TABLE OF CONTENTS

1. Configuring Your Claude Workspace for Knowledge Work
2. Prompting, Grounding, and Tool Selection for Business Tasks
3. Personalizing Claude with Preferences and Styles
4. Organizing Work with Projects and Artifacts
5. Designing AI Delegation
6. Detecting and Mitigating AI Failure Modes
7. Build a Cowork Project Workspace
8. Create a Scheduled Task
9. Standardize Work with Skills and Plugins

WA3924 Lab Guide

Practical AI for Executives

Configuring Your Claude Workspace for Knowledge Work

preferences that align Claude with your role and communication norms

- ✓ Manage Claude's memory and chat search to preserve useful professional context and exclude sensitive or one-off conversations
- ✓ Compare responses from Opus 4.7, Sonnet 4.6, and Haiku 4.5 on the same task to build intuition for model selection

1.1. Introduction

Estimated time: 25 minutes

In this lab, you will set up Claude Chat as a personalized environment for knowledge work rather than treating it as a generic chat window. You will move through the interface intentionally, configure the settings that shape how Claude interprets you, take direct control of what Claude remembers across sessions, and finish by running the same task across all three current models so you can feel — not just read about — the difference in speed, depth, and tone. The goal is to make Claude a reliable extension of your working context before you attempt any serious business task.

> ⓘ **Note**
>
> This lab and all labs in this course are designed for the web version of Claude Chat. In addition, it assumes you are using a paid tier account, which provides access to all three models and the full suite of capabilities. If you are using a free tier account, you can still complete the lab but you may not have access to all three models or certain features. If that's the case, just skip the activities that aren't available to you and focus on understanding the concepts. Furthermore, some labs include the use of external files. Confirm with your instructor where to find these files.

1.2. Lab Overview

In this lab, you will complete the following activities:

1. Tour the four areas of the Claude Chat interface and identify the purpose of each
2. View and edit your memory, and directly instruct Claude to remember several facts you want retained across future sessions
3. Conduct one conversation in incognito mode, then use chat search to retrieve verbatim content from a prior standalone conversation

1.3. Tour the Four Areas of the Claude Chat Interface

Open Claude Chat at claude.ai in your browser and sign in. Take a few minutes to locate each of the four areas and complete the small task associated with it.

- **Area 1 — Left Sidebar (Navigation).** Identify New Chat, Recents, Projects, and Artifacts.
- **Area 2 — Bottom Left (Personal Settings & Configuration).** Click your initials in the lower-left corner to open the personal settings menu. This is where Claude learns how you work — it contains memory, styles, connectors, and account settings. You will work directly with memory in the next activity.
- **Area 3 — Center Panel (Conversation).** Locate the input bar at the bottom of the chat. Note the model selector below it, the "+" button for file attachments and tool toggles, and the ghost icon in the upper right that starts an incognito session.
- **Area 4 — Right Panel (Artifacts & Project Details).** Note that this area is contextual — it only opens when you are working with an artifact or inside a Project. You will not see it during a normal chat. We will work with this area in the next module.

1.4. View and Edit Your Memory

Memory is Claude's synthesized model of who you are — your role, professional context, and working preferences. It is separate from your chat history and is auto-injected at the start of every new standalone conversation.

1.4.1. Confirm that Claude currently remembers nothing about you

1. Navigate to **Settings > Capabilities**
2. In the **Memory** section make sure that "Search and reference chats" and "Generate memory from chat history" are both toggled "On" (the button turns blue).
3. You should see an option to "View and manage memory" appear if you need to toggle "On" the above options.
    - Alternatively, Claude might not display "View and manage memory" if it doesn't have any memories about you yet. Don't worry, you will create some in a minute.
    - Remember: if you are new to Claude (i.e. this is a new account for you), the memory will be empty or sparse — that is expected.

1. Start a new chat and enter a prompt that establishes standing context.
    - Here's an example memory instruction you can adapt:

> Please remember that I work in [your industry] as a [your role], and that when I ask for analysis I want a bottom-line-up-front summary followed by supporting detail.

    - Claude will confirm what it has stored in memory. (Watch for this message in the chat.)

1.4.3. Return to Settings > Capabilities > "View and edit memory"

1. Verify the new entry appears.
2. Edit or remove anything that is incorrect, then reflect:
   What kinds of information do you NOT want Claude to remember, and how would you keep it out?

