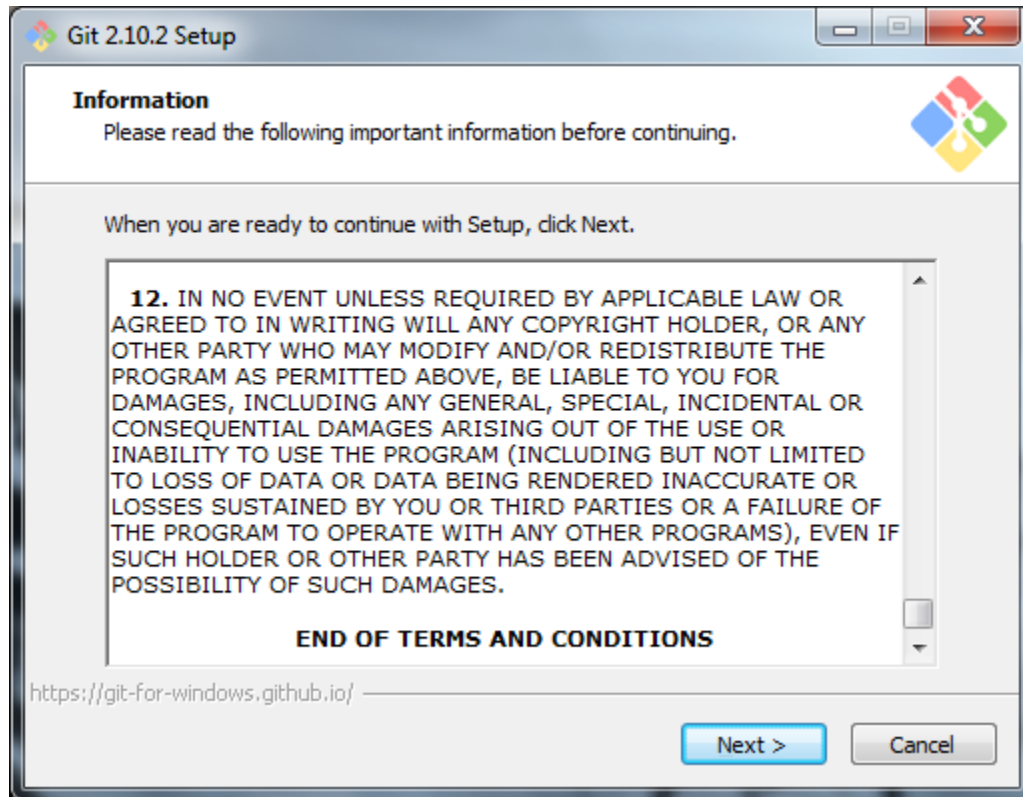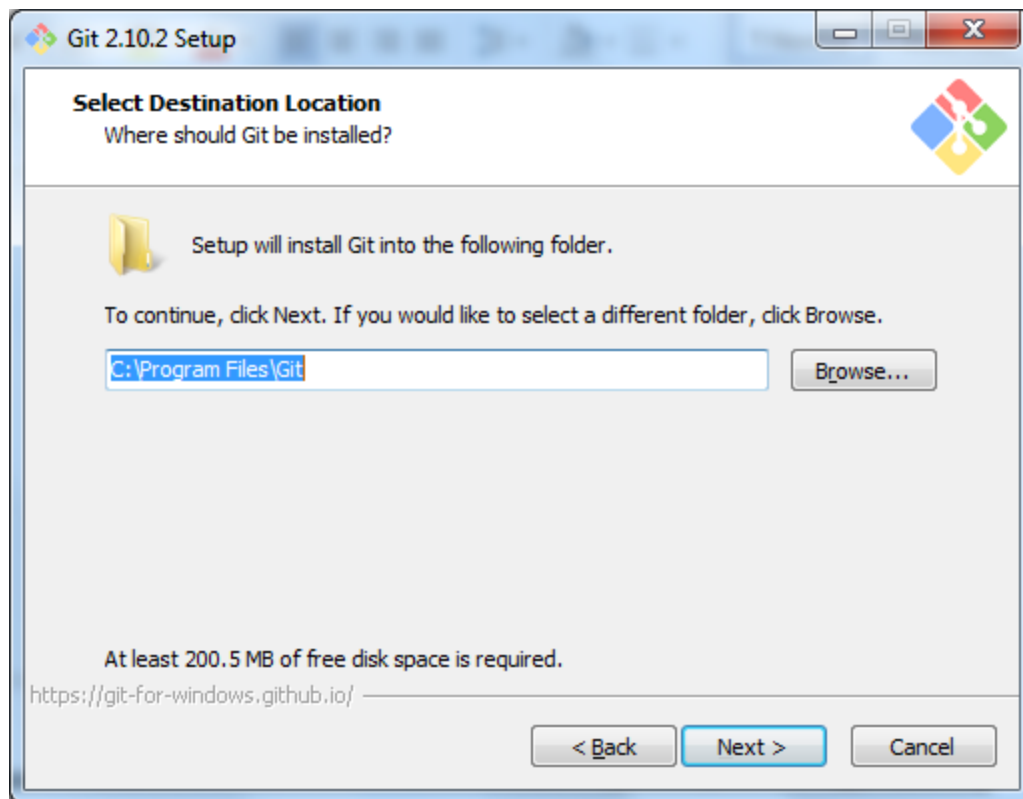This installation selects options that, post-deployment, will require running Git from the Git Bash window (for the command line interface). It was done to keep Git separate from the rest of the Windows/DOS environment and was recommended by several of the training videos. Note that it does require the user to become familiar with LINUX based commands to effectively use bash since you are not running from a command line. Some commands entered to configure Git after installation are also shown in this write-up.

