Installing Software

https://csci-1301.github.io/about#authors

May 19, 2021 (11:33:49 PM)

Contents

1 Generalities on Installing Software

You probably already installed a software in your life, be it VLC[1], Microsoft Teams[2], or Whatsapp[3]. However, depending on if you installed it on a phone, a tablet, a computer, and depending on the operating systems (Android, Windows 10, iOS, Ubuntu, etc.) your experience may have varied drastically.

Between the Play store, the command-line interface homebrew[4] and the act of downloading a software using your browser and then installing it using the navigator, there can be a lot of differences, but in all those circumstances you should keep security in mind. In addition of making sure that you are downloading the software from a trusted source, you should also be vigileant about the information the software will be able to access about e.g. your private life.

As data can be lost or corrupted upon downloading, many plat-form now uses checksum[5] to verify the integrity of the software you downloaded before installing it. This is an excellent practise that can also be performed "by hand", as explained for instance for the database manager MySQL[6]: the main idea is that the probability of the signature matching a tampered-with file is extremely low, and that as long as you are downloading the signature and the software from two different sources, you are considerably reducing the attack surface[7].

---

[1]http://www.videolan.org/
[2]https://www.microsoft.com/en-us/microsoft-teams/download-app
[3]https://www.whatsapp.com/
[4]http://brew.sh/
[5]https://en.wikipedia.org/wiki/Checksum
[6]https://dev.mysql.com/doc/refman/8.0/en/verifying-package-integrity.html
[7]https://en.wikipedia.org/wiki/Attack_surface

2 Executing Code Found on-line

As you progress in this class, you will be asked more and more to download and execute code hosted in our repository[8]. How can you tell that you can trust this code?

We have not implemented a checksum-matching (yet!), but you can trust this code as it was coded by your instructors, and hosted on a plat-form using two-factor authentication[9] and were every action is tracked using versioning[10]. Concretely, this means that only somebody who manages to steal your instructor's credentials, their phone, and thwart all the other instructor's vigilance would be able to host malicious code on our plat-form: while we certainly imagine that this is theoretically possible, we hope that you will agree that the probability is low enough for you to trust the code on this site.

As often, security is not absolute, but aims at providing reasonnable confidence. Executing "blindly" code found on-line, on the other hand, gives a good chance of facing unpleasant surprises: while there certainly is a lot of useful, good code on websites like stackoverflow[11]—your instructor probably uses such websites, by the way!—, copying-and-pasting it without understanding its purpose or general structure is almost guaranteed to, at best, not execute properly, at worst, make your system unstable or insecure.

3 Accessing an IDE

An IDE[12], for "Integrated development environment", is the software or service you will be using to write, compile, execute and debug your code. There are many available IDEs, and some can accomodate multiple different programming languages.

For C#, the main two choices are Visual Studio[13] and MonoDevelop[14]. The former is available on Windows, Mac OS and on virtual machines, the latter is available on all plat-forms (including but not limited to Linux operating systems).

To access one or the other, you will need either

- a computer with the right to install software on it,
- to access one of the computers in the computer lab[15], or
- a computer with internet access.

The third solution is a backup plan, as instead you will access a very minimal version of an IDE to test small snippets of code. You should not rely on it for the duration of this course.

3.1 Installing an Ide On Your Own Computer

This part gathers some references for you to install MonoDevelop or Visual Studio on your own computer, regardless of your operating system. It is strongly encouraged that you do so, especially if you want to continue in a CS/IT/Cyber degree, but is not mandatory[16]. The instructions are detailed, but there are plenty of ways this can go wrong: make sure you have read and followed those instructions carefully before asking for help! Once you've identified your issue, try again from scratch to see if you missed a point, and if you are still facing the same issue and can describe it, feel free to look for set-up help on the internet, from your classmates, and from your instructors.

---

[8]https://github.com/csci-1301/csci-1301.github.io/
[9]https://docs.github.com/en/github/authenticating-to-github/about-two-factor-authentication
[10]https://en.wikipedia.org/wiki/Software_versioning
[11]https://stackoverflow.com/
[12]https://en.wikipedia.org/wiki/Integrated_development_environment
[13]https://visualstudio.microsoft.com/
[14]https://www.monodevelop.com/
[15]https://www.augusta.edu/its/computers-printing.php
[16]Unless this class is fully online, of course.

3.1.1 Installing MonoDevelop On Your Own Computer

Unfortunately, MonoDevelop offers pre-packaged release only for linux distributions

- If you are using linux (e.g. Ubuntu, Debian, etc.), then please head out to MonoDevelop's download page[17].
- If you are using MacOS, you can have a look at the compilation instructions[18], but it is very likely that you will find them impossible to understand.
- If you are using Windows, you can have a look at the compilation instructions[19], but it is very likely that you will find them impossible to understand.

If you are a MacOs or Windows user, you could use a Virtual Machine[20] manager to run a linux-based distribution on top of your operating system, but it is simpler to pick Visual Studio instead.

3.1.2 Installing Visual Studio On Your Own Computer

Note that we are *not* installing "Visual Studio *code*", but simply "Visual Studio".

