# Process Journal
# Instructions and Sample

## Overview

The process journal documents your progress as you implement and test your code. It should include short written reflections and screenshots showing the development of your Python file at each stage. The journal can be typed (recommended) or handwritten. *You should create your own document for the process journal.*

The process journal should include the following components:

1. **Understanding the Problem [must have]:** Explain what the problem is asking you to do and how you plan to approach it.
2. **Writing the First Version of the Code [must have]:** Show and describe your initial implementation.
3. **Running and Testing [must have]**
    a. Why did you choose those particular tests?
    Typically, we offer a couple of sample tests. However, you should have more tests than what we provided: the minimal requirement is at least 3 different tests for each function, including the ones we show as examples.
    b. What were your expected outputs?
    c. Describe what happened when you ran your tests. Include any errors or unexpected behavior.
4. **Debugging and Adjustments:** Explain how you identified and fixed errors, including any changes you made to the code.
5. **Further Testing and Refinement:** Document any additional tests or adjustments after debugging.
6. **Resources Consulted:** List any web resources, peers, or other references you used, and summarize what you learned from them.
7. **Final Version of the Code [must have]:** Present your final working version of the code.
8. **Important: starting from Homework 4, you should also submit the Python file for the final version of your code.**

Note:

1. The components marked "[must have]" are the minimal requirements to get a pass for the process journal. If any of the must haves are missing, you will receive a "Not yet".
2. If you did any of the steps of 4, 5, and 6, you **must include them in your process journal**. It is an academic integrity violation if you are doing something but not reporting!
3. You don't need to follow this order strictly. You may organize your journal by date or merge sections as long as each required component is clearly identifiable. Notice how, in the example below, "resources consulted" are included in every step of the process.

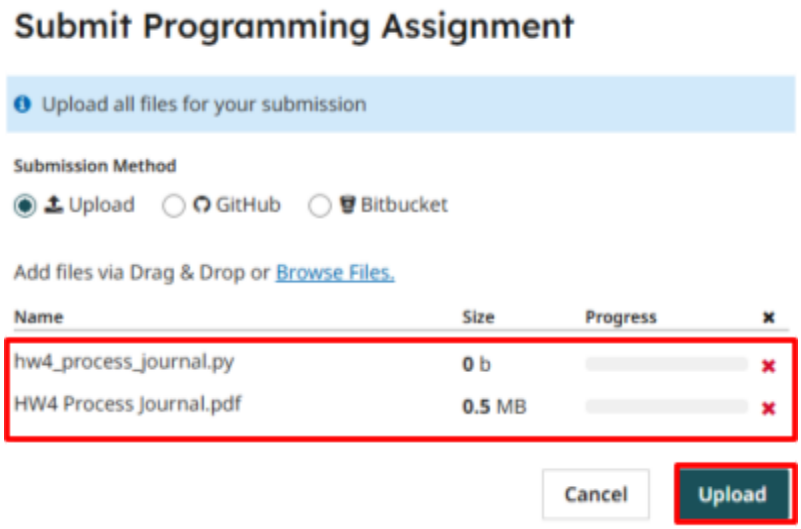
# Submission

**Submission Instructions for Process Journal - including Python file**

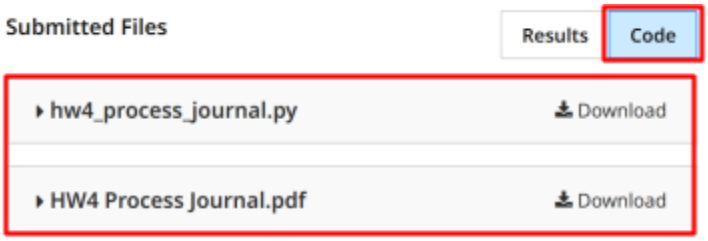
Starting from **homework 4 (not applicable to homework 3 resubmission, applicable to future resubmissions)**, you must submit both the **Python file (.py file)** containing your final implementation, and the **pdf of your process journal** to this question.

- You don't need to assign pages to the pdf (you will not see the option).
- For submission, using a laptop to submit is highly recommended.
- For submission, typing the process journal is highly recommended, as you can easily save it as a pdf and include screenshots of your progress.

