

- ✓ There is too much any human being can remember during the first week so making notes, reviewing, and retracing steps is recommended. So is giving yourself a break.
- ✓ Our school tries to be operating system agnostic. We believe in the existence of one true and [eternal Turing Machine](#). When it comes to implementation, faith and beliefs may vary but all are respected.
- ✓ There are times when a particular operating system is necessary and when it is, this course uses Microsoft Windows. Hello, Apple macOS people, welcome...we mean you no harm. That goes for you Linux gurus, too. We admire Apple users for their good taste in hardware and Linux geeks for their virtue of nerdiness. Windows runs about 90% of the desktop/laptop market worldwide. Microsoft PCs dominate 95% of the *business* world's desktops. Why? Apple is a *consumer* product company.
- ✓ **The [Visual Studio Community IDE](#) used for your C programming course runs only under a Windows operating system. [VS Code](#) can be a viable option for macOS and Linux users.** (There is a "Visual Studio for Mac" but it is similar in name only; VS for macOS does not support C/C++ programming.) Visual Studio IDE is the most used integrated development environment among professional programmers.
- ✓ For weekly activities, please feel free to talk about the activity with your colleagues but use only your own words in your answers. Share ideas, not files. Talk all you want, don't record anything or share files.
- ✓ A suggested format for file naming:
`term_course_Wnn_content.docx`
where *Wnn* is the week number and *content* describes what is in the file, e.g.
`term_CPR101_W01_File systems-VisualStudio.docx`
It's a good file name if, six months from now, you don't have to open the file to know what's in it.
- ✓ **How will your answers be evaluated? What about the content and expression of your answers?**
 - Marks are given only for your original work. Paraphrasing someone else's work does not count as your original work.
 - Marks are awarded for your clear explanation of concepts and ideas. Without proper spelling, good grammar, and appropriate usage, the explanation of concepts and ideas will suffer and accordingly so will the marks.
 - See the course marking rubric. Straightforward answers get straightforward marks. Answering with *why*, in addition to *what* or *how*, gets more marks. In the A and A+

grade range, your professor has found you provided responses beyond or below the surface of the questions. (e.g. how to "delete a file from a PC permanently so it cannot be recovered"? There is more than one way.) An A+ submission is easy to read and insightful. It communicates your answers clearly, correctly, completely, concisely, concretely, and with consideration for your reader; creativity is always welcomed. An A+ answer has the quality of articles you might read on reputable news sites or in print.

Part 1: Visual Studio Community introduction (33.3%)

Complete the Visual Studio project creation and Hello World program as described below.

Use Visual Studio IDE (Integrated Development Environment) which is available on all Seneca lab PCs or for your own Windows PC, or Visual Studio Code which is cross-platform. macOS users who know about Xcode are already fine with that.

See the CP4P_Visual-Studio-demo document for details and for full marks.

Answer these → questions in the ..._Activity-Answers.docx

Explanations and approaches to these questions is offered here in the instructions document. Skipping the details in the instructions skips marks. (If you are reading this, thanks, you are on the right path.)

- 1. What platform are you working on: Windows, macOS, Linux?
What code editor / IDE did you use to create the demonstration `helloWorld.c` source file?
- 2. Where is your `helloWorld.c` source file? Find the file and paste its Full Path:
 - Windows users: hints at using File Explorer to find your Visual Studio project's files are in the VS demo document.
 - macOS users: this may be useful "[Find the Absolute Path to a folder or directory in Mac OS X](#)", also [THIS](#) – see Part 1: 1. 4. 5. and ignore the rest.
- 3. What is in the `helloWorld.c` file? Source code, of course. Copy the text of your customized source statements as per the Visual Studio demo instructions and paste in the box in your answers document.

N.B. there is no need to submit the VS project or the .c source file itself with this week's activity...just your C source code.

Part 2: Source Code editing (33.3%)

- ➔ 4. Explain what your editor's code completion feature does for you and how you use it when writing code. E.g. IntelliSense
- ➔ 5. Other than the system wide shortcuts (e.g. Ctrl+ Save, Zundo, Xcut, Copy, Vpaste, Find), what three shortcuts will you find useful in your source code editor and explain why.
- ➔ 6. Compare the efficiency of using a mouse verses a keyboard when editing source code. (Sure, you are probably faster using a mouse now. But after thousands of lines of code compared to someone who uses mostly keyboard shortcuts?)