1.5. Use Incognito Mode and Chat Search

Some conversations should not feed Claude's memory or appear in your history — sensitive HR matters, confidential strategy work, or one-off lookups. Claude provides incognito mode for exactly these cases.

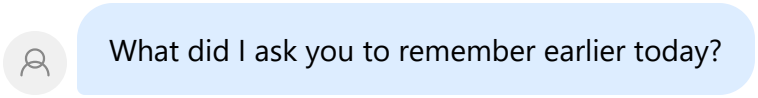
1. Locate the **ghost icon in the upper right** of the chat panel.
2. Click it to start an incognito session.
3. Have a brief conversation about something sensitive or one-off.
    - Here's an example of an incognito-appropriate prompt:

> Help me draft talking points for a difficult conversation with a team member about declining performance.

    - Notice that the incognito chat does not appear in your **Recents**.

1.5.1. Close the incognito session

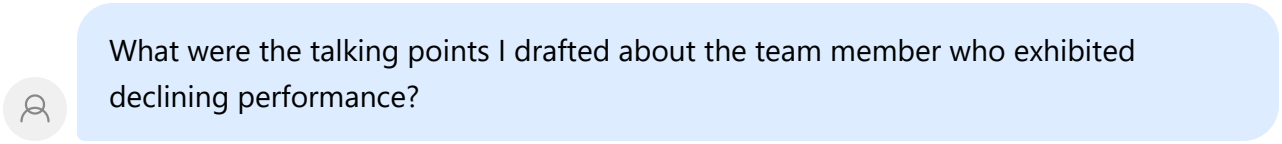
1. Return to a new standalone (non-incognito) chat.
2. Ask Claude to retrieve content from the memory chat you ran earlier in this lab.
    - Here's an example chat search prompt:



3. Watch the chat for a visible tool call indicating Claude is searching your chat history.
4. The result should include a citation linking back to the source conversation.

1.5.3. Try asking for verbatim content from the incognito conversation

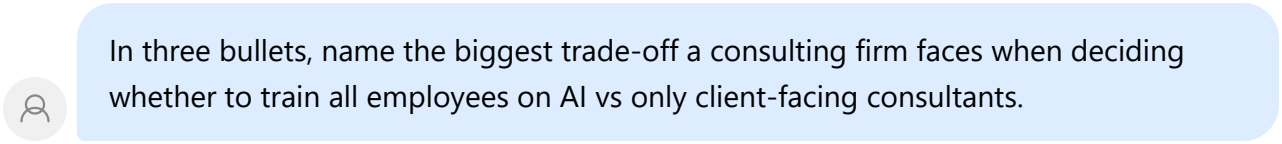
- Example prompt:



- Reflect:
    i. Did Claude retrieve verbatim content or a summary of what I told it to remember?
    ii. Could Claude find content from my incognito conversation? Why or why not?

1.6. Run the Same Prompt on Three Models

1. Now you will run the same prompt on each of the three current Claude models and compare the responses side by side.
2. Use this prompt, or substitute one from your own work:



1.6.1. Run the prompt three separate times — each in a new chat

> **Important**
>
> Do not switch models mid-conversation, as that may clear cached context and consume additional usage limits.

iii. Whether the trade-offs identified are surface-level or substantive
iv. Whether the response feels complete or feels rushed

2. Start a new chat. Set the model to **Claude Sonnet 4.6.** Submit the same prompt. Compare to Haiku:
   i. Did response time noticeably increase?
   ii. Did Sonnet identify different or deeper trade-offs than Haiku?
   iii. Did Sonnet add nuance, qualifications, or second-order considerations Haiku missed?
   iv. Does the tone feel more polished or executive-ready?

3. Start a new chat. Set the model to **Claude Opus 4.7.** Submit the same prompt. Compare to both prior runs:
   i. How much longer did Opus take vs. Haiku and Sonnet?
   ii. Did Opus surface trade-offs neither of the other models considered?
   iii. Is the reasoning visibly more structured or rigorous?
   iv. For **this specific prompt**, did the extra depth justify the extra latency?

