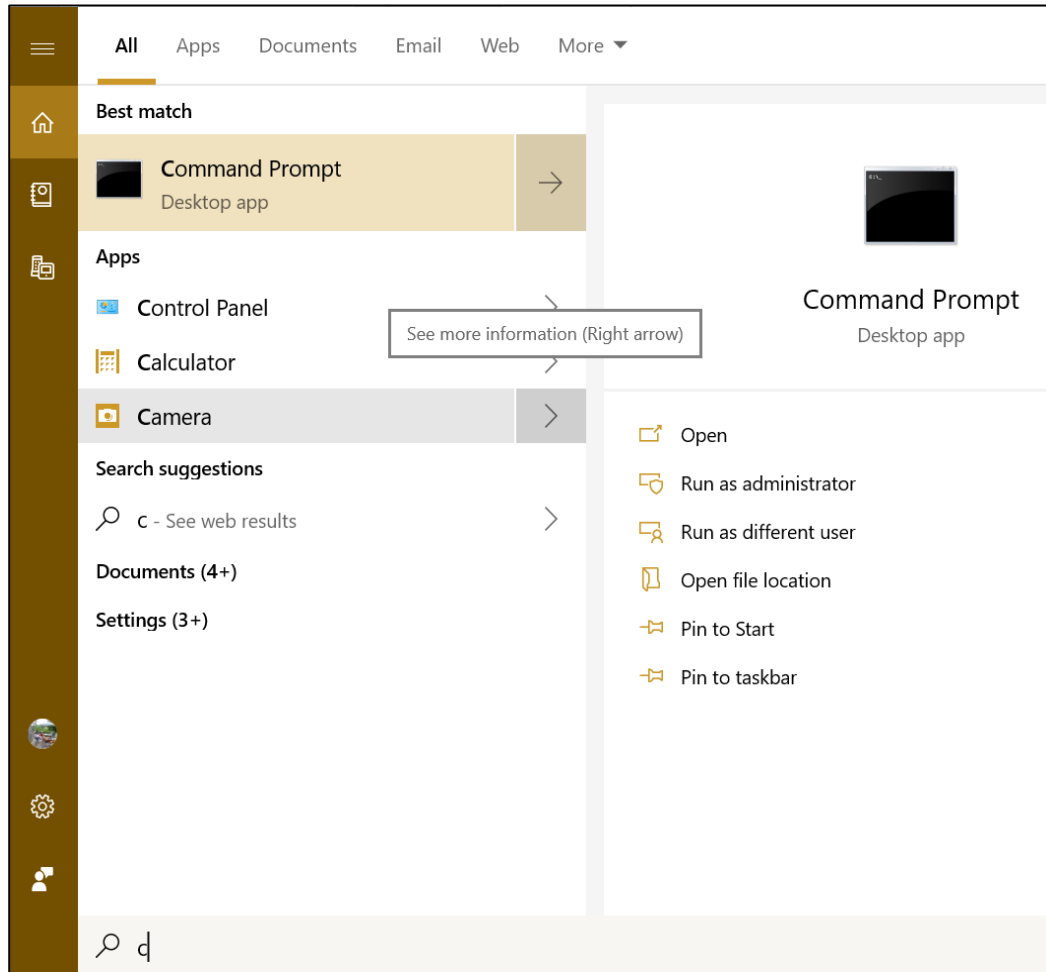
dotnet

- **dotnet** is a command line utility that has commands for creating, building, and running Net Core projects.
- You can do, in Visual Studio and (I think) Visual Studio for Mac, everything that dotnet does, but often dotnet is quicker and easier.
- **dotnet** is especially convenient when using Visual Studio Code.
 - We'll talk about VS Code in a follow-on presentation.

The Command Line

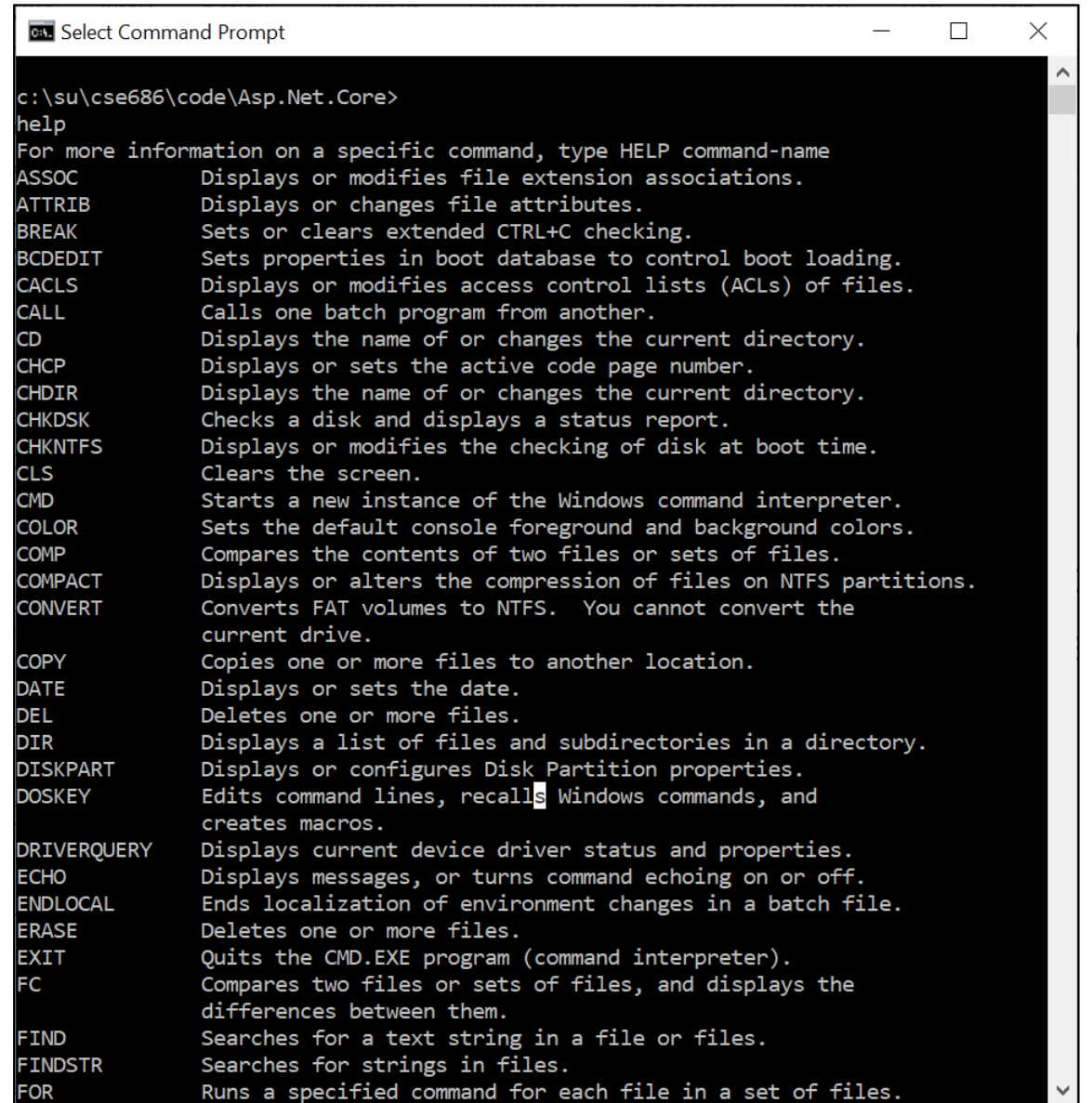
- The **dotnet** command runs from the command line. On Windows, That means either:
 - **Cmd.exe**
 - The default shell.
 - **Developer's cmd prompt**
 - Cmd with added environment variables to support code tool chains.
 - **PowerShell**
 - A powerful, but complex, shell.

