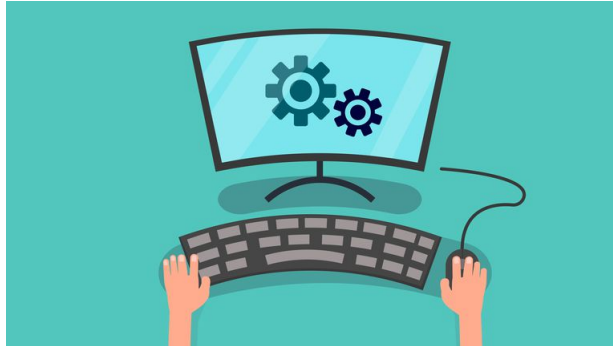


**CMPSC 102  
Discrete Structures  
Fall 2019**

**Practical 1:  
Installing Software and a Written Reflection**



## Summary

To learn how to set up a container-based platform, called Docker, for use in all class activities, labs and practicals. To configure Github and Github Classroom and a text editor called, Atom, for use in the class. Finally, you will be writing a page-long reflection piece about the challenges and solutions you encountered during the installation procedures.

### 1. Using a Container-based Platform, Docker



Docker is a platform for developers and system administrators to develop, deploy, and run software applications with containers. Docker is a new generation of virtualisation and it is revolutionizing the IT industry. Using Docker software developers no longer need to worry about installations and software versions on various operating systems their product may need to run on. All the necessary software is packaged into a container and software applications are run in it. Companies, such as Visa and PayPal, and universities, such as Cornell and Indiana University, have adopted the Docker platform since it simplifies the process of building, running, managing and distributing software applications.

The Department of Computer Science has adopted Docker in all computer science courses. The instructor in this course will deploy Docker containers with all the necessary software to run all

class exercises, lab and practical assignments. Students will then be able to work on developing programs on their own machines and run them using instructor-provided Docker container without having to install a dozen of software languages and tools. Before being able to use Docker, students must first complete basic installations of Docker and Git, which will be used for code distribution and submission. These installations are specific to the operating system of your machine.

1. As the first step of the Docker set up process, please check and make a note of the version of your operating system. If your laptop meets the requirements outlined below, then proceed to the next step. For older MacOS and Windows Operating Systems, including Windows 10 Home, see the next section (1.1) about Docker ToolBox.

### **Docker Desktop Specifications**

Mac:

- 2010 model or newer with hardware support for MMU, EPT, and Unrestricted Mode
- macOS Sierra 10.12 or newer
- 4 GB of RAM
- VirtualBox prior to version 4.3.30 cannot be installed

Windows:

- Windows 10 64bit Pro, Enterprise, or Education (Build 15063 or later)
- Virtualization is enabled in BIOS
- CPU SLAT-capable feature
- 4 GB of RAM

Linux:

- 64bit
- Kernel 3.10 or later

2. Now, go to <https://docs.docker.com/install/> and from the menu on the left-hand side select and follow the installation tutorial for your operating system (Linux, MacOS, or Windows). For example, if you have MacOS you can find installation guide on <https://docs.docker.com/docker-for-mac/install/>, if you have Windows operating system on (<https://docs.docker.com/docker-for-windows/install/> and for Linux you can proceed to <https://docs.docker.com/install/linux/docker-ce/binaries/>.

Please note that the Docker Desktop for Mac and Windows requires you to create an account to download. To bypass this, use the following download links:

- Windows: <https://download.docker.com/win/stable/Docker%20for%20Windows%20Installer.exe>
- Mac: <https://download.docker.com/mac/stable/Docker.dmg>

3. In this class we will use a command-line interface, which allows the user to interact with the computer by typing in commands in a “terminal” window. Computing professionals prefer to use the command line interface, built into operating systems like Linux, instead of using the graphical user interface. In many situations command line interface tends to be very efficient and effective, for example, it allows you to complete some tasks with a simple one line command instead of using the “pumping” motion of the mouse!

Once your docker installation is complete, please locate a terminal window on your machine. In MacOS and Linux, you can search for a “terminal” and on Windows machines look for “cmd” (command prompt). If you are using Windows, we recommend using a Linux-based terminal, a step that you will complete when installing Git in Section 2, step 2.