Your submission page should look like this (the file names can be different):




You are done! Once you submit, you might see a message "This assignment does not have an autograder configured" under "Results". This is expected. You can preview the submission under the "code" tab to make sure you see both the python file and the pdf.



# Sample Task

Write a function in Python named `compute_final_letter_grade()` to compute the final letter grade for CSE 8A-6R as per the following grading policy stated in our syllabus. Make sure to test your function to ensure that it works as expected.

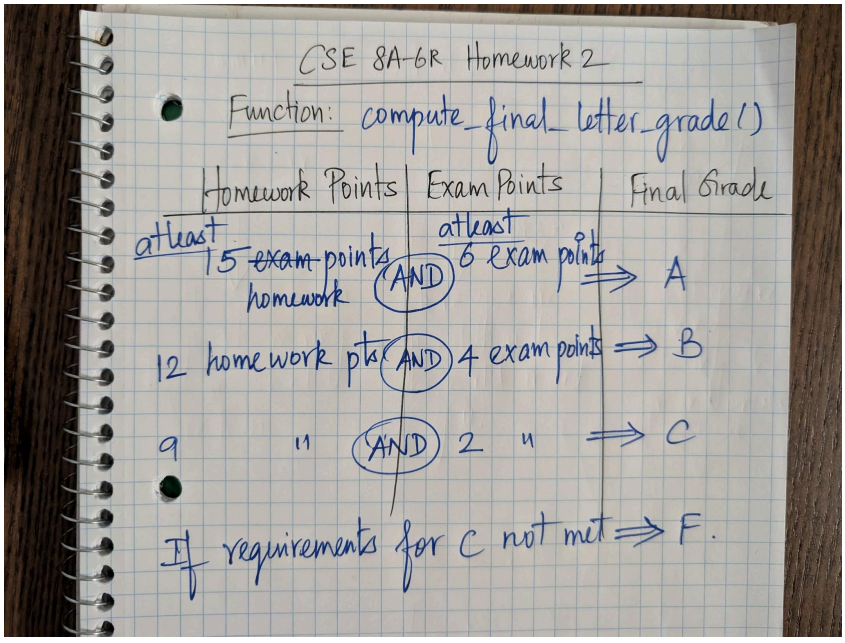
*Policy from CSE 8A-6R Syllabus (https://cse8a.site/syllabus)*
Your base grade is determined by meeting the minimum point thresholds for **both** homework and exams. There are 6 homework assignments (18 points max) and 2 exams (6 points max).

- **To earn an A:** You must have at least **15 total homework points** AND **6 total exam points**.
- **To earn a B:** You must have at least **12 total homework points** AND **4 total exam points**.
- **To earn a C:** You must have at least **9 total homework points** AND **2 total exam points**.
- **F:** Awarded if you do not meet the requirements for a C.