Cmd.exe



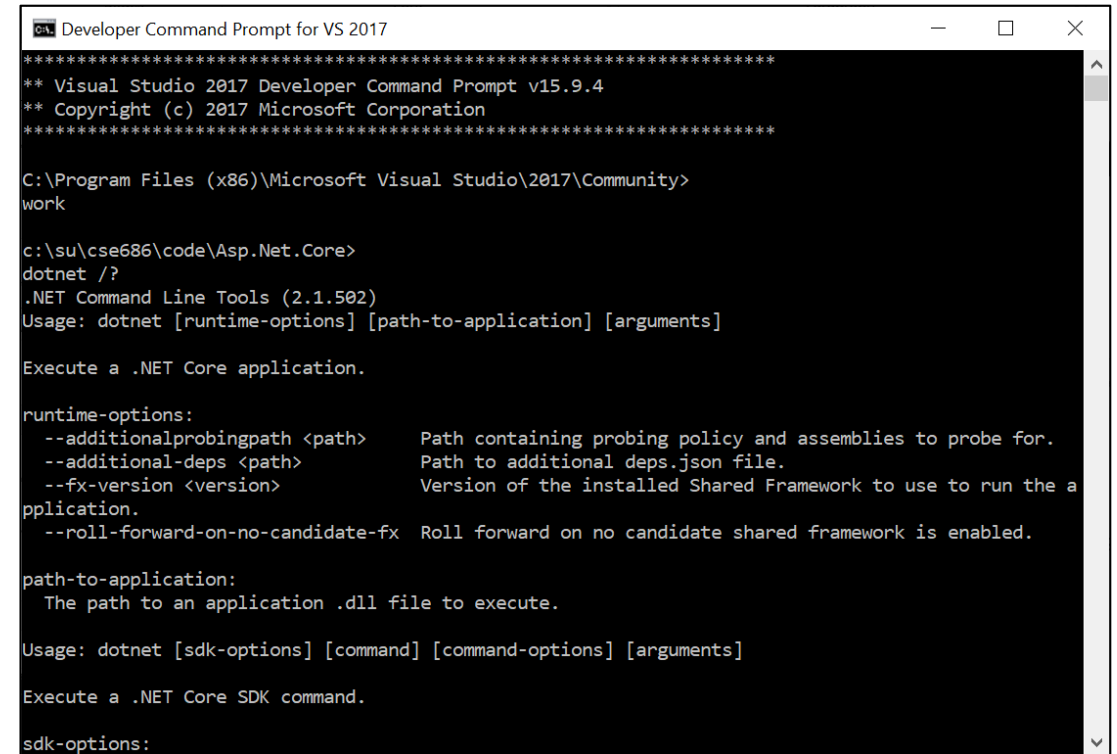
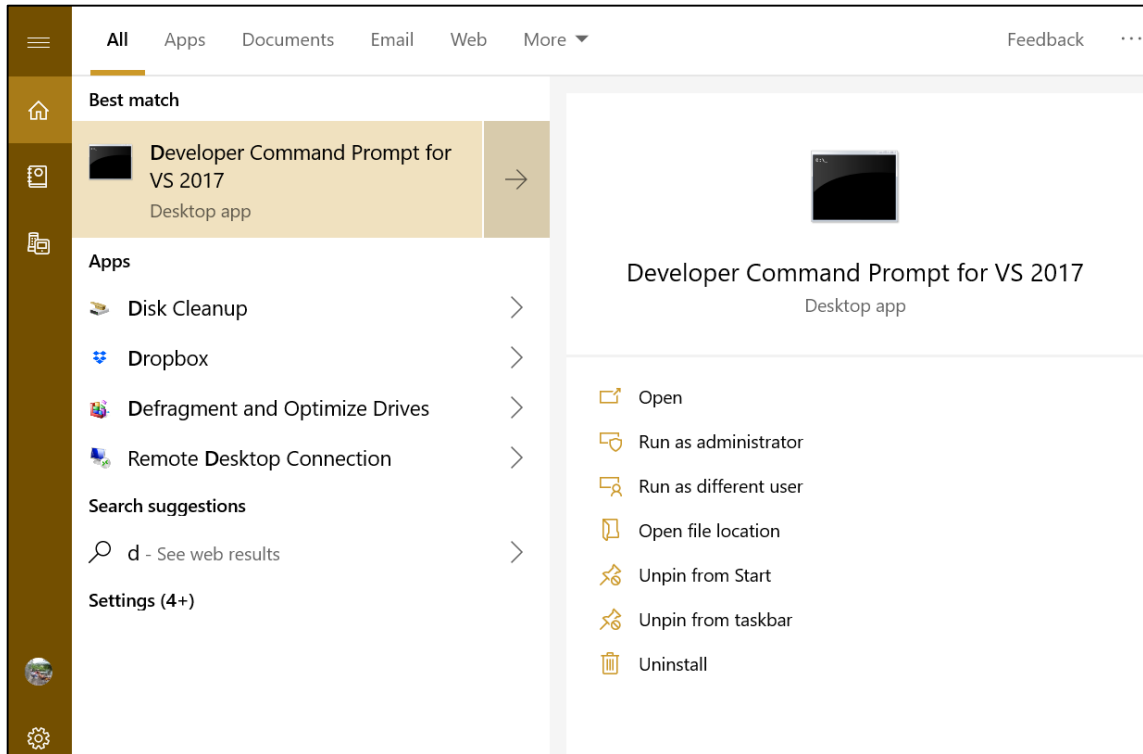
Functionality