Now in the terminal window type `docker run hello-world` command and press “Enter”. If you see an output similar to the one below, feel free to high five a technical leader - your docker set up was successful!

```
docker run hello-world
```

```
docker : Unable to find image 'hello-world:latest' locally
...
```

```
latest:
```

```
Pulling from library/hello-world
```

```
ca4f61b1923c:
```

```
Pulling fs layer
```

```
ca4f61b1923c:
```

```
Download complete
```

```
ca4f61b1923c:
```

```
Pull complete
```

```
Digest: sha256:97ce6fa4b6cdc0790cda65fe7290b74cfebd9fa0c9b8c38e979330d547d22ce1
```

```
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

```
...
```

### 1.1. Docker Toolbox for older Mac and Windows systems

Older Mac and Windows operating systems, such as Windows Home, that do not meet the requirements of Docker Desktop for Mac and Docker Desktop for Windows can use Docker Toolbox. You can follow the tutorial on [docs.docker.com/toolbox/toolbox\\_install\\_windows/](https://docs.docker.com/toolbox/toolbox_install_windows/), while keeping the following steps in mind.

1. Make sure your operating system version satisfies the requirements for the Docker ToolBox as outlined in the Step 1 of the Toolbox installation tutorial.

2. Check that virtualization is running using Specci as described in the Toolbox installation tutorial. If not enabled, then the BIOS settings must be changed. For example, to change this setting on an HP Windows 10 Home, you should first shut down your computer. Secondly, you turn on your machine, while holding the `esc` button and wait for the BIOS menu to appear. Then, select the `F10` key for the BIOS configuration. Now you should be in BIOS settings, where you can use the arrow keys to navigate to the “System configuration” tab and select “Virtual Technology” entry. Push `enter` and select “Enabled.” Finally, use `F10` key to “Save and Exit” and reboot your computer. Please note that the specific keys and operations are most likely to be different on your machine.
3. After you start the Docker Toolbox Set Up Wizard, you will be guided through the installation process. When you get to the list of components to select, make sure that the “VirtualBox” and “Git for Windows” options are checked.

## 2. Configuring Git and GitHub



During the subsequent practical and laboratory assignments, we will securely communicate with the GitHub servers that will host all of the project templates and your submitted deliverables. In this assignment, you will perform all of the steps to download Git and configure your account on GitHub, so that you are ready to start your first lab assignment using GitHub Classroom next week. You can also learn more about GitHub Classroom by visiting <https://classroom.github.com/>. As you will be required to use Git, an industry standard tool, in all of the laboratory and remaining practical assignments and during the class sessions, you should keep a record of all of the steps that you complete and the challenges that you face. You may see the course instructor or one of the teaching assistants if you are not able to complete a certain step or if you are not sure how to proceed.

1. If you do not already have a GitHub account, then please go to the GitHub website (<https://github.com/>) and create one, making sure that you use your “allegHENY.edu” email address so that you can join GitHub as a student at an accredited educational institution. You are also encouraged to sign up for GitHub’s “Student Developer Pack” at <https://education.github.com/pack>, qualifying you to receive free software development tools. Additionally, please add a description of yourself and an appropriate professional photograph to your GitHub profile. Unless your username is taken, you should also pick your GitHub username to be the same as Allegheny’s Google-based email account.
2. If you have not done so already in the previous section (1), install Git for the operating system of your laptop. Please note that if you installed Docker Toolbox and checked the

“Git for Windows”, you do not need to install Git again. Please see suggestions below relevant to your operating system.

**For Windows:** We recommend you complete the following steps.

- Open a terminal (command prompt) window in administrator mode. To do this, search for “cmd” in the start menu, then right-click and select “Run as Administrator”.
- Then, copy and paste the following command into that terminal, and hit enter.

```
$ @"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe"
-NoProfile -InputFormat None -ExecutionPolicy Bypass -Command
"iex ((New-Object System.Net.WebClient).DownloadString
('https://chocolatey.org/install.ps1'))" &&
SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
```

- To test that this step was completed successfully, open a new terminal, and type the command “choco --version”. If 0.10.11 is printed (or some other higher version), you have completed this step. If not, ask for the instructor or a technical leader for assistance.
- Run the following commands in a terminal window in administrator mode.