This install was done on a Windows 7 Notebook with 16 Gigs of Ram.
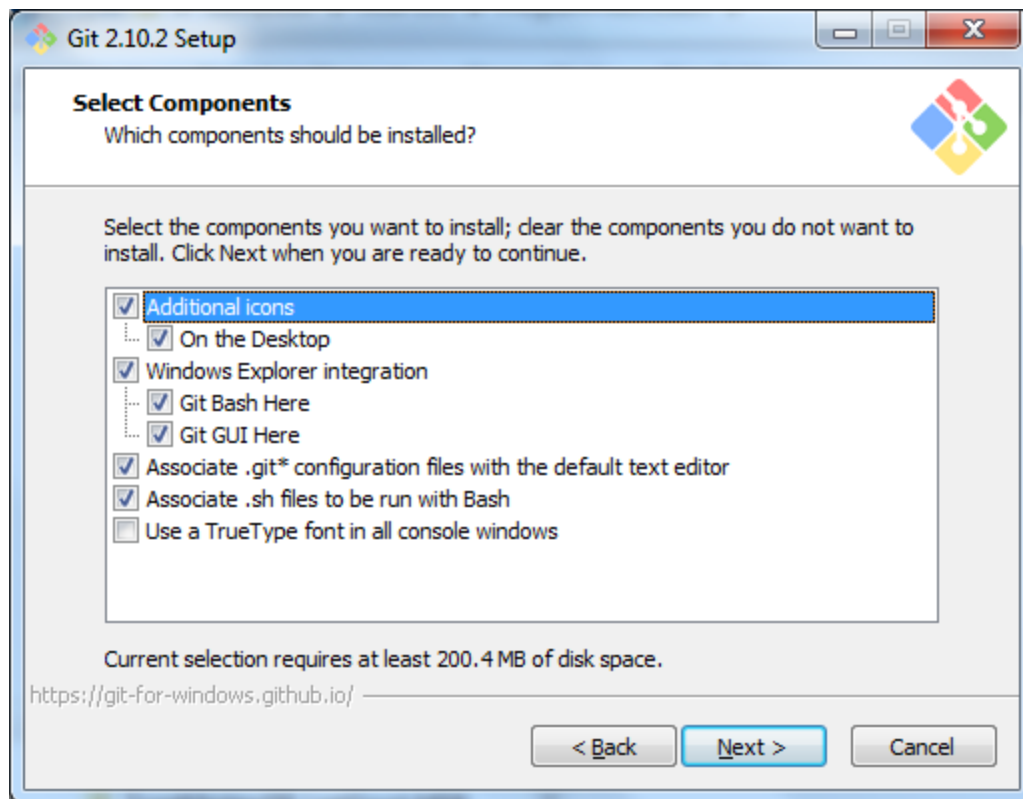
Git Installation setup screens:

Changed installation directory to be consistent with my desired setup – default and what I changed it to follows:

Git 2.10.2 Setup

Select Destination Location
Where should Git be installed?

Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git   Browse...

At least 200.5 MB of free disk space is required.

https://git-for-windows.github.io/

< Back   Next >   Cancel

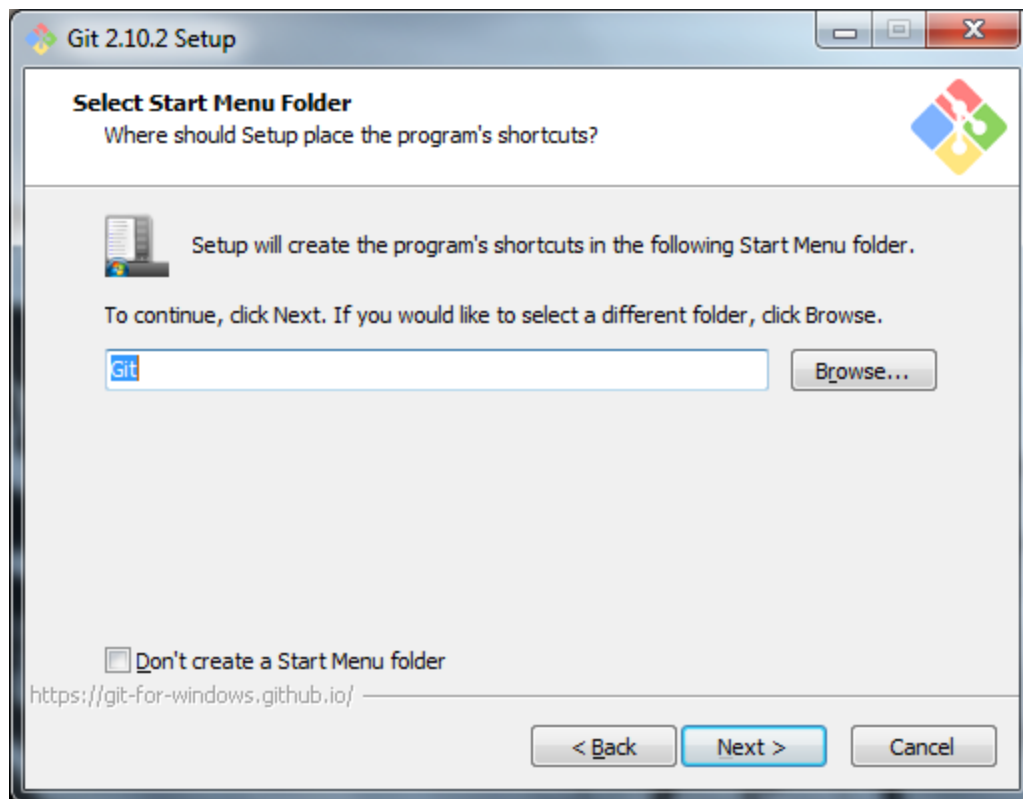Changed above to this (forgot to get screen capture):

