

- All the responses should be in your Github before ***the end of the class***.
- For questions/ clarifications, ask Instructor biplav.s@sc.edu and TAs vishalp@email.sc.edu, kausik@email.sc.edu.

Total points = (10 + 90) [+ 20 (optional)] = 100 points [120 max with optional].

Student Name:

Obtained =

The quiz tests your understanding of different concepts discussed in the class.

Create a sub-folder in your GitHub called Quiz-2. Have subfolders: **data, code, docs**.
Any text you write which is not data or code will be in file: **./docs/answer-comments.md** file

Q1: About Comparing energy consumption: [6 + 4 = 10 points]

Our aim will be to understand and compare energy consumption for AI alternatives for different tasks using an online tool:

<https://symbio6.nl/en/apps/ai-vs-simple-energy.html>

Instructions:

- Explore any three different tool settings with AI (LLM based) and classical method. Record the two tool alternatives in each setting and the energy difference. **[6 points]**
- Question: what is the average energy difference across the three settings you checked? Which approach (LLM or classical) is higher on an average and by how much? . **[4 points]**

Q2: Using LLMs for a task: [6 + 4 = 90 points]

Our aim will be to convert food recipes from text to a semi-structured format in .json using ChatGPT that can be, hopefully, executed by a robotic chef.

For reference, we will use data and code in: <https://github.com/biplav-s/course-ai-f25/tree/main/sample-code/future/llm-recipe-conversion>

- Food site: <https://www.instructables.com/Egg-Drop-Chicken-Noodle-Soup/>
- Outputs we want,
 - AI test case, in the **template format**: <https://github.com/biplav-s/book-trustworthy-chatbot/blob/main/ai-testcases/testcase-template.md>
 - **Sample .json (R3) output**: https://github.com/biplav-s/course-ai-f25/blob/main/sample-code/future/llm-recipe-conversion/data/example_r3_egg-drop-chicken-noodle-soup.json
- Sample code for json handling: <https://github.com/biplav-s/course-ai-f25/blob/main/sample-code/future/llm-recipe-conversion/r3-procesing-demo.ipynb>
- **Gaico code** for analyzing LLM output is here: <https://github.com/biplav-s/course-ai-f25/blob/main/sample-code/class2-gaico-usage/LLMsOutputAnalysisWithGaico.ipynb>

Instructions:

- a) Select the recipe of any two foods from the web which is not the example above. Suggestions: select from instructables.com or allrecipes.com
- b) You will use GPT/ChatGPT, or any other LLM/Chatbot of your choice, to convert these two recipes into R3 format. Along the way, you will explore different conversion strategies
 - a. Write the AI testcase using the template for this exercise **[10 points]**
 - b. Review R3's metadata from the sample code. You will use them for your foods' recipe outputs.
 - c. Use LLM/Chatbot to convert recipe text to R3 format
 - i. Record text of original recipe in a file: `./data/original_recipe1.txt`, `./data/original_recipe2.txt`. Feel free to clean content manually and report what you did, e.g., cleaning, in `./docs/answer-comments.md` file. **[5 + 5 = 10 points]**
 - ii. Note on prompting: A good robotic recipe will have instructions that are simple English sentences with one single task at a time. See sample recipe (egg-drop) in R3.
 - iii. Prompt-full approach: give at least **three** different prompts to create **full** R3 file. We will refer to them as (PF1, PF2, PF3, ...). **[30 points]**
 - iv. Prompt-partial approach: give at least **two** prompts to create **partial** R3 file contents and then combine them together using a Python code. For example, extract ingredients and instructions separately, and then combine the result in Python code that stitches the output into the R3 json. We will refer to them as (PP-1, PP-2, ...) **[10 points]**
 - v. For each food, run the approach and show the prompts and the results (.json). So, the total conversions you will have is at least, 4 (prompts: PF1, PF2, PF3, PP-1+PP-2) x 2 (food) = 8 .json files.
- c) Evaluate the results (.json file) for the LLM output using the following goodness score rubric. You will write a python function that will take a .json file as input and returns the goodness score out of 100. **[10 points]**
 - a. Correct .json format of output: 50 score; 0 for invalid
 - b. Number of following metadata from below in .json: 10 scores to each (50 maximum)
 - i. `recipe_name`,
 - ii. `data_provenance`,
 - iii. `macronutrients`,
 - iv. `ingredients`,
 - v. `instructions`,
 - c. Make sure your AI testcase reflects this rubric

Now answer the questions in file `./docs/answer-comments.md`

Q1. Which conversion approach does well (full or partial) for the foods whose recipe you chose ? **[10 points]**

Q2. What is the highest goodness score you obtained for each of the two chosen foods? **[10 points]**

Q3. [Optional – extra credits] Can you try any other prompting approach and show its goodness score? **[10 points]**

Q4. [Optional – extra credits] You can use GAIKO to compare across LLM outputs for Prompt-full approaches (PF1, PF2, PF3). **[10 points]**