- Cmd.exe
 - Create files and directories
 - Run shell commands
 - Run batch files and executables on your path
 - Run dotnet command



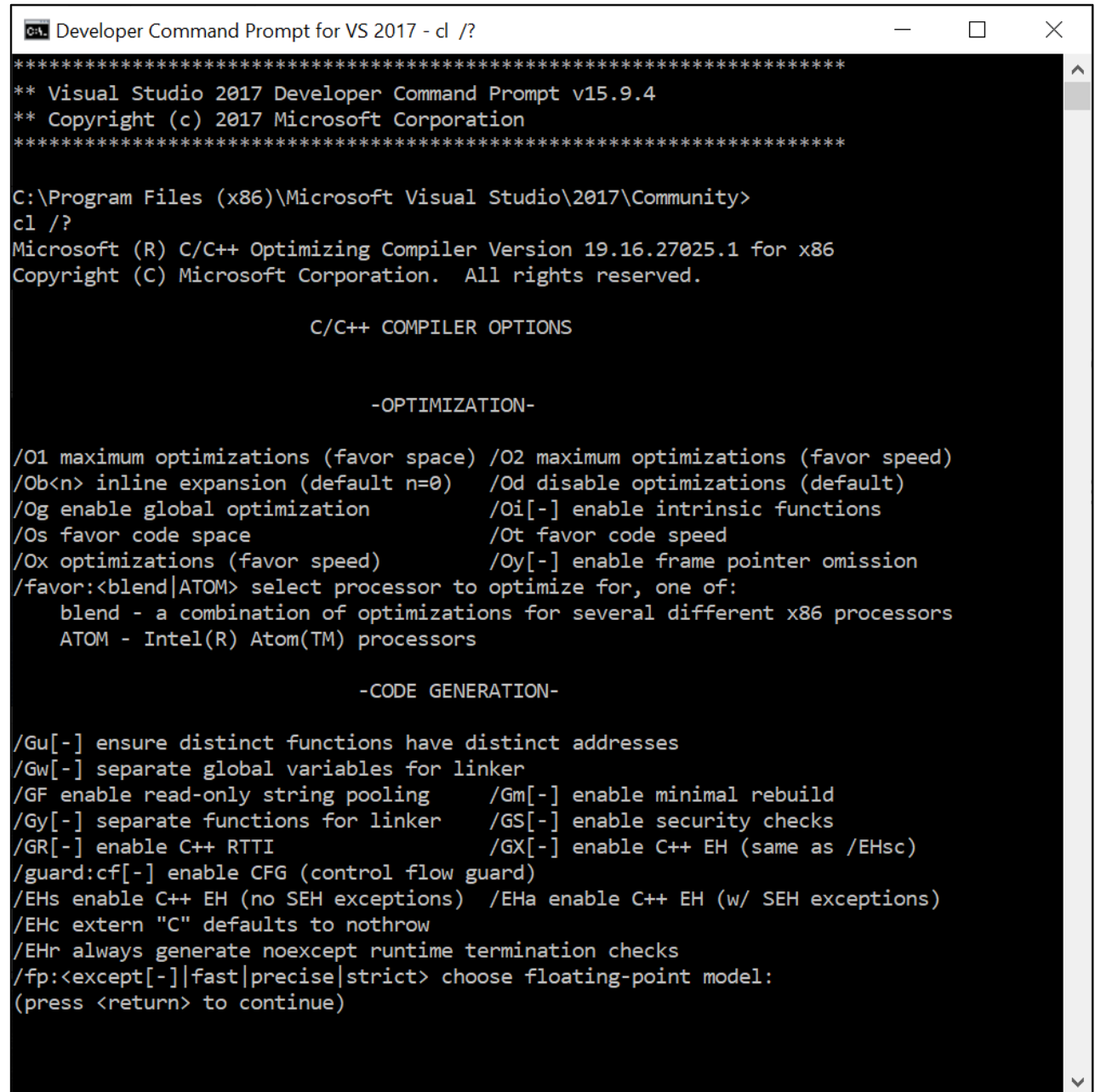
```
C:\su\cse686\code\Asp.Net.Core> help
For more information on a specific command, type HELP command-name
ASSOC      Displays or modifies file extension associations.
ATTRIB     Displays or changes file attributes.
BREAK      Sets or clears extended CTRL+C checking.
BCDEDIT    Sets properties in boot database to control boot loading.
CACLS      Displays or modifies access control lists (ACLs) of files.
CALL       Calls one batch program from another.
CD          Displays the name of or changes the current directory.
CHCP       Displays or sets the active code page number.
CHDIR      Displays the name of or changes the current directory.
CHKDSK     Checks a disk and displays a status report.
CHKNTFS    Displays or modifies the checking of disk at boot time.
CLS        Clears the screen.
CMD        Starts a new instance of the Windows command interpreter.
COLOR      Sets the default console foreground and background colors.
COMP       Compares the contents of two files or sets of files.
COMPACT    Displays or alters the compression of files on NTFS partitions.
CONVERT    Converts FAT volumes to NTFS. You cannot convert the
           current drive.
COPY       Copies one or more files to another location.
DATE       Displays or sets the date.
DEL        Deletes one or more files.
DIR        Displays a list of files and subdirectories in a directory.
DISKPART   Displays or configures Disk Partition properties.
DOSKEY     Edits command lines, recalls Windows commands, and
           creates macros.
DRIVERQUERY Displays current device driver status and properties.
ECHO       Displays messages, or turns command echoing on or off.
ENDLOCAL   Ends localization of environment changes in a batch file.
ERASE      Deletes one or more files.
EXIT       Quits the CMD.EXE program (command interpreter).
FC         Compares two files or sets of files, and displays the
           differences between them.
FIND       Searches for a text string in a file or files.
FINDSTR    Searches for strings in files.
FOR        Runs a specified command for each file in a set of files.
```