```
choco install git -y --params "/GitAndUnixToolsOnPath
/WindowsTerminal /NoShellIntegration"
```
- To test that this step was completed successfully, open a new terminal and type:

```
git --version.
```

 If git version followed by some version number is printed (or something similar), you have completed this step.

**For Mac:** In the terminal type: `git --version`. If Git is not installed on your machine already, then you will be prompted to install it with the Git installer.

**For Linux:**

- Type and run the following command in the terminal:

```
sudo apt-get install git
```
3. If you have never done so before, you must use the “ssh-keygen” program to create secure-shell keys that you can use to support your communication with GitHub. These keys enable you to send your files to GitHub without having to type in a password each time. But, to start, this task requires you to type commands in a terminal.
  4. Open the terminal, a window in which you can type commands, and now type the “ssh-keygen -t rsa -b 4096 -C your\_email\_used\_to\_create\_github\_account@allegheny.edu” command in it. Follow the prompts to create your keys and save them in the default directory. That is, you should press “Enter” after you are prompted to “Enter file in which to save the key...” and then type your selected passphrase whenever you are prompted to do so. Please note that a “passphrase” is like a password that you will type when you need to prove your identify to GitHub. What files does “ssh-keygen” produce? Where does this

program store these files by default? Invite a technical leader to investigate the answers to these questions.

5. Once you have created your ssh keys, you can raise your hand to invite either a technical leader or the course instructor to help you with the next steps as needed. First, you must log into GitHub and look in the right corner for an account avatar with a down arrow. Click on this link and then select the “Settings” option. Now, scroll down until you find the “SSH and GPG keys” label on the left, click to create a “New SSH key”, and then upload your ssh key to GitHub. You can copy your SSH key to the clipboard by going to the terminal and typing “`cat ~/.ssh/id_rsa.pub`” command and then highlighting this output. When you are completing this step in your terminal window, please make sure that you only highlight the letters and numbers in your key—if you highlight any extra symbols or spaces then this step may not work correctly. Then, paste this into the GitHub text field in your web browser.
6. Again, when you are completing these steps, please make sure that you take careful notes about the inputs, outputs, and behavior of each command. If there is something that you do not understand, then please ask the course instructor or the teaching assistant about it.

### 3. Text Editor for Programming



In this class we will use a text editor called, `atom`, to write and edit software. Please go to <https://atom.io>. You should see a Download button specific to your operating system. Use this link to install this editor onto your machine.

### 4. Reflection

Once you have completed all necessary installations, it is time to pull over your first GitHub Classroom repository.

#### GitHub starter link

<https://classroom.github.com/a/mNeYLLLL>

To use this link, please follow the steps below.

- Click on the link and accept the assignment.



- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the practical locally.
- As you are working on your practical, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):

```

- git add -A
- git commit -m ‘Your notes about commit here’
- git push

```

Alternatively, you can use the following commands to add multiple files from your repository:

```

- git commit <nameOfFile> -m ‘Your notes about commit here’
- git push

```

Be sure to read the README.md file in the GitHub Classroom repository for instructions on how to complete your first assignment.

## General Guidelines for Practical Sessions

Since this is your first assignment and you are still learning how to use the appropriate software, do not become frustrated if you make a mistake. Instead, use your mistakes as an opportunity for learning both about the necessary technology and the background and expertise of the other students in the class, the teaching assistants, and the course instructor.

- **Experiment!** Practical sessions are for learning by doing without the pressure of grades or “right/wrong” answers. So try things! The best way to learn is by trying things out.
- **Complete *Something*.** Your grade for this assignment is a “checkmark” indicating whether you did or did not complete the work.

- **Practice Key Laboratory Skills.** As you are completing this assignment, practice using the Ubuntu terminal until you can easily use its most important features. Additionally, ask a teaching assistant or a course instructor to teach you some of the advanced features of the terminal, thereby helping you to work more effectively.
- **Try to Finish During the Class Session.** Practical exercises are not intended to be the equal of the laboratory assignments. If you are simply a slow typist, I've given you until the end of the day, but ideally you should complete the assignment by the end of the class period.
- **Help One Another!** If your neighbor is struggling and you know what to do, offer your help. Don't "do the work" for them, but advise them on what to type or how to handle things. If you are stuck on a part of this practical session and you could not find any insights in either your textbook or online sources, formulate an intelligent question to ask your neighbor, a teaching assistant, or a course instructor. Try to strike the right balance between asking for help when you cannot solve a problem and working independently to find a solution.