D2-1 Resource Document: Course Information and Setup

The Class Website

The website can be found at: obagy.com/cs120

- **Assignments** are at the top. It will be updated weekly with the current assignment.
- **Style Guidelines** gives the required programming style for this class. Be sure to read through it, as you will be graded based on it.
- **Lectures** is where you can find the slide decks from class.

Discord

Discord will be the primary mode of communication for this course.

Discord is a messaging app with servers, where each server has multiple channels. Each channel is a separate chat, based on topic. Please keep in mind that this is an academic context, so do not say or do anything in the Discord that you would not do in the classroom.

Here is the link to join the Discord server:
https://discord.gg/ZwP45bH3a

The link can also be found on the class website.

When you first join, you'll only have access to one channel. Select your class section to unlock the rest of the server.

Here are some other channels:

- `announcements` -- **IMPORTANT** to keep an eye on. This is where any class updates are sent.
- `oh-announcements` -- if a TA needs to reschedule/cancel their office hours, they'll announce it here
- `important-links` -- links to the class website and OH schedule
- `general` -- any general comments or anything off-topic you want to chat about
- `questions-general` -- any non-assignment specific questions
- `questions-short-pa` -- short PA questions
- `questions-long-pa` -- long PA questions

Update your name in the server to be either your netid or First name last initial.

1. At the top, select `CSC 120 Spring 2026` to open the dropdown menu

2. Select `Edit Server Profile` (it's the third to last option)
3. Under “Server Nickname”, change it to either your netid or your first name and last initial (e.g. Katelyn R)
    a. Your netid is the first part of your email, not your student ID number!

Office Hours

Office Hours are managed by a Discord bot, QueueBot, to keep things organized. You will join the queue by sending command messages into the `join-queue` channel:

- `!q join` - Join the **online** queue
- `!q join-inperson` - Join the **in-person** queue
- `!q leave` - Leave the queue
- `!q list` - Get a list of people in line

You must join the queue whether you are online or in-person

Muting a Channel

This server can have a lot of traffic. You can choose to mute a channel so you stop receiving notifications from it. However, please **do not mute the announcements channel!**

1. Right click on the channel you'd like to mute (if on mobile, press and hold instead)
2. Under the `mute channel` option, you have the choice to mute it for a given amount of time, or indefinitely
3. Alternatively, you can change your notification settings to @mentions only in the option below

---

GradeScope

GradeScope is what you will use to turn in your short and long PAs (programming assignments) ICAs (in-class activities), and Discussion Section problems.

How to submit an assignment

When you submit to GradeScope, you will be asked to upload your code files. If you have multiple files to submit, make sure to select **all** of them before uploading to GradeScope. You can resubmit as many times as you'd like, but make sure that your final submission has all of your files.

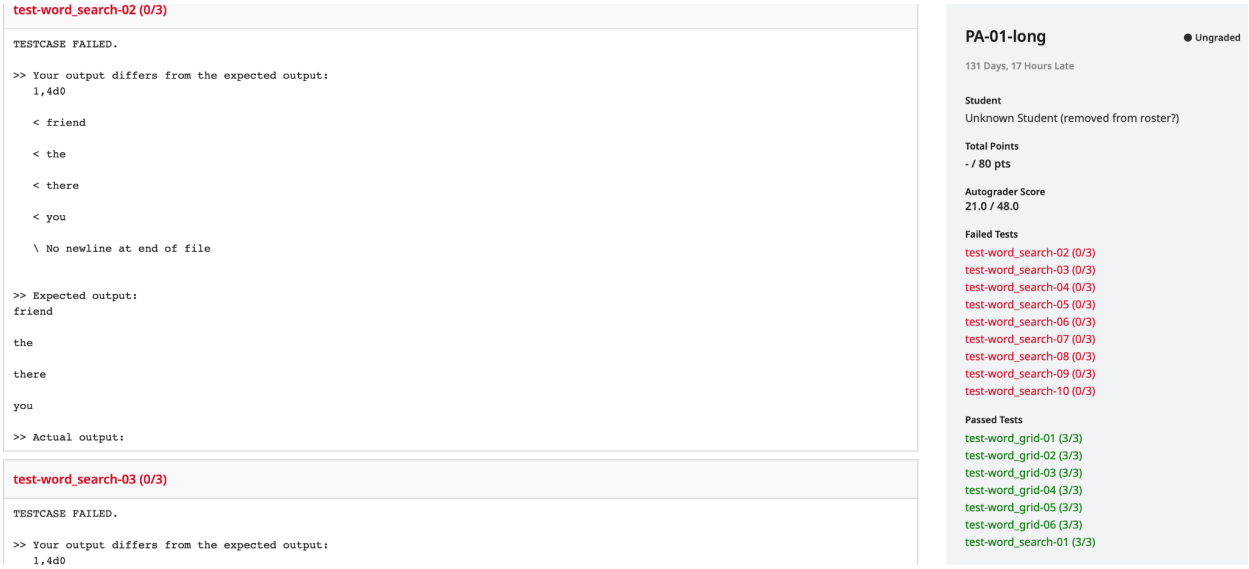
If you're submitting an **ICA**, the best way to submit it is by PDF. Many students use Google Docs to put their code and photos together, then export that to a PDF to submit it. It's helpful to your TA if you put the word of the day on the first page of your document.