Developer's Command Prompt



Functionality

- Developer Command
 - A superset of the Cmd.exe functionality
 - Specifically, has environment variables to access C++ tool chain.
 - We won't need that capability in this course, but do in CSE687-OOD.



```
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community>
cl /?
Microsoft (R) C/C++ Optimizing Compiler Version 19.16.27025.1 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

C/C++ COMPILER OPTIONS

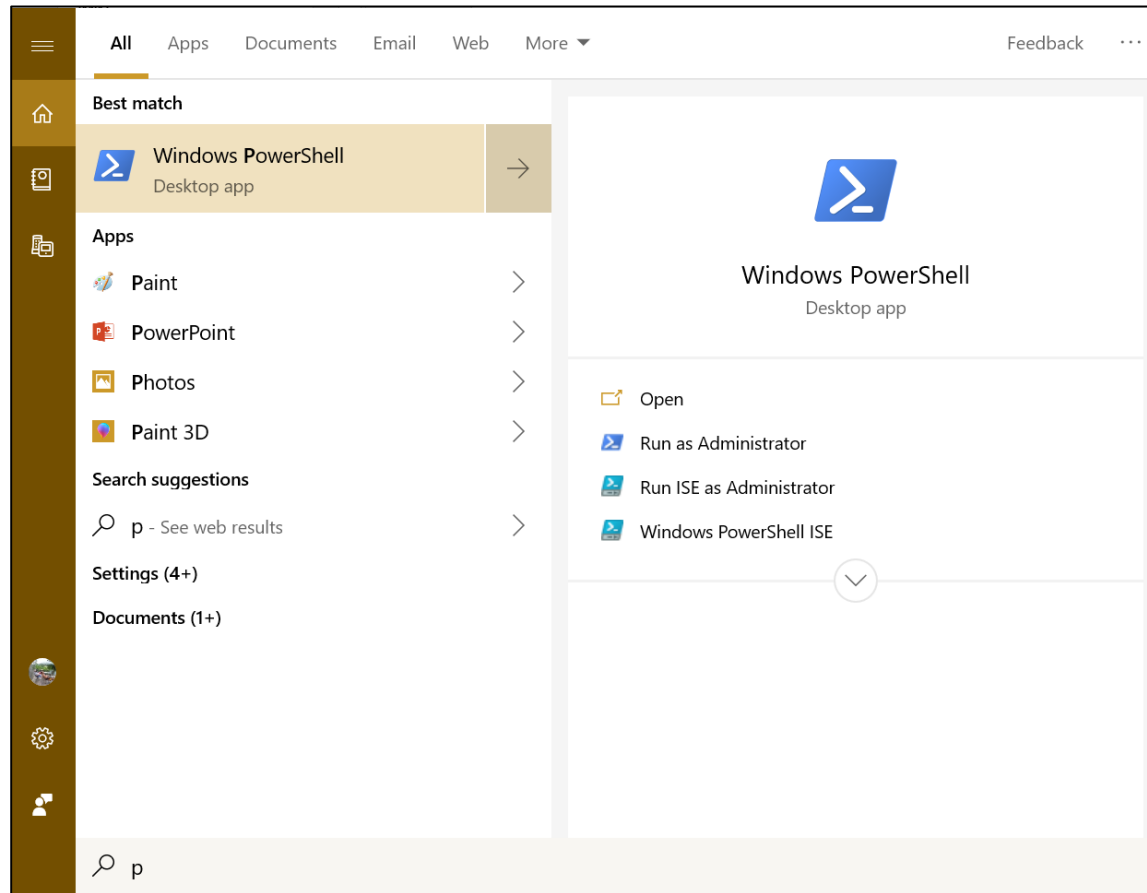
-OPTIMIZATION-

/O1 maximum optimizations (favor space) /O2 maximum optimizations (favor speed)
/Ob<n> inline expansion (default n=0) /Od disable optimizations (default)
/Og enable global optimization /Oi[-] enable intrinsic functions
/Os favor code space /Ot favor code speed
/Ox optimizations (favor speed) /Oy[-] enable frame pointer omission
/favor:<blend|ATOM> select processor to optimize for, one of:
    blend - a combination of optimizations for several different x86 processors
    ATOM - Intel(R) Atom(TM) processors

-CODE GENERATION-

/Gu[-] ensure distinct functions have distinct addresses
/Gw[-] separate global variables for linker
/GF enable read-only string pooling /Gm[-] enable minimal rebuild
/Gy[-] separate functions for linker /GS[-] enable security checks
/GR[-] enable C++ RTTI /GX[-] enable C++ EH (same as /EHsc)
/guard:cf[-] enable CFG (control flow guard)
/EHs enable C++ EH (no SEH exceptions) /EHa enable C++ EH (w/ SEH exceptions)
/EHc extern "C" defaults to nothrow
/EHr always generate noexcept runtime termination checks
/fp:<except[-]|fast|precise|strict> choose floating-point model:
(press <return> to continue)
```