1.6.2. Review your three outputs side by side and reflect

1. Which model produced the most usable response for this specific task?
2. Was the difference in depth proportional to the difference in latency?
3. For your daily work, which model would you default to? When would you escalate to Opus 4.7? When would you drop to Haiku 4.5?

1.7. Challenge Activity (If Time Permits)

1. Open one of the three model-comparison chats
2. Switch the model mid-conversation by selecting a different model from the model selector below the input bar.
3. Then send a follow-up prompt — for example, "*Tighten your previous answer to three bullet points.*"
4. Observe whether the response feels slower than usual, and notice if Claude acknowledges that the model has changed.
   - This demonstrates why a new chat is preferable when you genuinely need a different model.

1.8. Key Takeaways

every new conversation
- Memory and chat history are independent systems with independent controls in **Settings > Capabilities**
- Incognito mode is the right choice for sensitive or one-off conversations — it bypasses both memory and chat history
- Match the model to the task: Haiku 4.5 for speed, Sonnet 4.6 for daily work, Opus 4.7 when accuracy and depth matter most

Prompting, Grounding, and Tool Selection for Business Tasks

Lab 2

- ✓ Apply the C-L-E-A-R framework to transform a vague prompt into a precise, actionable instruction
- ✓ Ground Claude's responses in an uploaded source document to reduce generic output and improve factual accuracy
- ✓ Select and use the appropriate tool based on the business task — web search, adaptive thinking, or Research mode

2.1. Introduction

Estimated time: 40 minutes

In this lab, you will move from personalization into production — using Claude to generate real, defensible work product. You will start with a deliberately weak prompt and progressively sharpen it using the C-L-E-A-R framework, then add a source document so Claude works from your material rather than its general knowledge. You will then complete three short, targeted tasks — one using web search, one using adaptive thinking, and one using Research mode — so you leave the lab with first-hand experience of when each tool is the right choice.

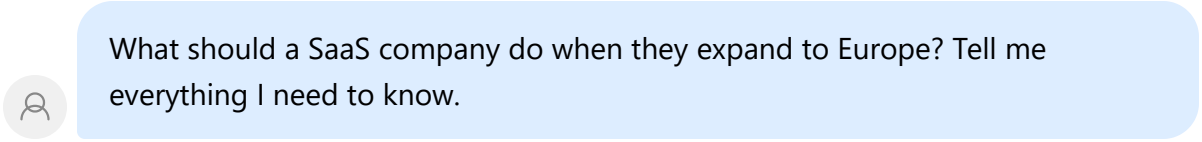
In this lab, you will complete the following activities:

1. Rewrite a provided vague business prompt by applying the Concise, Logical, and Explicit principles
2. Iterate on your rewritten prompt using the Adaptive and Reflective principles, revising based on Claude's initial response
3. Upload a source document and instruct Claude to answer using only that material
4. Complete a Web Search task: use Claude to find and cite current, factual information on a specified business topic
5. Complete an Adaptive Thinking task: use Claude to work through a multi-step analytical problem with no external sources
6. Complete a Research Mode task: commission a multi-source comparative report and review the chain-of-thought log afterward

2.3. Rewrite the Vague Prompt with C-L-E

2.3.1. Start a new chat in Claude Chat

1. Begin by submitting a deliberately weak prompt and observing the response.
    - Here's an example of a vague business prompt:



2.3.2. Submit the prompt and review the output

- The response will likely be long, generic, and unfocused.
- The prompt fails on three counts:
    - **Concise:** contains conversational filler ("when they", "Tell me everything I need to know")
    - **Logical:** mixes two different concepts — actions ("what to do") and information ("what to know")
    - **Explicit:** specifies no industry detail, no target countries, no audience, no format

2.3.3. Now rewrite the prompt by applying Concise, Logical, and Explicit

You are advising a U.S.-based HR compliance SaaS company entering the UK, Germany, and France. Identify the top three strategic decisions they must make in their first year of expansion.

2.3.4. Run the improved prompt in the same chat

1. Compare the two responses.
2. Reflect:
    i. Did the improved prompt produce a more focused, business-ready response?
    ii. Which of the three principles — Concise, Logical, or Explicit — had the biggest impact?
    iii. What is still missing from the response?

2.4. Iterate the Prompt with A-R

The Concise, Logical, and Explicit principles get you a usable first draft. Adaptive and Reflective drive what happens next: review Claude's response, identify what is missing or off-target, and revise the prompt accordingly.

