Special POGIL Activity

Programming Debate: President's Word Game

Objective

In this activity, your group will:

1. **Ask different AI models** to generate the complete code for the **President's Word Game** program (Lab 4)
2. **Analyze and compare** the AI-generated solutions to understand their logic and efficiency.
3. **Evaluate the AI's approach** and see if you can improve the generated code.
4. **Modify and refine AI prompts** to get better, clearer, or more optimized responses.
5. **Present your findings** and explain how AI can be a tool for learning, not just an answer provider.

Role	Name	Email
Facilitator		
Spokesperson		
Quality Control		
Process Analyst		

Please write the names and email addresses of the group members in the table above and share the document with professor Aydin (aydinn@kenyon.edu) and the TA lin7@kenyon.edu Use a different color to enter your answers.

Step 1: Understanding the Problem

Project Overview

You will develop a **word guessing game** based on President John F. Kennedy's **1961 Inaugural Address**. The game will:

- Show the player **two words** from the speech.
- Ask the player to guess **which word appears more often** in the speech.
- Keep track of the player's **score** based on correct guesses.
- Exclude **common words (stopwords)** like *"the"*, *"and"*, *"to"*, etc.

Key Programming Concepts

- Reading from files (speech text & stopwords)
- Parsing and processing words (cleaning punctuation, lowercasing)
- Searching and comparing word frequencies

- Using randomization to select words
- Using functions (breaking code into smaller tasks)
- Handling user input and errors

Step 2: Generating and Analyzing AI Solutions

Each group member will **ask a different AI model** for the complete solution to the problem. The AI models you can use include (you may use others as well):

- **Claude** → https://claude.ai/
- **Google Gemini** → https://gemini.google.com/
- **Microsoft Copilot** → https://copilot.microsoft.com/
- **Meta AI** → https://ai.meta.com/meta-ai/
- **ChatGPT** → https://chat.openai.com/

Each group member should **individually** ask their assigned AI to generate a full solution.

Step 2.1: Asking AI for a Complete Solution

Each group member will **ask their assigned AI model** to generate the full program. Use a well-structured prompt, such as:

"Write a Python program that plays a word game where the user guesses which of two words appears more frequently in a given speech. The program should load a stopwords list from a file (provided) and exclude these words when counting word frequency. Use functions for organization."

After receiving a response, **analyze the AI-generated code** by filling out the following:

Member 1 - AI Model Used: _____

1. **What prompt did you use?**

   ---

2. **How did the AI organize the program? List the main steps or functions it created to solve the problem.**

   ---

3. **What part of the AI-generated code worked correctly without modification? Identify any functions or program segments that were immediately useful and explain why.**

   ---

4. **Run the AI-generated code. Does it work as expected? If there are errors, what are they?**

5. **Were there any unnecessary or overly complex parts in the AI's solution that you do not understand? Did it include concepts or features of the language not covered in this course?**

Member 2 - AI Model Used: _____

1. **What prompt did you use?**

---

2. **How did the AI organize the program? List the main steps or functions it created to solve the problem.**

---

3. **What part of the AI-generated code worked correctly without modification? Identify any functions or program segments that were immediately useful and explain why.**

---

4. **Run the AI-generated code. Does it work as expected? If there are errors, what are they?**

---

5. **Were there any unnecessary or overly complex parts in the AI's solution that you do not understand? Did it include concepts or features of the language not covered in this course?**

Member 3 - AI Model Used: _____

1. **What prompt did you use?**

---

2. **How did the AI organize the program? List the main steps or functions it created to solve the problem.**

---

3. **What part of the AI-generated code worked correctly without modification? Identify any functions or program segments that were immediately useful and explain why.**

---

4. **Run the AI-generated code. Does it work as expected? If there are errors, what are they?**

---

5. **Were there any unnecessary or overly complex parts in the AI's solution that you do not understand? Did it include concepts or features of the language not covered in this course?**

Member 4 - AI Model Used: _____

1. **What prompt did you use?**

---

2. **How did the AI organize the program? List the main steps or functions it created to solve the problem.**

---

3. **What part of the AI-generated code worked correctly without modification? Identify any functions or program segments that were immediately useful and explain why.**

---

4. **Run the AI-generated code. Does it work as expected? If there are errors, what are they?**

---

5. **Were there any unnecessary or overly complex parts in the AI's solution that you do not understand? Did it include concepts or features of the language not covered in this course?**

Step 2.2: Improving the AI's Response

Now, refine your AI prompt to get a **better response**. Think about:

- Making your request **clearer** (e.g., specifying how the files should be processed).
- Asking for **more structured functions** (e.g., ensuring modular programming).
- Ensuring the code **runs efficiently** (e.g., avoiding unnecessary loops).

1. **What improvements did you make to your prompt?**

---

2. **Ask the AI again with the improved prompt. How did the new response compare to the first one?**

---

3. **Did the AI now generate a better solution? What changed?**

---

4. **What would be an ideal way to phrase the prompt if you had to ask again?**

Step 3: Observations and Refinements

After comparing all the AI-generated solutions, discuss:

1. **What were the biggest differences between the AI models' responses?**

---

2. **Which AI provided the clearest and most structured code? Why?**

---

3. **What mistakes or inefficiencies did you find in the AI-generated code?**

---

4. **How did refining your prompt help improve the AI's response?**

---

5. **What are your observations based on comparing the solutions provided by AI tools with your own solution to Lab 4?**

Key Takeaway: AI is a Learning Tool, Not a Shortcut

This activity was not about finding **the best AI tool** but about understanding how to:

- Write effective prompts to get useful AI responses.
- Analyze AI-generated code and understand its logic.
- Identify weaknesses and improve AI-generated solutions.
- Use AI as a tool to enhance your programming knowledge and skills, not as a way to circumvent the learning process.