Part 3: Managing and Backing up your files (33.4%)

You can work with .c source files, Visual Studio projects (in **repos** or **Projects** folder), or any other course files ...

- on a USB drive for portability between home and school,
 - easy to carry but easy to lose
- in Microsoft 365 OneDrive, available wherever you have a network connection,
 - If you haven't yet tried Microsoft 365 OneDrive at Seneca, now is the time: 1TB of cloud storage. Install OneDrive (the local version) on your computer – even your smartphone – or, in a pinch, click and drag from local folders to the web app.
- on your own computer.
 - Your laptop is fine if you don't mind taking it, and its power brick, with you everywhere.

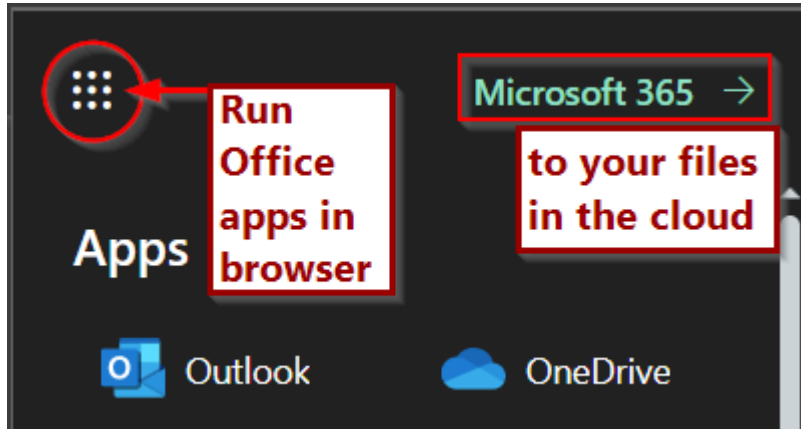
ICT professionals always backup their files. That is, at least one copy of your files exists somewhere else than the system you created the files on.

Decide where to keep the active version of your files. The other two options will then be for backup.

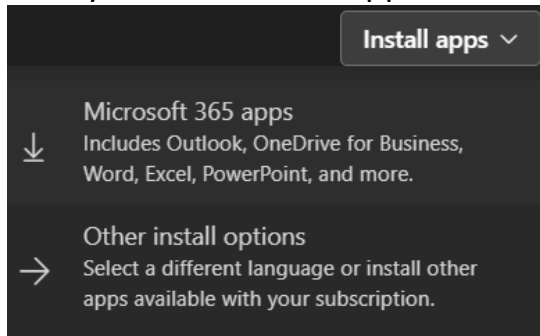
Synchronization of your PC's files with a cloud data service makes your system *interdependent* with the cloud. The synchronization is convenient by making you and your files geographically independent. Literally. The files *coexist* in two or more places at the same time. This is very handy when working between school and home, or lab PC and laptop, or other combinations. *But synch is neither a copy nor a complete backup* because those systems/platforms are not independent.

MS 365, Office apps

Sign on to mySeneca.ca, click on the grid in the top left which opens the Office Apps menu for online / browser access – useful for basic editing of files stored on OneDrive in the cloud.



Clicking on Microsoft 365 takes you to office.com. Look near the top right. You'll see the control to "Install apps" locally on your computer. Compared to the online version, locally installed Office apps are faster and have more features.



You can use either online or local Office apps to access your files 'in the cloud', i.e. on OneDrive.

➔ 7. Where will you keep the active version of your files?

- on a USB drive with copies on your computer and OneDrive?
- in Microsoft 365 OneDrive with copies on your computer and USB?
- on your computer with copies on OneDrive and USB?

... and why did you decide that?

➔ 8. On a practical basis, how will you manage between home and school so that you work only with the active file, never change the copies, but keep the copies up to date with the active file after changes?

Be careful here. Synchronising your local drive with any cloud storage means one file is in two places at the same time – delete it here and it is gone there. (Microsoft OneDrive calls it backup, but it is really synch'd. See [THIS](#) but ignore their software pitch.)

Made a decision? Time to backup.

A good backup software option is [Duplicati](#). See [this review](#).

Who? You do, or you've set up an automatic backup job.

When? When files are changed. This likely happens every day.

What? A backup is TWO COPIES in addition to the original active file.

Where? At least one copy is in a geographically separate location on an independent system. Neither copy is on the system with the original active file.

Why? Has the storage location with your active files ever failed on you?

No? When it does, you'll know why.