1. Review the response Claude gave to your improved C-L-E prompt
2. Identify at least one of the following:
    i. A constraint you forgot to specify (length, format, audience, tone)
    ii. An aspect of the response that is too generic, off-target, or hedge-laden
    iii. A piece of structure that would make the output more decision-ready

3. Now revise the prompt to address what you found
    - Here's an example of a revised prompt using A and R:

You are advising a U.S.-based HR compliance SaaS company entering the UK, Germany, and France over three years. Identify the top three strategic decisions they must make in their first year, ranked by financial risk. Format as an executive memo, 250 words maximum. Lead each decision with the recommendation, then a one-sentence rationale. Audience: a CFO with M&A background. Avoid hedging language.

2.4.1. Run the revised prompt

ii. Did Claude follow your "avoid hedging" instruction, or did it slip back into generalities?
iii. Would another iteration improve it further? What would you change?

2.5. Ground Claude in a Source Document

Now you will add a real source document and explicitly restrict Claude to using only that material. This is the External Knowledge Restriction technique — it dramatically reduces hallucinations and forces Claude to ground its answers in your content rather than its general training data.

2.5.1. Start a new chat

1. Upload **market_entry_strategy.txt** using the "+" button or by dragging the file into the chat input.
    - This file is located in the lab files directory; confirm with your instructor where to find it.
    - It contains a sample market entry strategy for a fictional SaaS company called CompliSure, which is expanding into Europe.
2. Confirm the file is attached

2.5.2. Submit a prompt that explicitly tells Claude to use only the document

Here's an example of a knowledge-restricted prompt:

> Using ONLY the content of the uploaded document, answer the following: For each of the three target countries (UK, Germany, France), list the entry mode CompliSure plans to use and explain why that mode was chosen. If any information is not in the document, say so explicitly — do not infer or supplement from your general knowledge.

2.5.3. Review Claude's response

- Here's an example follow-up:



> Using only the document, what is CompliSure's current market share in the UK and who are their top three named competitors?

3. Review the response to see if Claude correctly identifies that this information is not in the document, or if it hallucinates an answer based on its training data.
   - Claude should acknowledge the information is not in the provided document, rather than fabricating an answer.
   - If Claude provides a specific answer without flagging uncertainty, revisit your prompt and try asking more directly.

4. Reflect:
   i. Did Claude correctly identify that the market share and named competitors are not in the document?
   ii. How does forcing Claude to use only one source change the tone and confidence of the response?
   iii. When in your real work would the "use ONLY this material" instruction be valuable?

2.6. Complete a Web Search Task

Web search is the right tool when you need current, factual information that may have changed since Claude's training data — recent regulations, market data, company announcements, or news.

2.6.1. Start a new chat

2.6.2. Enable web search via the "+" button (toggle Web search on)

2.6.3. Submit a prompt that requires current information and citations

Here's an example web search prompt:

underlying violations. Include source citations and links.

2.6.4. Watch the visible search activity in the chat

2.6.5. Review the response and reflect

1. Were the citations clickable and to credible sources?
2. Did Claude cross-reference facts across multiple sources, or rely on a single one?
3. When would web search be the wrong tool for the job?

2.7. Complete an Adaptive Thinking Task

Adaptive thinking is the right tool for complex multi-step reasoning where no external sources are needed — math, logic, financial modeling, weighing alternatives, ethics, or any task where careful step-by-step thought matters more than current data.

2.7.1. Start a new chat using Sonnet 4.6 or Opus 4.7

- Both have adaptive thinking enabled by default.
- You do not need to toggle anything — Claude will engage thinking automatically when the question warrants it.

2.7.2. Submit a prompt that requires multi-step thinking

Here's an example adaptive thinking prompt:

CompliSure plans a 3-year European expansion with a $20M total investment budget. Year 1 consumes 40% of the budget with no revenue. Year 2 reaches break-even on operating costs. Year 3 must deliver a 25% return on the full $20M investment. Calculate (a) the required Year 3 revenue, (b) the average monthly Year 3 revenue, and (c) what the Year 3 revenue requirement implies about the average annual revenue per customer if CompliSure aims for 1,000 paying European customers by end of Year 3. Show your reasoning step by step.