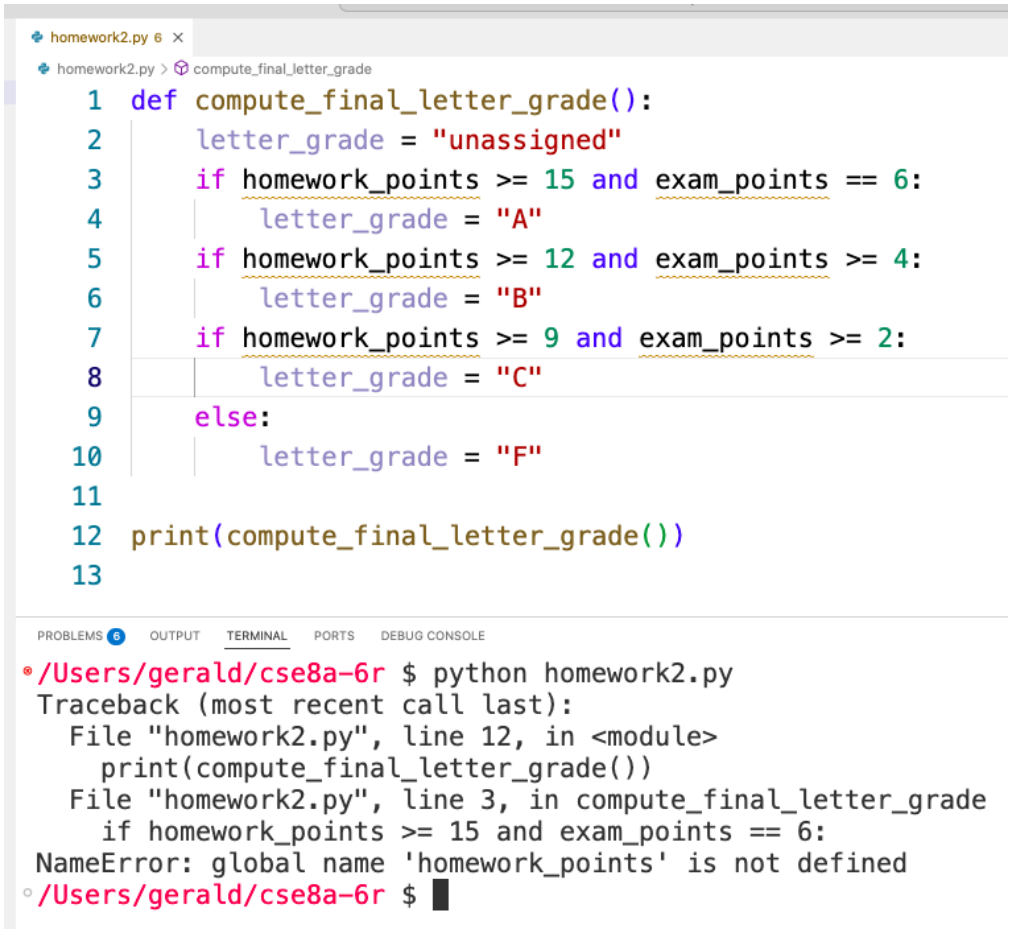
---

# Sample Process Journal

**Step 1: Understanding the problem**
I read the code writing task and I discussed it with my study group partner (Hailey) and we came up with the following rough sketch of what our function should do.

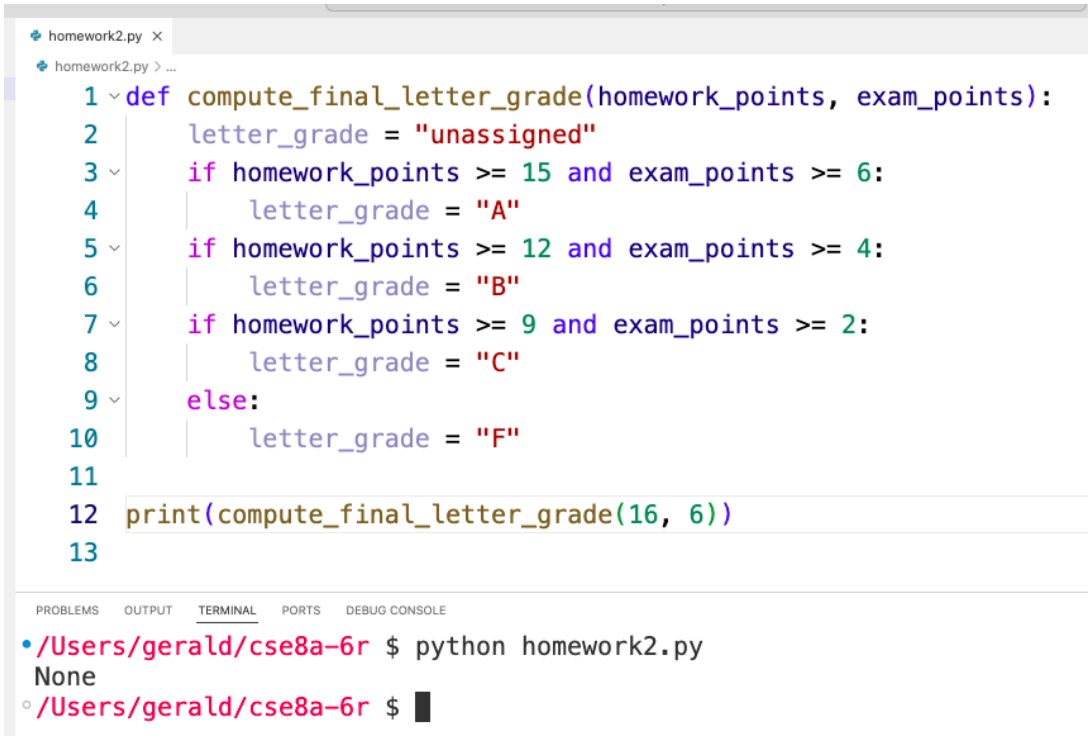


**Step 2: Writing the first version of my python function**
After understanding the problem, I started writing a python function in VS Code. I created a file named *homework2.py* on my computer and wrote the first version of my function. I got the following error.