PowerShell



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

PS: C:\su\CSE686\code\Asp.Net.Core >
work
PS: C:\su\CSE686\code\Asp.Net.Core >
dotnet new /?
Usage: new [options]

Options:
-h, --help            Displays help for this command.
-l, --list            Lists templates containing the specified name. If no name is specified, lists all templates.
-n, --name            The name for the output being created. If no name is specified, the name of the current directory
is used.
-o, --output          Location to place the generated output.
-i, --install         Installs a source or a template pack.
-u, --uninstall      Uninstalls a source or a template pack.
--nuget-source       Specifies a NuGet source to use during install.
--type              Filters templates based on available types. Predefined values are "project", "item" or "other".
--force             Forces content to be generated even if it would change existing files.
--lang, --language  Filters templates based on language and specifies the language of the template to create.

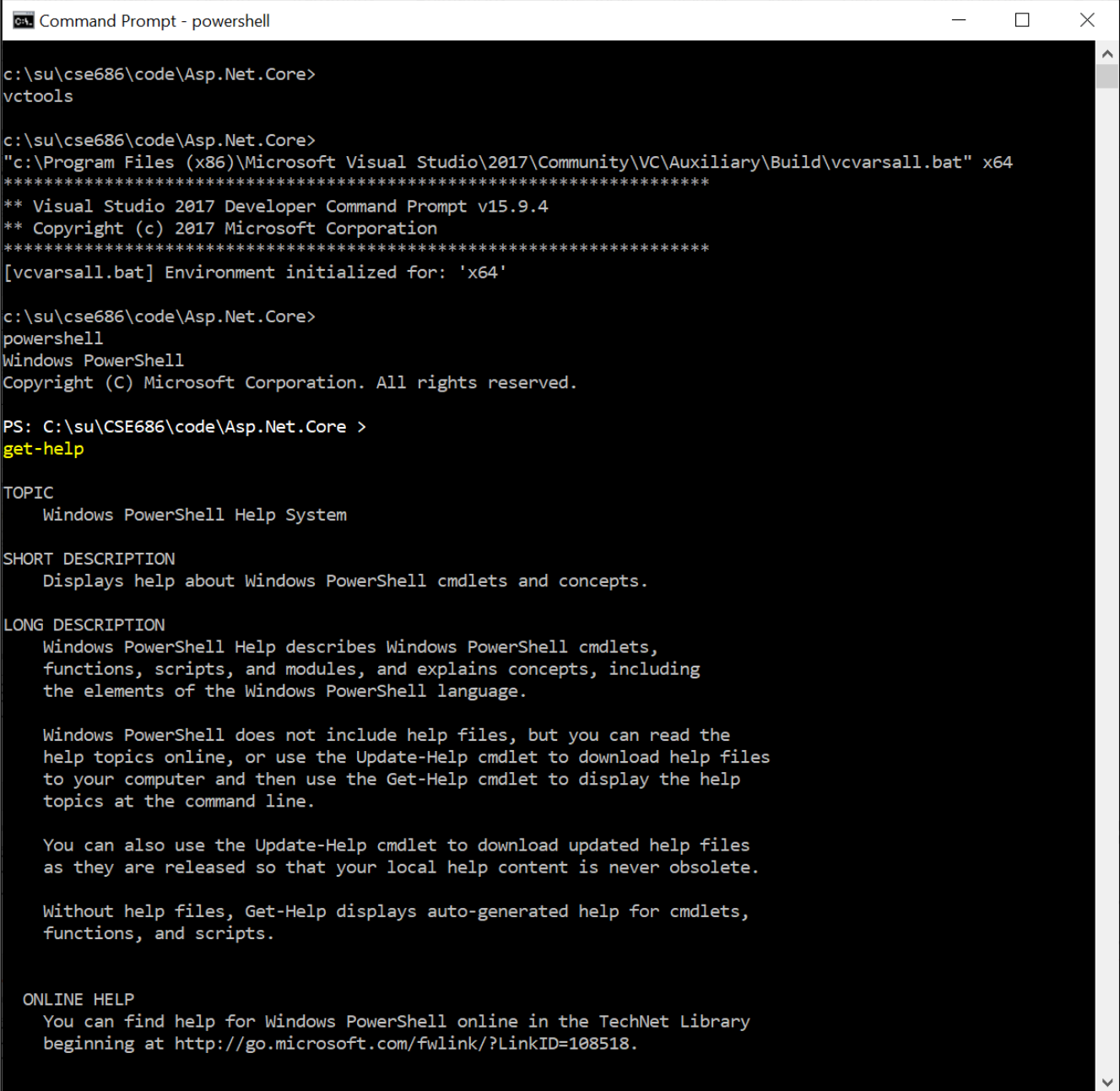
No templates matched the input template name: /?.

Templates
```

Templates	Short Name	Language	Tags
Console Application	console	[C#], F#, VB	Common/Console
Class library	classlib	[C#], F#, VB	Common/Library
Unit Test Project	mstest	[C#], F#, VB	Test/MSTest
NUnit 3 Test Project	nunit	[C#], F#, VB	Test/NUnit
NUnit 3 Test Item	nunit-test	[C#], F#, VB	Test/NUnit
xUnit Test Project	xunit	[C#], F#, VB	Test/xUnit
Razor Page	page	[C#]	Web/ASP.NET
MVC ViewImports	viewimports	[C#]	Web/ASP.NET
MVC ViewStart	viewstart	[C#]	Web/ASP.NET
ASP.NET Core Empty	web	[C#], F#	Web/Empty
ASP.NET Core Web App (Model-View-Controller)	mvc	[C#], F#	Web/MVC

Functionality

- PowerShell.exe
 - A superset of the Cmd.exe functionality
 - Has a lot of operations for system administration.
 - Does not have environment variables to access C++ tool chain.
 - We can add those by opening a cmd prompt, executing vcvarsall.bat, and then opening powershell from within the cmd shell.