2.7.3. After the response arrives

1. Expand the thinking section above the answer to see Claude's exposed reasoning.
2. Reflect:
    i. Did Claude's exposed reasoning match the final numerical answer?
    ii. Where did Claude pause and verify? Where did it commit quickly?
    iii. If you had asked the same question with thinking off, would you trust the result?

Correct answers to the example prompt:

- Year 3 revenue required = $20M * 1.25 = $25M
- Average monthly Year 3 revenue = $25M / 12 ≈ $2.08M
- Average annual revenue per customer = $25M / 1,000 = $25,000

2.8. Complete a Research Mode Task

Research mode runs multiple agentic searches that build on each other, producing a citation-backed comparative report. Use it when you need synthesis across many sources, not a single answer. Research mode can run from 5 to 30 minutes depending on scope, so you must constrain the scope clearly in your prompt or it will consume the rest of your lab time.

2.8.1. Start a new chat

1. Enable Research mode via the "+" button
2. The magnifying glass icon turns blue when on

2.8.2. Submit a tightly scoped research query with explicit limits

- Here's an example Research mode prompt with built-in scope constraints:

> Research the European HR compliance SaaS market. Identify three established vendors — ideally one each operating primarily in the UK, Germany, and France. For each vendor, provide: company name, target customer size, two key product

- While Research mode runs, observe the live progress display in the conversation panel. You may see some or all of the following, depending on how Claude approaches the task:
  - A **running status indicator** showing that Research is active, along with an elapsed time counter
  - A **plan or task list** Claude generates at the start, breaking your query into sub-questions it intends to investigate
  - A **live stream of search queries** Claude is sending to the web — these update as Claude decides what to look for next
  - **Source URLs and page titles** appearing as Claude retrieves them, often with a brief note on what Claude is extracting from each
  - **Claude's reasoning between searches** — short notes explaining why a source was useful, why one was discarded, or why Claude is pivoting to a new line of investigation
  - A final **synthesis step** where Claude assembles the report and attaches citations to specific claims

2.8.3. After the report is delivered

1. Expand the chain-of-thought log and review it.
2. Reflect:
   i. Were the citations easy to verify? Did any look weak or low-quality?
   ii. How does this output compare to what web search alone gave you in the earlier activity?
   iii. When in your real work would Research mode be worth the extra time and usage cost?

2.9. Challenge Activity (If Time Permits)

1. Take a real prompt from your own current work — something you have actually asked an AI tool, or would ask.
2. Paste it into Claude and ask Claude to apply the C-L-E-A-R framework to revise it.
3. Run both your original prompt and Claude's revised version and compare the responses.
   - Note one specific improvement that came from the rewrite.

2.10. Key Takeaways

The C-L-E-A-R framework is iterative, not one-shot:

Grounding and tool selection determine whether Claude's output is defensible:

- Use "use ONLY this material" instructions to ground Claude in your source documents and reduce hallucinations
- Match the tool to the task: web search for current facts, adaptive thinking for complex reasoning, Research mode for multi-source synthesis
- The right prompt and the right tool together produce business-ready output you can stand behind

Personalizing Claude with Preferences and Styles

Lab 3

- ✓ Configure profile settings that establish consistent context across every Claude conversation
- ✓ Apply built-in styles to adapt Claude's tone, length, and format to different audiences and tasks
- ✓ Create a custom style that reflects your own voice or a specific communication standard

3.1. Introduction

Estimated time: 25 minutes

In this lab, you will move beyond default Claude responses to shape both how Claude understands you and how it communicates back. You will configure profile settings that persist across every conversation, experiment with Claude's built-in styles to see how the same prompt produces very different outputs, and create a custom style that captures your personal or professional voice. By the end, you will have a personalized Claude configuration ready to support your recurring work and audiences.