**Step 3: Understanding and fixing the error message**
I went to our TA Andrew's office hours to understand why I am getting this error and the TA explained so beautifully that `homework_points` and `exam_points` should be passed in as parameters to the function `compute_final_letter_grade()` so that it understands what those variables are. After attending Andrew's office hours which was awesome btw, I fixed my code by adding two parameters: `homework_points` and `exam_points.` I also learned that I should pass in two arguments when I call the function to test it. I made these fixes and the previous error disappeared! Yay! Thank you Andrew!!!

**Step 4: Facing and fixing another error**
But my program still does not work as expected. I tried testing using homework_points = 16 (out of 18) and exam_points = 6 (out of 6). I expected the letter_grade to be "A" but I got None as shown below.




My study group partner, the amazing Hailey, and I reviewed our lecture 2 notes on functions and we seeked the help of our study group leader Sierra. Sierra helped us understand that we forgot to **return** the letter_grade at the end of our function. Sierra was so kind and patient with us throughout our study group. Thank you Sierra! :)

Although our function started returning letter grades, it returned the wrong letter grades! :(

For example, we tested our function using the following test case:
`print(compute_final_letter_grade(16, 6))`

We expected the letter grade to be “A” but we got a “C”! But why? :(

**Step 5: Learning the difference between separate ifs and if/elif/else statements using ChatGPT**
Over the weekend, Hailey and I worked together on our homework but we were completely stuck at this point. We turned to our friend, ChatGPT, to see if it can help us.

Our prompt to ChatGPT and its response is shown in this link below. We used the “Share” button in ChatGPT to generate this link.

Help from ChatGPT: https://chatgpt.com/share/68e34e4e-685c-8003-801b-e30fece928de

**Step 6: Writing and testing the final version of the code**
After learning the crucial difference between separate ifs and if/elifs from ChatGPT, we came up with the final version of our code and we tested with 4 different test cases (to test the 4 different grades, A, B, C, and F). The final version of our code and the test cases are shown below.

homework2.py

```
def compute_final_letter_grade(homework_points, exam_points):
    letter_grade = "unassigned"
    if homework_points >= 15 and exam_points >= 6:
        letter_grade = "A"
    elif homework_points >= 12 and exam_points >= 4:
        letter_grade = "B"
    elif homework_points >= 9 and exam_points >= 2:
        letter_grade = "C"
    else:
        letter_grade = "F"
    return letter_grade

print(compute_final_letter_grade(16, 6)) # expected: "A"
print(compute_final_letter_grade(13, 5)) # expected: "B"
print(compute_final_letter_grade(10, 2)) # expected: "C"
print(compute_final_letter_grade(15, 1)) # expected: "F"
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
/Users/gerald/cse8a-6r $ python homework2.py
A
B
C
F
/Users/gerald/cse8a-6r $
```

**Step 7: Feeling Happy!**
My study group partner Hailey and I felt so happy about completing this homework assignment! We learned so much in this process. We feel so accomplished right now!!! We would like to thank our TA Andrew, our study group leader Sierra, and ChatGPT for all their help in making us learn functions and conditionals. We can't wait to show what we have learned in our midterm exam!

The final version of course is pasted below.

Filename: **homework2.py**

```
def compute_final_letter_grade(homework_points, exam_points):
    letter_grade = "unassigned"
    if homework_points >= 15 and exam_points >= 6:
        letter_grade = "A"
    elif homework_points >= 12 and exam_points >= 4:
        letter_grade = "B"
    elif homework_points >= 9 and exam_points >= 2:
        letter_grade = "C"
    else:
        letter_grade = "F"
    return letter_grade

print(compute_final_letter_grade(16, 6)) # expected: "A"
print(compute_final_letter_grade(13, 5)) # expected: "B"
print(compute_final_letter_grade(10, 2)) # expected: "C"
print(compute_final_letter_grade(15, 1)) # expected: "F"
```