C:\ProgramFilesCoders\Git

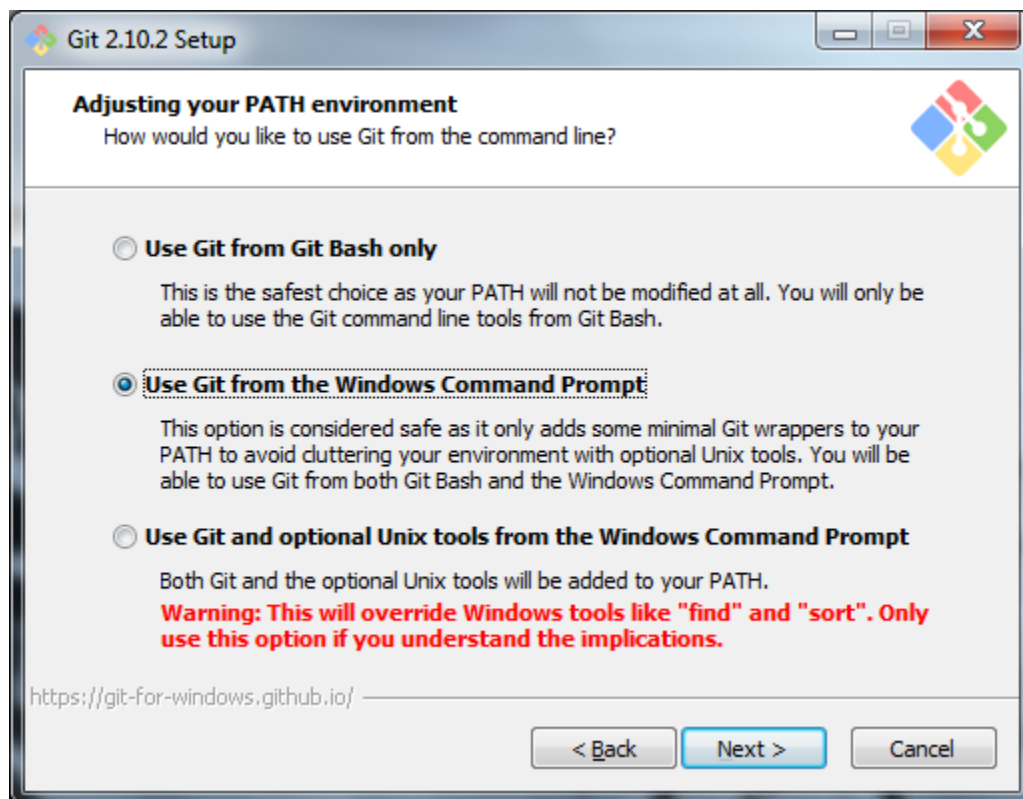Did this by manually changing the value, not browsing.

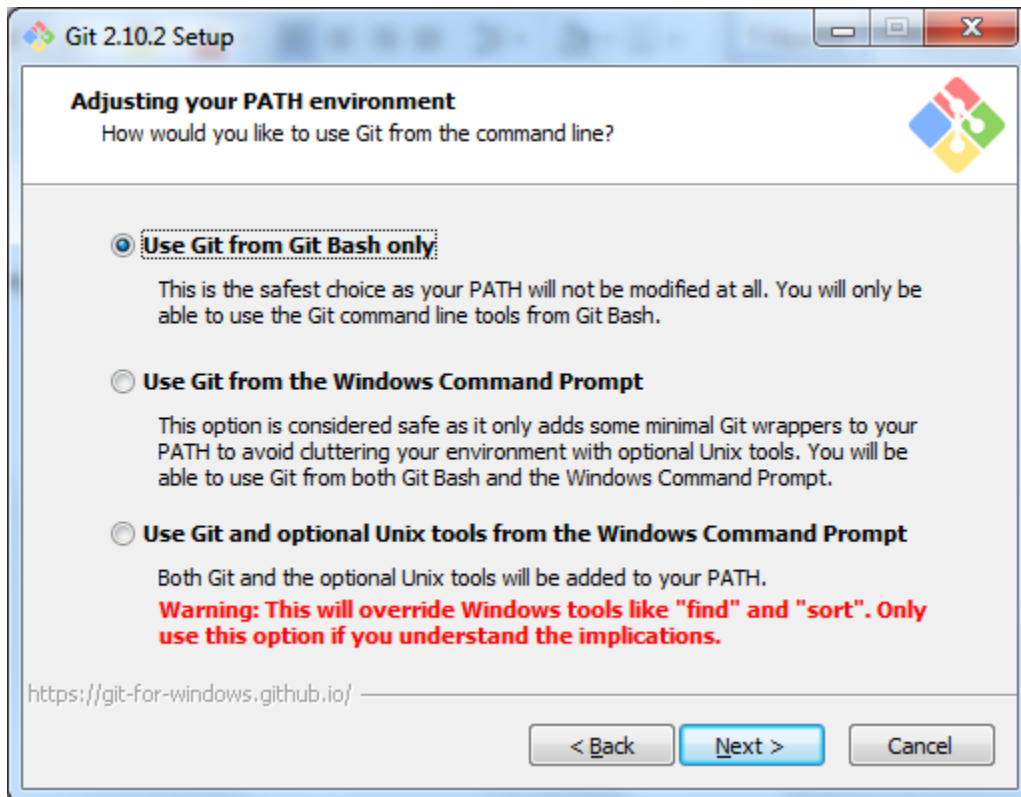Concern: this may hook widnows notepad in as default text editor

