In this activity, you'll leverage a generative AI tool to help you debug a Python script. This activity is optional and will not affect your completion of the course.

Scenario

You're an IT professional at a software startup. Your manager wants to streamline communication and be able to send targeted messages to relevant teams. To automate thi you want to write a Python script that can categorize each employee by their department. This script will help you make sure all your communications go to the right employe to use a generative AI tool to write the Python script.

Activity Tools

In this activity, you will use a browser-based generative AI tool, such as Gemini, ChatGPT, or Microsoft Copilot. Instructions in this activity will refer to Gemini, but you can us generative AI tool of your choice.

Step 1: Access the template and supporting materials

The following template and supporting materials will help you complete this activity. Keep them open as you proceed to the next steps.

To use the template and supporting materials for this course item, click the following links and select *Use Template*.

Link to activity template: Prompt an AI tool

Link to supporting materials: Prompt engineering best practices

OR

If you don’t have a Google Account, you can download the template and supporting materials directly from the following attachments.

Prompt engineering best practices
DOCX File

Prompt an AI tool
DOCX File

Step 2: Access Gemini

To access Gemini:

Go to gemini.google.com.
Sign in to your personal Google account.

Refer to the resource about how to Create a Google Account, if you don't already have one. For further assistance signing into Gemini, please refer to Gemini Apps Help

Note: Before you use Gemini, review the following information:

You must be over 18 years old to use Gemini.

Review the Gemini Apps Privacy Notice.

Please don’t enter private or confidential information in your Gemini conversations or any data you wouldn’t want Google to use to improve its products, services, and machir technologies.

Gemini is not available in certain countries and languages. For more details, refer to documentation about Where you can use the Gemini web app.

Feedback from a wide range of experts and users helps Gemini improve every day. So when you try Gemini, you can provide feedback using the thumbs up or thumbs down the option to further explain in a comment.

Step 3: Prompt Gemini to debug a Python script

A prompt is input that provides instructions to an AI tool about how to generate output. You can prompt Gemini using text or speech and can phrase your prompts in a variety prompt should provide clear and specific instructions that will guide the tool towards generating a targeted response.
To get started, type or speak this prompt or something similar into Gemini:

Write a Python script that categorizes employees based on the department they work in.

Note: If you are reviewing the provided code using a code editor or IDE and the output is not what you expected, prompt Gemini with the error message or details about the c Gemini will provide an updated code snippet.
Then type or copy and paste your prompt into the Prompt section of the Prompt an AI tool template.

Step 4: Review the output

Copy and paste Gemini’s output into the Output section of the template. Sometimes, Gemini may offer multiple responses. You can choose the one that most closely meets y

Then, carefully review the output to ensure it is factually correct, specific, and detailed enough. As you review the output, consider the following questions:

Is the output accurate?

Is the output unbiased?

Does the output include the information I need?

Is the output relevant to my project or task?

Is the output consistent if I use the same prompt multiple times?

Reflect on the ways in which the output meets the goals of your categorization script. For example, does the script accurately categorize employees, and does the script form in a way that’s easy for you to interpret?
Then, in the Notes section of the template, list at least two ways that the output could better meet your needs.

Step 5: Provide specific follow-up requests

Working to refine your prompts and provide more context can help the generative AI tool produce clearer, more specific, and more useful information. Think of this process as conversation with someone, where they tell you something and then you respond by asking for more information or context. Engaging in a back-and-forth dialogue with gene can lead to more personalized and effective output. And issuing follow-up requests can help produce content that better aligns with your needs.
Review the guidelines in the Prompt engineering best practices document and refer to the notes you listed in Step 4 about how Gemini’s output could better meet your nee review the following examples, which demonstrate how a prompt could be iterated upon to produce a more useful output, and implement at least two of these best practices up requests to your original prompt:
Add a verb. It's helpful to include a verb in your prompt to produce useful output for your intended task. The provided prompt already includes a verb, but using a different ver or “construct” would give you a different output.
For example: *Write a Python script that categorizes employees based on the department they work in.*

Add context. Supply more information about the task to help the tool provide better results.

For example: *Write a Python script that categorizes employees based on the department they work in so that emails can be sent to employees based on their departments.*

Add a specific goal. Include the final goal or specify the outcome that you need. For instance, you could specify that the script should only categorize employees who have w startup for longer than 90 days.

For example: *Write a Python script that categorizes employees based on the department they work in, but only categorize employees who have worked at the company for lo days.*

Reflect on whether you should format your results. Give specific instructions for how to return the output, such as in a table with headers.

For example: *Write a Python script that categorizes employees based on the department they work in and formats the output in a table with headers.*

Supply examples. Demonstrate the kind of output you're hoping for.

For example, you could provide the AI tool with a small table with a column for employee names and departments so the tool has a reference for what the output should look

Reflect on the ways in which the iterated output meets your goals. The objective is not to get the perfect output, but to get enough information for you to work with. Be sure to your output before using it.

In the Observations section of the template, list at least two ways that you iterated on your prompt with follow-up requests and describe how those iterations made the output goals more closely.