```
Command Prompt - powershell
C:\su\cse686\code\Asp.Net.Core>
vctools

C:\su\cse686\code\Asp.Net.Core>
"c:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Auxiliary\Build\vcvarsall.bat" x64
*****
** Visual Studio 2017 Developer Command Prompt v15.9.4
** Copyright (c) 2017 Microsoft Corporation
*****
[vcvarsall.bat] Environment initialized for: 'x64'

C:\su\cse686\code\Asp.Net.Core>
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS: C:\su\CSE686\code\Asp.Net.Core >
get-help

TOPIC
    Windows PowerShell Help System

SHORT DESCRIPTION
    Displays help about Windows PowerShell cmdlets and concepts.

LONG DESCRIPTION
    Windows PowerShell Help describes Windows PowerShell cmdlets,
    functions, scripts, and modules, and explains concepts, including
    the elements of the Windows PowerShell language.

    Windows PowerShell does not include help files, but you can read the
    help topics online, or use the Update-Help cmdlet to download help files
    to your computer and then use the Get-Help cmdlet to display the help
    topics at the command line.

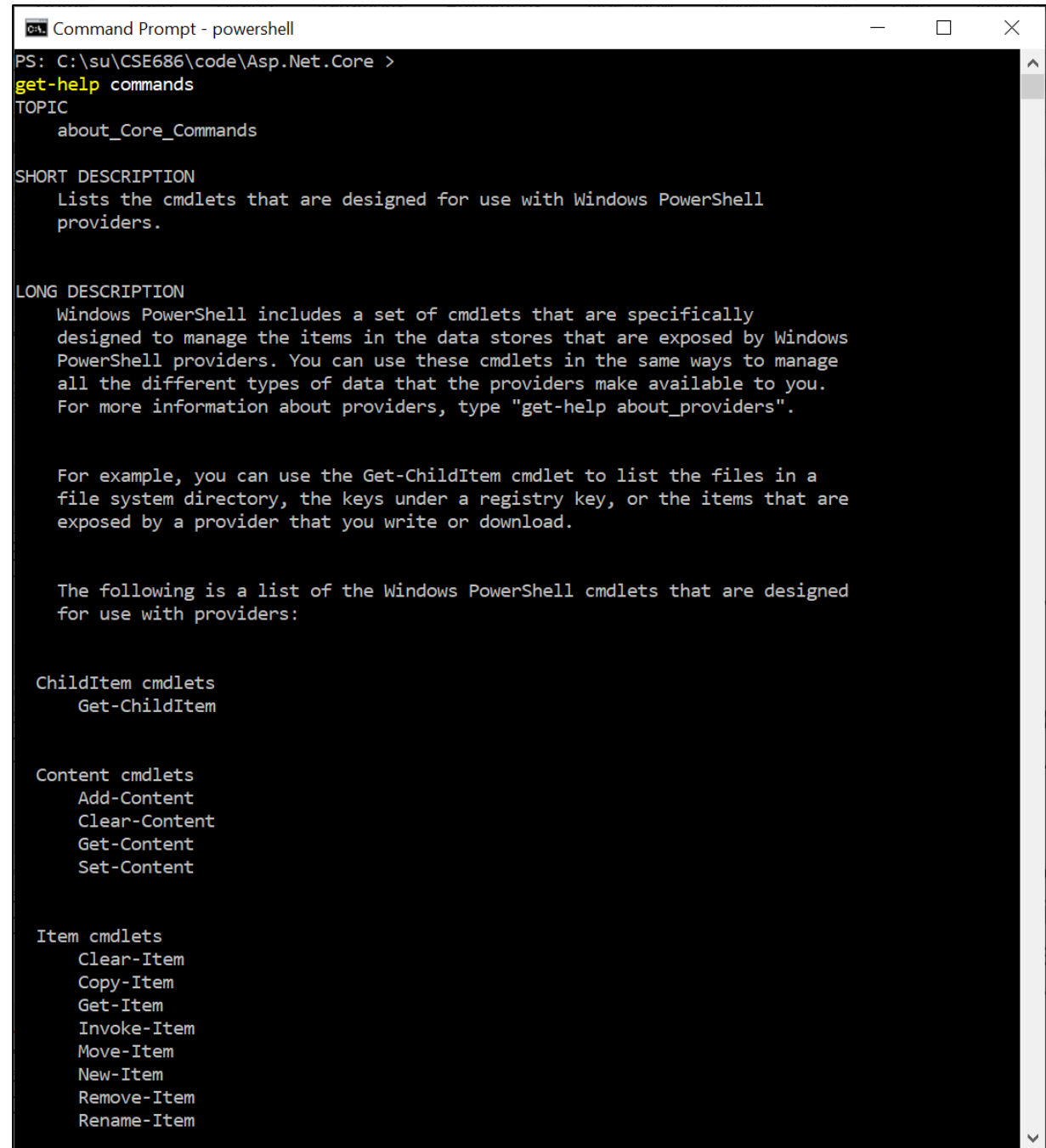
    You can also use the Update-Help cmdlet to download updated help files
    as they are released so that your local help content is never obsolete.

    Without help files, Get-Help displays auto-generated help for cmdlets,
    functions, and scripts.

