**CS 5001 : Final Synthesis - Code Based**

**Description**

For your midterm, I wanted you to prove to me an understanding of theory. For your final, I want to see an understanding of practice. Instead of a final, you’ll be creating a lab manual of sorts to accompany this course.

I’d like you to be creative and show me your mastery of the course topic. Each assignment should be 100% original, should demonstrate you understand the topic, and provide a neat, complete, and accurate key (with lots of comments). Although you are welcome to look up other assignments for inspiration. Do not just go look up assignments in Java and restate them. This should be your own creation. If I find work that is not original, this will result in 0 credit.

Each assignment should not be a repeat or2 a restatement of the lab you completed. I wrote the lab assignments. I want to see you use the topic in a different way.

You may cover multiple topics in a lab if it makes sense to do so. If you do so, it is advisable to include a concept map. A concept map is a simple document that lists the concepts and where they can be found within your submission. This ensures that during grading I can find and assess your information. For example:

|  |  |
| --- | --- |
| 1. Recursion | I demonstrated recursion in lab assignment 1: Recipes and Me in the function load\_recipe() found on line 13.  The function uses recursion instead of looping to make sure all ingredients are added. |
| 1. Error Handling | Error handling was demonstrated in lab assignment 1: Recipes and Me in the load\_file() function found on line 102. You’ll notice that I throw the appropriate exception here when the user enters an incorrect file name.  I further demonstrate it in all other functions making sure the user sends the date that’s expected. Those functions are :  process\_data(...) on line 45  get\_indgrediants(...) on line 50  notify\_user(...) on line 67 |
| … | … |
|  |  |

If you have any doubts, come see me during office hours and we’ll take a look at your attempt.

Format:

Topic Introduction:

An introductory section describing the concept the lab is to address along with a brief explanation of that topic, but not as much as you would include in the mid-way synthesis.

Assignment Goals:

A bulleted list of learning objectives

Assignment Description:

A complete description of the assignment including a numbered list of steps, and any starting code you want to provide.

Key:

The assignment as you would like to have seen it completed.

Note: I’m not asking for a rubric, but feel free to include one.

**Minimum Topics to cover:**

1. Recursion
2. Error Handling in Python
3. Dictionaries and Sets
4. Classes (Basic)
5. Stacks and Queues
6. Efficiency (Very Basic)
7. First half of the semester topic - your choice
8. First half of the semester topic - your choice

**Scoring :**

Each section will be considered individually based on completeness, accuracy, and quality. An Additional 10 points will also be considered for overall quality and design in a separate category.

Introduction and goals : 3 Points

Was the topic explained well?

Are the goals complete and do they match the section?

Problem/Key : 7 Points

Was the problem challenging enough?

Did the overall assignment show understanding of your understanding of the topic?

Additional adjustments :

Section quality : Points in a separate category, but additional points might be deducted per module if all the entire report points are deducted.

Was the grammar correct?

Was the overall structure neat and organized?

Was the code correct and written well?

Originality : up to 100% deduction

Was the section original or just restated from an existing assignment?

Was it a direct copy from another source?

**Final Submission**

Your final submission should also include a basic cover page(title, course, name, date, etc), an index for all of your chapters, and your content. File format needs to be pdf or html.

Tips:

* Have fun with it. Create a theme and try to stay on that theme.
* Don’t wait until it is due. Create your content as we go through the semester.

Your submission does not have to look exactly like this, but here’s an example to better explain my expectations:

[Example Synthesis Chapter - Project](https://docs.google.com/document/d/1xVfI0q13QgUYDdQd0d43ZNqO-wLB8NBVUIkcanwuE8o/edit?usp=sharing)

**Grading Statement**

Submit a grading statement that uses the rubric correctly and assign yourself a grade. There is no guarantee you’ll get this grade, but I’d like to see what you think you deserve.

**Addendum: Project Option**

If you’d like to submit your final as a project instead, we can discuss this. You’ll need to write a proposal to make sure the complexity of your project makes up for the reduced amount of narrative. A project also requires a concept map instead of making it optional, and you’ll need to make sure to include a well written grading statement.

**Rubric**

|  |  |
| --- | --- |
| Concept 1 : Recursion | 10 Points |
| Concept 2 : Error Handling in Python | 10 Points |
| Concept 3 : Dictionaries and Sets | 10 Points |
| Concept 4 : Classes (Basic) | 10 Points |
| Concept 5 : Stacks and Queues | 10 Points |
| Concept 6 : Efficiency (Very Basic) | 10 Points |
| Concept 7 : First half of the semester topic - your choice | 10 Points |
| Concept 8 : First half of the semester topic - your choice | 10 Points |
| Overall Quality | 10 Points |
| **Total Possible Points (without a bonus)** | **90/100Points** |
| Possible adjustments |  |
| Bonus Points for creativity and going above and beyond | +10 Points |
| Additional chapter not requested (repeatable) | +10 Points |
| Extra complexity or outstanding quality | +10 Points |
| Work originality | -100 Points |
| Code quality | -50 Points |

All work must be 100% original and I reserve the right to ask you to explain any part of your submission. You will automatically receive 0 credit on the entire assignment if it is copied from an online source or another student. The only assistance for this project should be from your TA or your professor.