If you're submitting a **long PA**, once you submit, the autograder will run automatically. It will run your code and give you feedback on it. The auto grader may take up to a few minutes to run. In order to pass the autograder, you need to have the **exact** output as expected. *Even an extra space could make you fail the autograder!*

How to read the autograder

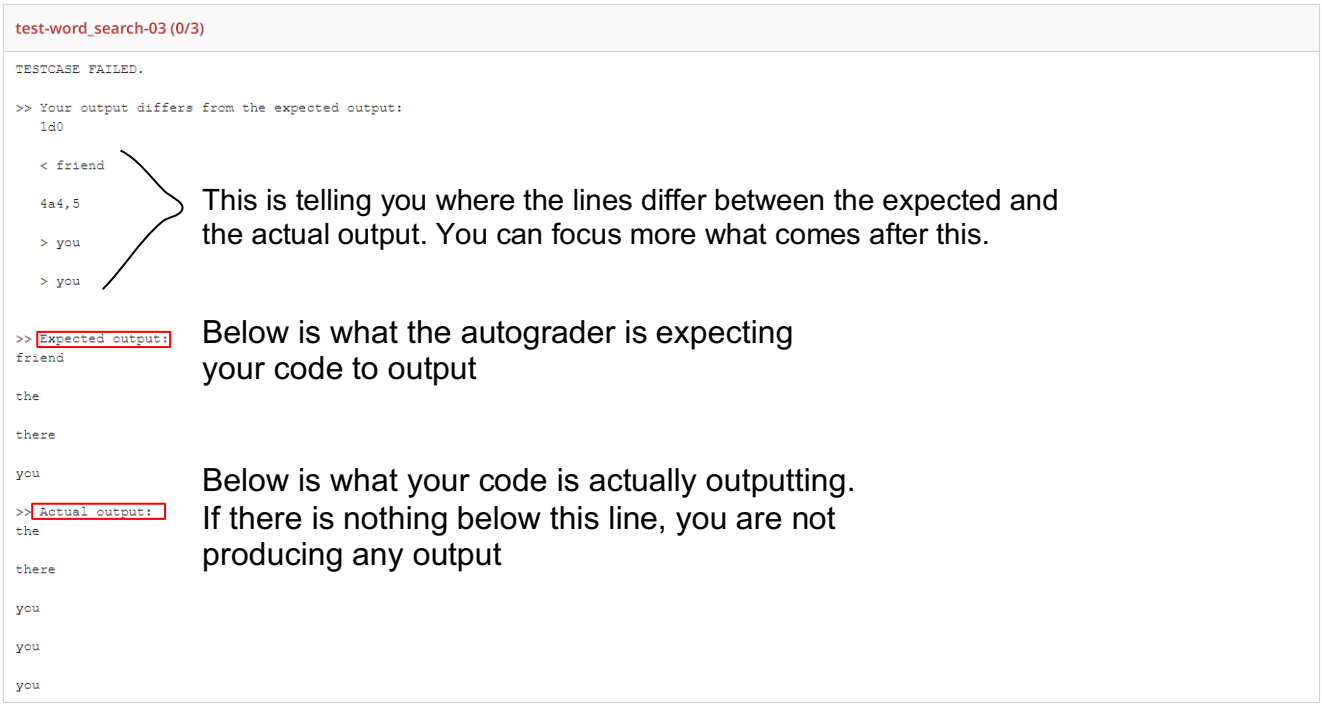
You will see several test cases on the right side of the screen. Green means it passed, red means it failed. You can see the difference between your output and the expected output if you click on the test case.