Have you ever accidentally lost or deleted a file and could not get it back?

Never mind, we all know the answer to that one.

The active file on your computer or in the cloud does not count as a copy. It is the original. A copy stored on the same system does not count because the system could fail. Will the cloud fail? Not likely. Can your file evaporate in a cloud? Oh yes, lots of ways. Any way can ruin your day.

- TWO copies
 - One copy is on a nearby USB removable/external drive or a local network attached storage system – something you can put your hands on. This copy must be available if an Internet connection is not.
 - The second copy can be in the cloud, but not the same cloud as the active file or the first copy.
- geographically separate location for storing a copy = a safe place nowhere near the active files
 - If Godzilla wrecks your building, your backup is safe in an offsite location. (The movie always ends before Godzilla wrecks the offsite location, too.)
 - Cloud storage is a good option. Email works in a pinch.

- Encrypt offsite backups. E.g. use 7zip archives with AES-256 encryption then copy the .7z file offsite.
 - an independent system = once the backup is done, the backup device / drive / location is disconnected from the source of active files.
 - This means the external drive is unplugged. The cloud data storage location is *not* synchronized with your computer's file system.
 - If the copy is on the same drive, a plugged-in USB drive, or is in an available network location / mapped network drive, then it is on a *dependent* system and is subject to accidental deletion, ransomware encryption, and other misfortunes.
 - If two clouds are used for backup, they are likely in geographically different locations and not connected to each other (that's good), but because neither is available without an Internet connection (that's bad), there is a critical dependency over which you have no control: your ISP. Thus the need for a locally available copy.
- ➔ 9. Considering the above, where are the locations of your two backup copies? How frequently do you do a backup? Exactly how does the backup get done? (click and drag?, script?, backup utility?) If your active files became completely unavailable tomorrow morning, what would you do then?

Notes

Make a copy of all your programming projects by storing the `/repos` or `/Projects` folder in an encrypted .zip archive, then copy/move the archive file to a geographically separate, independent system.

Some (will) use GitHub for backup. DO. NOT. DO. THAT.

Not convinced? <https://devops.com/why-do-you-need-github-backup/>

TL;DR Don't do it.

The principle of GitHub is synchronizing repos for software development. Its [use case](#) and feature set never included backup. Google it and all you'll find is how to back up your GitHub repos outside of GitHub. If you are not paying for a private GitHub repo, then it's public. Anyone can fork your project and then you'll be forked. Doing a commit and push does the job...sort of. Sure, you can roll back to prior versions, but that will get crazy messy with different mixes of files in each push. Getting push versus pull mixed up at 4am or discovering the many

ways as yet unknown to shoot yourself in the foot while [kludging](#) GitHub as ersatz backup...please let us know how that went for you.

Backup is a place where your data does not change unless you upload another backup version. Even then, at least one previous version should be maintained.

When things go wrong (not if)

Windows File History (macOS Time Machine) backs up and provides a generational version history of files and folders referenced in “[Libraries](#)”. This is a good option for your second backup copy because its storage location can be independent of your computer’s file system. Libraries, seen in File Explorer, are virtual containers of files and folders on the local computer. Windows, by default, has Libraries which reference your Documents and other commonly used folders. External drives and other folders or locations to be included in File History must be referenced in a new or existing library. Removable drives can be referenced in a library, but beware, they are not included in the File History.

USB file recovery: Web pages suggesting the use of CHKDSK and ATTRIB to recover USB file deletions always offer an option “if that doesn’t work” to “buy our software”. That is because CHKDSK and ATTRIB simply will not, never have, and *cannot* recover *deleted* files. That is *not* what those DOS/Windows operating system commands can do. [This site](#) is the only one we’ve seen that is honest about exactly what those commands do, but those are for rare cases, not simple file deletion.

USB flash drives are wonderfully portable which makes them easily losable. If you do lose your drive, usually by forgetting it in the lab PC, how will it ever get back to you?

*Little Bo-Bleep
has lost her USB-flash-drive-with-files-on-each-of-her sheep,
And doesn't know where to find it;
Without her name,
Home ne'er it came,
But would have if she had signed it.*

(Sadly, she didn't have a backup.)

- Rename your USB drive's volume label to your name.
- Put a **!!_PLEASE_RETURN_TO_!! .txt** file in the root with your email address, contact information, and a nice message promising a reward. (coffee, at least)
- Put your phone number or email on the outside of the USB drive.
- And it might come home wagging its data behind it.