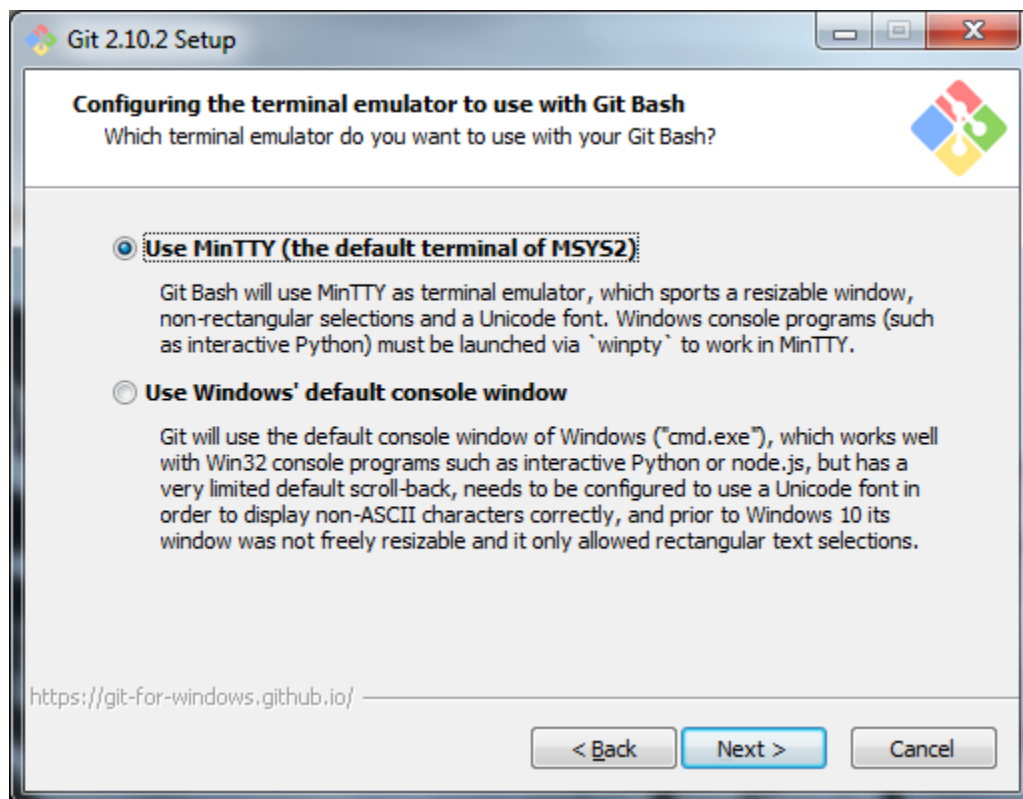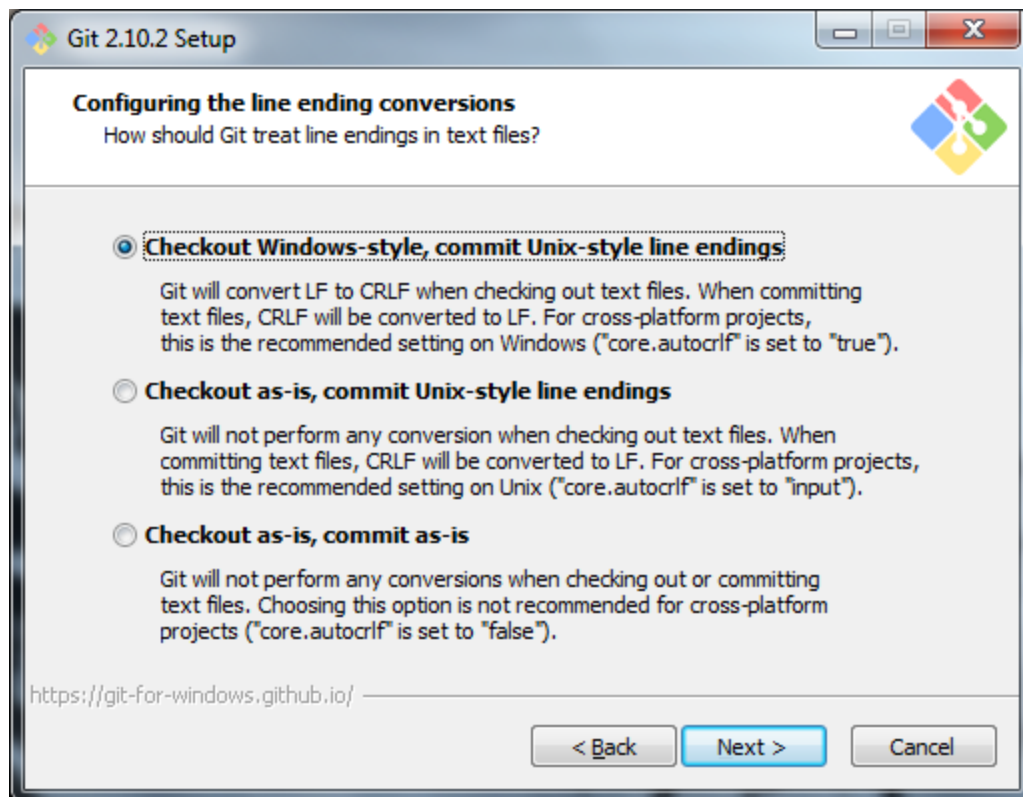Changing this to notepad++ may be indicated after installation