In the example image below, you can see that the student passed all of the test cases for word grid. However, they failed almost all of the word search test cases. In the expected output, we see four words printed, separated by blank lines. In the actual output, we see nothing being printed. This means that your program isn't printing any output for this test case, despite there being an expected output.



You can also see that this test case is `word-search-02`. You can find the exact input and expected output in the **test cases zip file** on the **assignments** page of the class website!

Annotated Failed Testcase



In general, once you submit your assignment for the last time, whatever score you got on the autograder is final. However, you may lose your autograder points if you hard code the output, or bypass doing the assignment in some way.

D2L

D2L is where you'll go to watch the extra **OCA** videos. OCAs are extra video lectures that are typically weekly–with later ones often being tailored to the current assignments. You can find the OCAs under the **content** tab.

VS Code

VS Code is an IDE (Integrated Development Environment), for writing code. It has useful features like autocomplete and a debugger. For this course, we recommend you use VSCode. If you use another editor, TA's may not be able to help with issues you run into using your editor.

You can download VSCode at https://code.visualstudio.com.

When you open VS code, you will see a welcome tab with options to change colorschemes and settings.
If you want to keep your projects in a common folder, you must first create that folder on your computer, then open this folder through VS Code. Simply choose `File > Open Folder` from the menu. To add files from other folders to your workspace, you can use `File > Add a Folder to Workspace`. If you create new files with a folder open, they will automatically be placed within the open folder.

You can open your first project by selecting `New File...` on the welcome tab, or `File > New File` from the tool bar across the top of the window. Choose a Python file format, or simply add `.py` to the end of your filename.

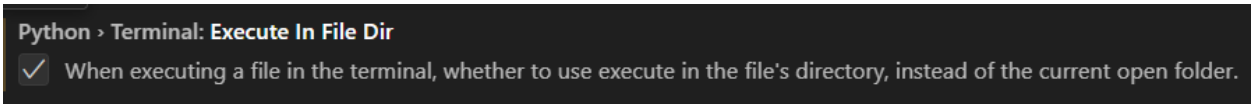
VSCode should prompt you to install the Python extension when you open a Python file. Otherwise, you can install the extension through the extensions tab, the grid icon on the left sidebar, by searching for "Python"

Once you have the extension downloaded, you can run your code with the green arrow in the top corner. Your output will show up at the bottom of the screen in the terminal.

Optional VS Code Tips and Tricks

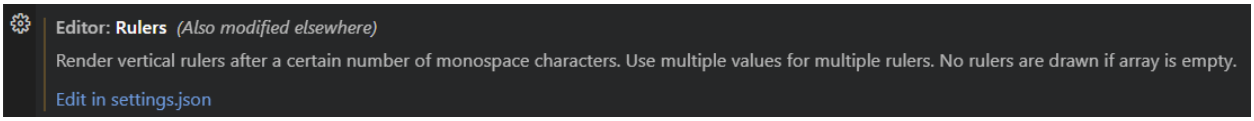
1. **Use your python file's folder to look for files when trying to open/read a file**

In order to make VS Code open and read files in your current folder/directory so you don't have to specify the full file path, you can turn this option on by going into the settings by clicking on the gear in the bottom left corner, or by using the short-cuts (windows: **Ctrl+,** mac: **Cmd+,**). In the settings search bar, type "*execute in file dir*" and hit the checkbox to enable it.



2. **Adding a line ruler to see where the character limit is**

Per the style guide, you cannot have a line longer than 79 characters. To make this easy to keep track of, you can add a small vertical line to indicate where the 79th character is in your editor. To add this, go into the settings and search "*ruler*" and look for the image below:



Click the "Edit in settings.json", and then add or modify the file so that it contains the content in the image below as an item within the curly braces. Make sure to save with ctrl/cmd + s

```
"editor.rulers": [
    79
],
```

3. **Use your python file's folder to look for files when trying to open/read a file when using the debugger**

Getting the debugger to read files from your program's directory is a bit different than tip 1—you only really need to do this if you plan on using the debugger. Go to the settings and search "*launch*" and click to edit the settings.json. This will add a new item to the bottom of the file. Just copy and paste the following so the new part of your settings looks like this with the rest of the settings left untouched:

```
"launch": {
        "version": "0.2.0",
        "configurations": [
            {
                "name": "Python: Current File",
                "type": "debugpy",
                "request": "launch",
                "program": "${file}",
                "console": "integratedTerminal",
                "justMyCode": false,
                "cwd": "${fileDirname}",
                "purpose":["debug-in-terminal"]
            }
        ],
        "compounds": []
    }
```