ONLINE HELP
    You can find help for Windows PowerShell online in the TechNet Library
    beginning at http://go.microsoft.com/fwlink/?LinkID=108518.
```

PowerShell Operations

- PowerShell.exe
 - Syntax for commands is:
 - Verb Noun
 - Example: get-Help commands
 - Useful exploratory commands:
 - Get-help
 - Get-Help commands
 - Get-Help objects
 - Get-help About_Methods
 - Get-help About_Properties
 - Get-help About



```
Command Prompt - powershell
PS: C:\su\CSE686\code\Asp.Net.Core >
get-help commands
TOPIC
    about_Core_Commands

SHORT DESCRIPTION
    Lists the cmdlets that are designed for use with Windows PowerShell
    providers.

LONG DESCRIPTION
    Windows PowerShell includes a set of cmdlets that are specifically
    designed to manage the items in the data stores that are exposed by Windows
    PowerShell providers. You can use these cmdlets in the same ways to manage
    all the different types of data that the providers make available to you.
    For more information about providers, type "get-help about_providers".

    For example, you can use the Get-ChildItem cmdlet to list the files in a
    file system directory, the keys under a registry key, or the items that are
    exposed by a provider that you write or download.

    The following is a list of the Windows PowerShell cmdlets that are designed
    for use with providers:

ChildItem cmdlets
    Get-ChildItem

Content cmdlets
    Add-Content
    Clear-Content
    Get-Content
    Set-Content

Item cmdlets
    Clear-Item
    Copy-Item
    Get-Item
    Invoke-Item
    Move-Item
    New-Item
    Remove-Item
    Rename-Item
```

Examples

```
Command Prompt - powershell
PS: C:\su\CSE686\code\Asp.Net.Core >
get-help about_methods
TOPIC
    about_methods
SHORT DESCRIPTION
    Describes how to use methods to perform actions on objects in Windows PowerShell.
LONG DESCRIPTION
    Windows PowerShell uses objects to represent the items in data stores or the state of the computer. For example, FileInfo objects represent the files in file system drives and ProcessInfo objects represent the processes on the computer.

    Objects have properties, which store data about the object, and methods that let you change the object.

    A "method" is a set of instructions that specify an action you can perform on the object. For example, the FileInfo object includes the CopyTo method that copies the file that the FileInfo object represents.

    To get the methods of any object, use the Get-Member cmdlet. Use its MemberType property with a value of "Method". The following command gets the methods of process objects.

    PS C:\>Get-Process | Get-Member -MemberType Method

    TypeName: System.Diagnostics.Process

    Name                MemberType Definition
    ----                -
    BeginErrorReadLine  Method      System.Void BeginErrorReadLine()
    BeginOutputReadLine Method      System.Void BeginOutputReadLine()
    ...
    Kill                Method      System.Void Kill()
    Refresh             Method      System.Void Refresh()
    Start              Method      bool Start()
    ToString           Method      string ToString()
    WaitForExit        Method      bool WaitForExit(int milliseconds), System.Void Wai
WaitForExit()
    WaitForInputIdle   Method      bool WaitForInputIdle(int milliseconds), bool Wai
WaitForInputIdle()

    To perform or "invoke" a method of an object, type a dot (.), the method
```

```
Command Prompt - powershell
PS: C:\su\CSE686\code\Asp.Net.Core >
get-help get-member -examples
NAME
    Get-Member
SYNOPSIS
    Gets the properties and methods of objects.

    Example 1: Get the members of process objects

    PS C:\>Get-Service | Get-Member
    TypeName: System.ServiceProcess.ServiceController
    Name                MemberType Definition
    ----                -
    Name                AliasProperty Name = ServiceName
    Close              Method      System.Void Close()
    Continue           Method      System.Void Continue()
    CreateObjRef       Method      System.Runtime.Remoting.ObjRef CreateObjRef(Type
requestedType)
    Dispose            Method      System.Void Dispose()
    Equals             Method      System.Boolean Equals(Object obj)
    ExecuteCommand     Method      System.Void ExecuteCommand(Int32 command)
    GetHashCode        Method      System.Int32 GetHashCode()
    GetLifetimeService Method      System.Object GetLifetimeService()
    GetType            Method      System.Type GetType()
    InitializeLifetimeService Method      System.Object InitializeLifetimeService()
    Pause             Method      System.Void Pause()
    Refresh            Method      System.Void Refresh()
    Start              Method      System.Void Start(), System.Void Start(String[]
args)
    Stop              Method      System.Void Stop()
    ToString           Method      System.String ToString()
    WaitForStatus     Method      System.Void WaitForStatus(ServiceControllerStatus
desiredStatus), System.Voi...
    CanPauseAndContinue Property    System.Boolean CanPauseAndContinue {get;}
    CanShutdown       Property    System.Boolean CanShutdown {get;}
    CanStop           Property    System.Boolean CanStop {get;}
    Container         Property    System.ComponentModel.IContainer Container {get;}
    DependentServices Property    System.ServiceProcess.ServiceController[]
DependentServices {get;}
    DisplayName       Property    System.String DisplayName {get;set;}
    MachineName       Property    System.String MachineName {get;set;}
    ServiceHandle     Property    System.Runtime.InteropServices.SafeHandle
ServiceHandle {get;}
    ServiceName       Property    System.String ServiceName {get;set;}
```

Now: dotnet command

```
Command Prompt - powershell
PS: C:\su\CSE686\code\Asp.Net.Core >
dotnet --info
.NET Core SDK (reflecting any global.json):
  Version: 2.1.502
  Commit: c74ce8f29f

Runtime Environment:
  OS Name: Windows
  OS Version: 10.0.17763
  OS Platform: Windows
  RID: win10-x64
  Base Path: C:\Program Files\dotnet\sdk\2.1.502\

Host (useful for support):
  Version: 2.1.6
  Commit: 3f4f8eebd8

.NET Core SDKs installed:
  1.0.4 [C:\Program Files\dotnet\sdk]
  1.1.0 [C:\Program Files\dotnet\sdk]
  2.0.2 [C:\Program Files\dotnet\sdk]
  2.1.2 [C:\Program Files\dotnet\sdk]
  2.1.202 [C:\Program Files\dotnet\sdk]
  2.1.402 [C:\Program Files\dotnet\sdk]
  2.1.500 [C:\Program Files\dotnet\sdk]
  2.1.502 [C:\Program Files\dotnet\sdk]