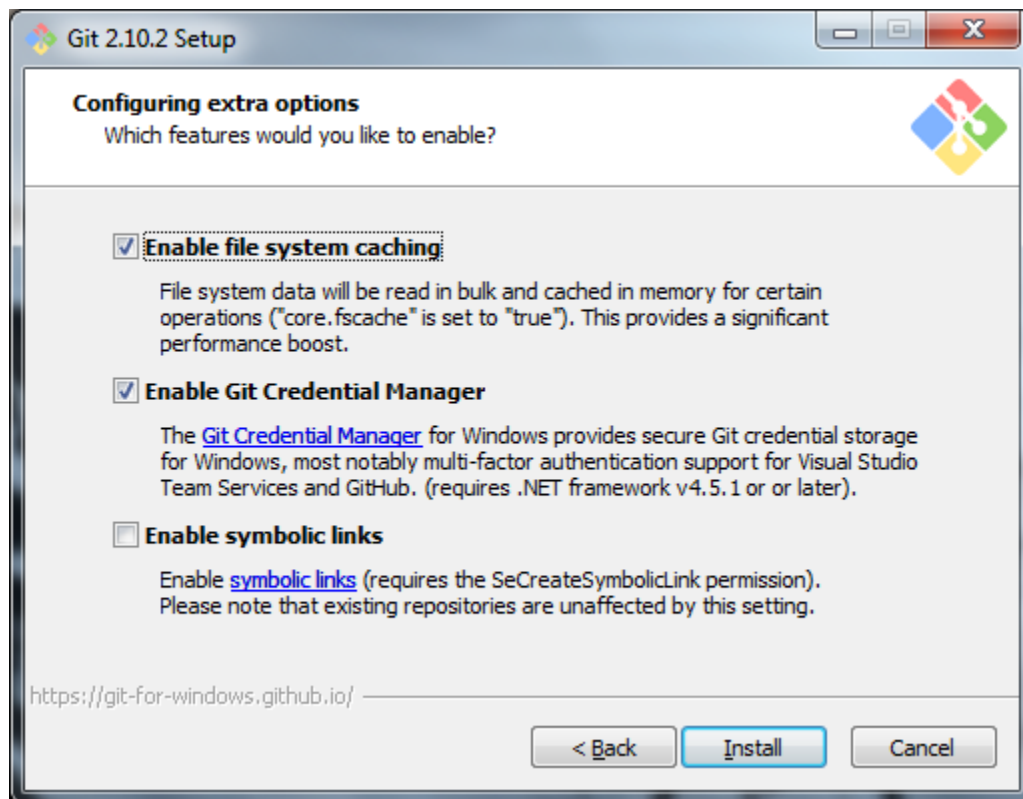Default (this is what it was before I changed the settings during installation):

