# *Instructor Prep-sheet:*

# **Leveraging Generative AI for Software Engineers Instructor Demo**

## **TL;DR:**

This prep-sheet is designed to give you an idea of how to conduct your live ChatGPT demonstration. A few things to remember:

1. Do a dry run of your demo before you are live and in front of students
2. Be prepared for slight variance in ChatGPT’s outputs each time you do the demo
3. Highlight the impressive capabilities of ChatGPT, along with its limitations!

* *If you end up with a demo that was particularly interesting, please share the link* [*here*](https://docs.google.com/spreadsheets/d/1eWa3s_gkPYEfLKoBXG-83LErmMJ60elgCjuePFlPo4Y/edit#gid=0) *along with a short description!*

## **Sample Materials:**

* [Sample ChatGPT Thread with Analysis](https://docs.google.com/document/d/1gxnqi5Z7YWfYDY71qt8bWEsXf3q1gJlSvKRUbieh-hA/edit?usp=drive_link)

## **Preparation:**

* Review the [sample ChatGPT thread](https://docs.google.com/document/d/1gxnqi5Z7YWfYDY71qt8bWEsXf3q1gJlSvKRUbieh-hA/edit?usp=drive_link) in **detail**
  + This is an example of how you can structure the demo.
  + It is recommended to start the demo with the same prompt every time.
  + At the end of the day this is your demo - take it in whatever direction you find most impactful based on your students and teaching style
* Login to [ChatGPT](https://chat.openai.com/) (you will just use the free version for this demo)
* Do a “dry-run” of your demo before you go live in class!

## **Instructor Demo Outline:**

* *This is a simplified outline of* [*this sample ChatGPT thread*](https://docs.google.com/document/d/1gxnqi5Z7YWfYDY71qt8bWEsXf3q1gJlSvKRUbieh-hA/edit?usp=drive_link) *in which a user asks ChatGPT to help them complete a software engineering task.*
* *You can base your demonstration off of this sample, or come up with your own.*
* *If you decide to come up with your own, be sure to highlight the impressive capabilities of the tool, as well as the limitations.*
* *It is also important to make clear the need for the user to be a skilled software engineer in order to get the most out of the tool - ChatGPT is not a replacement for a good software engineer!*

1. **Query:**
   1. Instructor wants to use ChatGPT for software engineering tasks, specifically for developing a calculator app in Python.
2. **Response:**
   1. ChatGPT generates a formal requirements elicitation for the calculator software.
3. **Analysis of the response:**
   1. The requirements elicitation looks clear and concise.
   2. The instructor notes that there isn’t much they would change - ChatGPT did a great job here.
4. **Next Query to ChatGPT:**
   1. Instructor asks for an object-oriented design for the calculator program.
5. **Response:**
   1. ChatGPT provides an object-oriented design for the calculator program and includes the implementation code.
6. **Analysis of the response:**
   1. The design separates responsibilities between the Calculator and CalculatorApp classes.
   2. However, there is a gap between the formal requirements and the implementation.
7. **Last Query to ChatGPT:**
   1. Instructor requests test cases for the app.
8. **Response:**
   1. ChatGPT generates test cases to verify the functionality of the calculator app.
9. **Analysis of the response:**
   1. The test cases cover different arithmetic operations and edge cases.
10. **Final remark:**
    1. The inconsistency between the requirements and the implementation highlights the limitations of ChatGPT in understanding real-life software development practices.
    2. ChatGPT's responses are based on learned examples, which may not encompass all necessary aspects of a real-world application.