3.1.2.1 For Windows

1. Visit Azure Dev Tools for Teaching[21].

2. Log in using your Augusta University credentials.

3. Select "Download software".

4. Look for Visual Studio. The path is Education → Software → Visual Studio Enterprise 2019. You can search "Services" for the "Education" group and then click "Software" if the education group is not immediately displayed. It should look like the following:
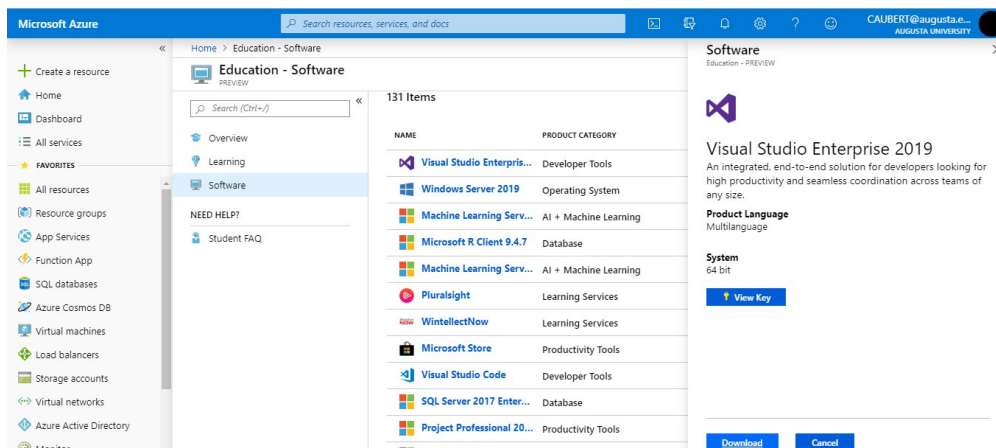
Normally, the following direct link should get you to the right page: https://portal.azure.com/?Micro soft_Azure_Education_correlationId=8ee63052-dc32-46f7-a109-e26793622dbf#. Type "Education" in the search bar to get started on the path to the program to download.

---

[17]https://www.monodevelop.com/download/#fndtn-download-lin
[18]https://www.monodevelop.com/developers/building-monodevelop/#macos
[19]https://www.monodevelop.com/developers/building-monodevelop/#windows
[20]https://en.wikipedia.org/wiki/Virtual_machine
[21]https://aka.ms/devtoolsforteaching

5. Download and install Visual Studio (leave all the options on their default settings).

   If possible, check the box next to "Workload" → "Windows" → ".NET-Desktop Development", or to "Workload" → "Other Toolsets" → ".NET Core cross-platform development":

6. Enter the product key you obtained previously, following the instructions in the documentation[22]. Normally, clicking on "View key" on the screen pictured in the fourth step above should give you access to a key, that you simply need to copy-and-paste in the menu you can access on Visual Studio by clicking on "Select File" → "Account Settings" → "License with a Product Key".

3.1.2.2 For Mac and Other Linux Systems
You can either install another version of Visual Studio or "emulate" Windows on your computer.

For the first option, download a version of Visual Studio at https://visualstudio.microsoft.com/vs/mac/. It differs a bit from the Windows version, but that should not impact your experience in this class. The only Visual Studio feature we rely on is the ability to create "Console Apps with C#", which is equally available in both the Windows and Mac versions.

For the second option, you will need a Virtual Machine[23] manager. This option works for Linux systems as well.

1. You can use

   a) "VMware Fusion 10.x Pro" (only for MacOS, available on OnTheHub[24],
   b) Virtual Box[25] (for Linux and Mac),
   c) Virtual Machine Manager[26] (for Linux).

2. Download a version of "Microsoft Operating Systems" from Azure Dev Tools for Teaching[27],
3. Install and run your version of Windows from your virtual machine, and follow the instructions for windows to install Visual Studio.

---

[22]https://docs.microsoft.com/en-us/visualstudio/ide/how-to-unlock-visual-studio?view=vs-2019
[23]https://en.wikipedia.org/wiki/Virtual_machine
[24]https://e5.onthehub.com/WebStore/OfferingDetails.aspx?o=637dd37b-06b5-e711-80f7-000d3af41938&pmv=00000000-0000-0000-0000-000000000000&ws=2020165a-723a-de11-b696-0030485a8df0&vsro=8
[25]https://www.virtualbox.org/
[26]https://virt-manager.org/
[27]https://aka.ms/devtoolsforteaching

3.2 Accessing One of the Computers in a Computer Lab

Please refer to this page from AU's Information Technology[28] to know where the computer labs are located. Visual Studio should be pre-installed on every computer.

3.3 Compiling Code On-Line

As a backup or only to test snippets of code, you can compile C# code online. Multiple online platforms exist, such as:

- https://www.tutorialspoint.com/compile_csharp_online.php
- https://www.onlinegdb.com/online_csharp_compiler
- https://www.jdoodle.com/compile-c-sharp-online/
- https://dotnetfiddle.net/

Note that none of them are endorsed by the school and that they can pose security and privacy challenges: never enter any sensitive information and do not rely on them too heavily. However, they can be a good support if you would like to test a short snippet of code but do not have access at the moment to a computer with MonoDevelop or Visual Studio installed.

---

[28]https://www.augusta.edu/its/computers-printing.php