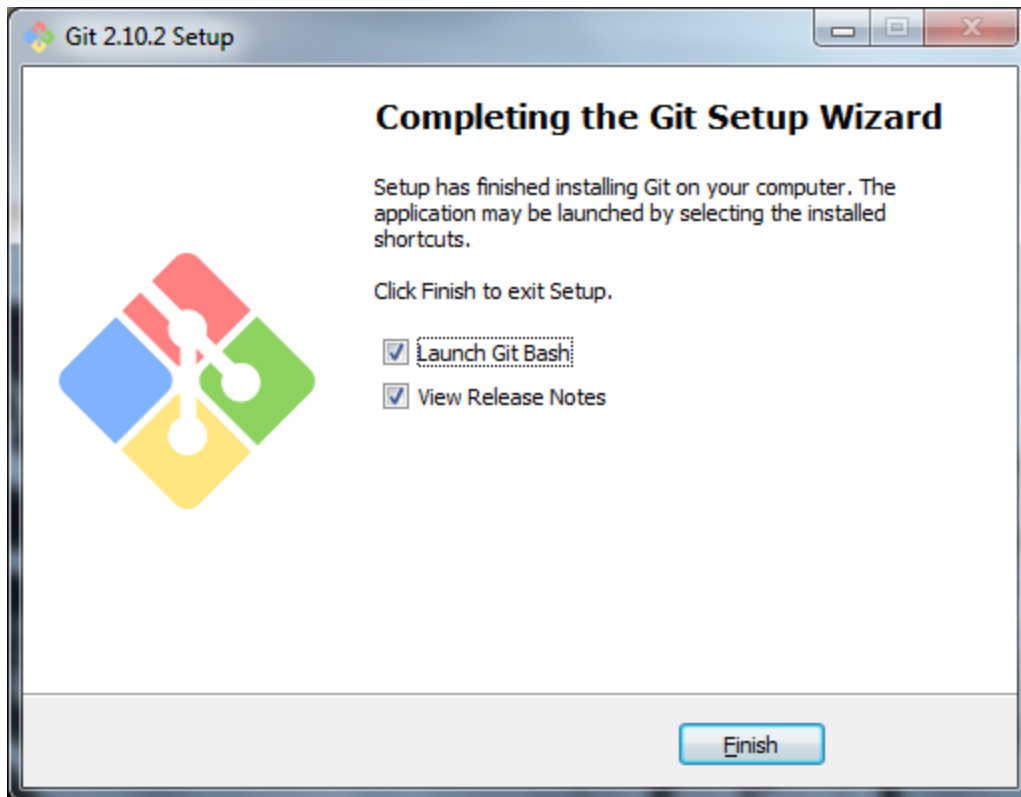Changed above from default as per training videos for beginners:

This setting may result in alteration to text files but keeping it anyway:

Git 2.10.2 Setup

Configuring the line ending conversions
How should Git treat line endings in text files?

● Checkout Windows-style, commit Unix-style line endings

Git will convert LF to CRLF when checking out text files. When committing
text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When
committing text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing
text files. Choosing this option is not recommended for cross-platform
projects ("core.autocrlf" is set to "false").

https://git-for-windows.github.io/

[< Back] [Next >] [Cancel]

Git 2.10.2 Setup

Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?

● Use MinTTY (the default terminal of MSYS2)

Git Bash will use MinTTY as terminal emulator, which sports a resizable window,
non-rectangular selections and a Unicode font. Windows console programs (such
as interactive Python) must be launched via `winpty` to work in MinTTY.

○ Use Windows' default console window

Git will use the default console window of Windows ("cmd.exe"), which works well
with Win32 console programs such as interactive Python or node.js, but has a
very limited default scroll-back, needs to be configured to use a Unicode font in
order to display non-ASCII characters correctly, and prior to Windows 10 its
window was not freely resizable and it only allowed rectangular text selections.

https://git-for-windows.github.io/

[< Back] [Next >] [Cancel]

Keeping defaults:

Git Bash Windows: Commands after installation …

```
$ git config --global user.name "<username>"

$ git config --global user.email "<email>"

$ git --version
git version 2.10.2.windows.1

$ git config --global core.editor "'C:/ProgramFilesCoders/Notepad_pp/notepad++.exe' -
multiInst -nosession"

$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=C:/ProgramFilesCoders/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
credential.helper=manager
user.name=<username>
user.email=<email>
core.editor='C:/ProgramFilesCoders/Notepad_pp/notepad++.exe' -multiInst -nosession
```

Note: The path for Windows Notepad is unique to my system (I did not use the defaults during installation). You will need to use the correct path from your system to replicate this environment choice (or the text editor of your choice).