.NET Core runtimes installed:
  Microsoft.AspNetCore.All 2.1.4 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.All]
  Microsoft.AspNetCore.All 2.1.6 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.All]
  Microsoft.AspNetCore.App 2.1.4 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 2.1.6 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 1.0.5 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 1.1.2 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 2.0.0 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 2.0.3 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 2.0.9 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 2.1.4 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 2.1.6 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]

To install additional .NET Core runtimes or SDKs:
  https://aka.ms/dotnet-download
PS: C:\su\CSE686\code\Asp.Net.Core >
```

```
Command Prompt - powershell
PS: C:\su\CSE686\code\Asp.Net.Core >
dotnet /?
.NET Command Line Tools (2.1.502)
Usage: dotnet [runtime-options] [path-to-application] [arguments]

Execute a .NET Core application.

runtime-options:
  --additionalprobingpath <path>      Path containing probing policy and assemblies
to probe for.
  --additional-deps <path>            Path to additional deps.json file.
  --fx-version <version>              Version of the installed Shared Framework to u
se to run the application.
  --roll-forward-on-no-candidate-fx  Roll forward on no candidate shared framework
is enabled.

path-to-application:
  The path to an application .dll file to execute.

Usage: dotnet [sdk-options] [command] [command-options] [arguments]

Execute a .NET Core SDK command.

sdk-options:
  -d|--diagnostics  Enable diagnostic output.
  -h|--help         Show command line help.
  --info           Display .NET Core information.
  --list-runtimes  Display the installed runtimes.
  --list-sdks     Display the installed SDKs.
  --version       Display .NET Core SDK version in use.

SDK commands:
  add      Add a package or reference to a .NET project.
  build   Build a .NET project.
  build-server  Interact with servers started by a build.
  clean   Clean build outputs of a .NET project.
  help   Show command line help.
  list   List project references of a .NET project.
  migrate Migrate a project.json project to an MSBuild project.
  msbuild Run Microsoft Build Engine (MSBuild) commands.
  new    Create a new .NET project or file.
  nuget  Provides additional NuGet commands.
```

dotnet command

```
Command Prompt - powershell

SDK commands:
  add          Add a package or reference to a .NET project.
  build        Build a .NET project.
  build-server Interact with servers started by a build.
  clean        Clean build outputs of a .NET project.
  help         Show command line help.
  list         List project references of a .NET project.
  migrate      Migrate a project.json project to an MSBuild project.
  msbuild      Run Microsoft Build Engine (MSBuild) commands.
  new          Create a new .NET project or file.
  nuget        Provides additional NuGet commands.
  pack         Create a NuGet package.
  publish      Publish a .NET project for deployment.
  remove       Remove a package or reference from a .NET project.
  restore      Restore dependencies specified in a .NET project.
  run          Build and run a .NET project output.
  sln          Modify Visual Studio solution files.
  store        Store the specified assemblies in the runtime package store.

  test        Run unit tests using the test runner specified in a .NET project.
  tool         Install or manage tools that extend the .NET experience.
  vstest       Run Microsoft Test Engine (VSTest) commands.

Additional commands from bundled tools:
  dev-certs   Create and manage development certificates.
  ef           Entity Framework Core command-line tools.
  sql-cache   SQL Server cache command-line tools.
  user-secrets Manage development user secrets.
  watch       Start a file watcher that runs a command when files change.

Run 'dotnet [command] --help' for more information on a command.
PS: C:\su\CSE686\code\Asp.Net.Core >
```

```
Command Prompt - powershell

PS: C:\su\CSE686\code\Asp.Net.Core >
dotnet new --help
Usage: new [options]

Options:
  -h, --help          Displays help for this command.
  -l, --list          Lists templates containing the specified name. If no name is specified, lists all templates.
  -n, --name          The name for the output being created. If no name is specified, the name of the current directory is used.
  -o, --output        Location to place the generated output.
  -i, --install       Installs a source or a template pack.
  -u, --uninstall    Uninstalls a source or a template pack.
  --nuget-source      Specifies a NuGet source to use during install.
  --type              Filters templates based on available types. Predefined values are "project", "item" or "other".
  --force             Forces content to be generated even if it would change existing files.
  -lang, --language  Filters templates based on language and specifies the language of the template to create.

Templates

-----
Short Name      Language      Tags
-----
Console Application      console      [C#], F#, VB      Common/Console
Class library            classlib     [C#], F#, VB      Common/Library
Unit Test Project        mstest      [C#], F#, VB      Test/MSTest
JUnit 3 Test Project     junit       [C#], F#, VB      Test/JUnit
JUnit 3 Test Item        junit-test  [C#], F#, VB      Test/JUnit
xUnit Test Project       xunit       [C#], F#, VB      Test/xUnit
Razor Page               page        [C#]               Web/ASP.NET
MVC ViewImports          viewimports [C#]               Web/ASP.NET
MVC ViewStart            viewstart   [C#]               Web/ASP.NET
ASP.NET Core Empty       web         [C#], F#           Web/Empty
ASP.NET Core Web App (Model-View-Controller) mvc         [C#], F#           Web/MVC
ASP.NET Core Web App     razor       [C#]               Web/MVC/Razor Pages
ASP.NET Core with Angular angular      [C#]               Web/MVC/SPA
ASP.NET Core with React.js react        [C#]               Web/MVC/SPA
ASP.NET Core with React.js and Redux reactredux  [C#]               Web/MVC/SPA
```

So What?

- If we use Visual Studio Code:
 - VS Code can't create .net core projects
 - So we create a project folder, call dotnet new console (or whatever)
 - Open Visual Studio Code
 - Open the newly created folder
 - When you do this VS Code will ask if you want to create resources to work with that folder.
 - Select yes – that builds a json settings file to manage builds and debugging
 - Start editing files and running

That's all Folks!