

02_Setup_Computational_Environment

Published

Assign To

Edit

:

Due: Before Week 2

Format: Individual, completion-based

Goal: Ensure every student has a fully working computational environment for the course.

I will be away during Week 1, so this assignment is designed to be completed independently.

Take your time and follow each step carefully.

This setup gives you a modern scientific computing stack.

Learning Objectives

By completing this assignment, you will:

- Set up a professional programming environment
- Install tools used in scientific computing and research
- Create your first reproducible software workspace
- Learn the basics of version control and environments

Required Tools

You will install and configure:

- **Visual Studio Code**
- **Miniconda (Python + environment manager)**
- **GitHub Desktop**
- **A GitHub account**

These tools will be used throughout the course.

Part 1 – Create a GitHub Account

1. Go to <https://github.com> 

2. Click **Sign up**

3. Choose:

- A professional username (your name or similar)
- An email address you check regularly

4. Choose the **Free** plan

5. Verify your email

✓ Checkpoint

- You can log in and view your GitHub profile page.
-

Part 2 – Install Visual Studio Code

1. Go to <https://code.visualstudio.com> ➔ (<https://code.visualstudio.com/>)
2. Download and install the version for your operating system
3. Launch VS Code

Install extensions

Open the Extensions panel and install:

- **Python**
- **Jupyter**
- *(Optional) GitHub Pull Requests and Issues*

✓ Checkpoint

- VS Code opens and extensions install successfully.
-

Part 3 – Install Miniconda

1. Go to <https://docs.conda.io/en/latest/miniconda.html> ➔ (<https://docs.conda.io/en/latest/miniconda.html>)
2. Download **Miniconda (Python 3)** for your OS
3. Install using default options

- Allow Conda to initialize your shell if prompted

Verify installation

Open a terminal (a powershell on Windows, or Applications/Utilities/Terminal on MacOS) and run:

```
conda --version
```

If it says something like 'conda not found', then it isn't on your "path" (the list of directories where your operating system looks for programs). Troubleshoot with ChatGPT, or similar.

✓ Checkpoint

- A Conda version number is printed.

Part 4 – Create a Conda Environment

In your terminal:

```
conda create -n compsci python=3.11
```

Activate it:

```
conda activate compsci
```

Install packages:

```
conda install numpy scipy matplotlib jupyter pandas
```

✓ Checkpoint

- Your terminal prompt shows `(compsci)`.

Part 5 – Install GitHub Desktop

1. Go to <https://desktop.github.com>  (<https://desktop.github.com/>)
2. Download and install
3. Sign in using your GitHub account
4. Configure your name and email

✓ Checkpoint

- GitHub Desktop opens and shows you as logged in.
-

Part 6 – Create Your Course Repository

1. In GitHub Desktop, click **New Repository**

2. Name it:

```
compsci2026
```

3. Choose a location on your computer

4. Click **Open in Visual Studio Code**

✓ Checkpoint

- The same folder is visible in both VS Code and GitHub Desktop.
-

Part 7 – First Program Test

In VS Code, open a terminal and activate your environment:

```
conda activate compsci
```

Create a file called `test.py` containing:

```
print("Hello, computational world!")
```

Run it directly from the Visual Studio Code terminal:

```
python test.py
```

Here is how it should look (approximately):

A screenshot of a terminal window titled "ComputationalSeismology". The terminal shows the following command and output:

```
python test.py
Hello, computational world!
```

Commit the file using GitHub Desktop with a suitable comment. Then "Push origin" (upper right on GitHub Desktop - your might have to click on "Fetch origin" and then "Commit" before this appears). Here is how GitHub Desktop should look (approximately):

The screenshot shows the GitHub Desktop application interface. At the top, it displays the current repository as "GLY6739.017S26_ComputationalSeismology" and the current branch as "main". A message indicates "Last fetched just now". Below this, the "Changes" tab is selected, showing a single change: a new file named "test.py" has been created. The code in "test.py" is a single line: `print("Hello, computational world!")`. The commit message is "Created test.py". The commit button at the bottom is labeled "Commit 1 file to main".

Finally, navigate to your test.py file on github.com. It should look like this (approximately):

The screenshot shows the GitHub web interface for the repository "GLY6739.017S26_ComputationalSeismology". The "Code" tab is selected in the sidebar. The "test.py" file is shown in the main pane. The code content is:

```
1 print("Hello, computational world!")
```

The commit history shows a single entry from "gthompson" with the message "Created test.py" and timestamp "e06592a · now".

✓ Final Checkpoint

- Python runs
 - The file executes correctly
 - The repository has at least one commit
-

Submission Instructions

Submit the following to Canvas:

- A screenshot showing:
 - VS Code open
 - `test.py`
 - Terminal output from running `test.py`.
 - A screenshot of your `test.py` program on `github.com`.
-

Troubleshooting

If something doesn't work:

- Copy the exact error message
 - Take screenshots
 - Get help from other students, or email me at `thompson@usf.edu`
 - Bring the issue to Week 2 — debugging is part of computational science
-

One-sentence takeaway

This week is about making your computer a predictable, reproducible scientific tool.

Points 5

Submitting a file upload

Due	For	Available from	Until
Jan 20 at 9am	Everyone	Jan 13 at 11am	-

[+ Create Rubric](#)[🔍 Find Rubric](#)