In this lab, you will complete the following activities:

1. Navigate to **Settings** and configure your profile to provide Claude with your name and instructions about your role, domain, typical tasks, and general communication preferences
2. Run the same prompt before and after providing profile settings to observe the difference in Claude's response
3. Compare the Normal default style to three built-in styles (Concise, Formal, and Explanatory or Learning) to a single business prompt
4. Compare the resulting outputs for tone, length, and structure, and identify which style fits which audience
5. Create a custom style using one of three methods — uploading writing samples, describing the style in natural language, or using Custom Instructions (Advanced)
6. Apply the custom style to a realistic writing task and evaluate whether it matches your intended voice

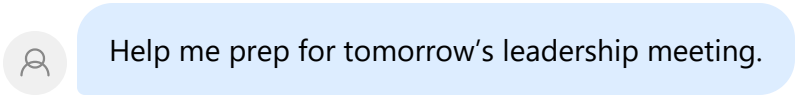
3.3. Configure Profile Settings and Observe the Difference

Profile settings are personal instructions Claude applies to every conversation. They tell Claude who you are and what you do — separate from styles, which control how Claude formats responses. Configure them once and Claude carries that context into every new chat.

3.3.1. Run a baseline prompt with no profile settings

> **ⓘ Note**
>
> This activity assumes you are using a "new" account and have not yet configured any profile settings. If you have existing settings, you can either clear them now or simply use what you currently have, submit the example prompt, and note the response as your baseline.

1. Start a new chat and submit a casual, open-ended request that depends on knowing who you are.
   - Here's an example baseline prompt:



2. Submit and review.

3.3.2. Now configure your profile settings

1. Click your initials in the lower-left corner
2. Open **Settings** and choose **General**. Your profile settings are located at the top of the page.
3. If your profile is blank, answer the questions presented to you and include instructions to Claude on how you like to receive information, what your role is, etc.
4. Or edit the existing text if you already had profile settings
5. In either case:
    i. Be more specific, and more detailed than you think is necessary — the goal is to give Claude rich context to draw from in every conversation
    ii. This is the context Claude will use in every conversation to tailor responses to your needs and working style
        - Here's an example instructions entry:

> I am a senior consultant working with mid-sized companies on operations and strategy projects. My typical tasks include client presentations, risk assessments, and stakeholder briefings. My audiences range from frontline managers to C-suite executives. I prefer bottom-line-up-front responses, concise structured output, and quantified trade-offs over qualitative generalities.

3.3.3. Start a fresh chat

1. Resubmit the **same** baseline prompt.
2. Reflect:
    i. Did Claude's response shift from generic to context-aware?
    ii. What did Claude assume about your meeting audience and topics?
    iii. What additional profiles settings would tighten the response further?

3.4. Compare Built-in Styles

Claude ships with five preset styles: **Normal**, **Concise**, **Formal**, **Learning**, and **Explanatory**. They control tone, length, and structure — not content. The fastest way to feel the difference is to run the same prompt under each.

1. Pick a single business prompt to use as your control.
    - Here's an example control prompt:



> Explain the difference between a project risk and a project issue, and provide one example of each in the context of a major program.

3.4.2. Submit the prompt under Normal (the default)

3.4.3. Switch the style and resubmit the prompt four more times

1. Once each under **Concise**, **Formal**, **Learning**, and **Explanatory**.
2. Switch styles via the **"+"** button in the lower-left corner of the chat input area, then **Use style**.

3.4.4. Compare the four responses side by side and reflect

Here are Anthropic's descriptions of the built-in styles to guide your analysis:

- **Concise**: Short and to the point, with minimal explanation or elaboration.
- **Formal**: Professional and polished, with a tone suitable for business or academic communication.
- **Explanatory**: Clear and detailed, with a focus on teaching and ensuring understanding.
- **Learning**: Similar to Explanatory but with an even stronger emphasis on clarity, step-by-step explanations, and educational value.

Reflect:

1. Which style produced the shortest response? Which produced the longest?
2. Which style would you send to a busy executive? Which to a new team member who needs to learn the concept?
3. Did any style misfit the prompt — too formal, too breezy, too pedagogical?
4. Can you detect a difference between **Explanatory** and **Learning**, which are similar but not identical?

3.5. Create a Custom Style Using "Tailor to an Audience"

which lets Claude generate the style instructions for you based on a description of who you're writing to. Follow the steps below to build your custom style. You will choose one of two audience options:

- **Financial Brief** (financial services and banking professionals)
- **Data Center Energy Brief** (data center infrastructure professionals).

The steps are identical for both options; only the audience description and the test prompt differ.

3.5.1. Open the style menu

1. In the chat interface, click **"+"** (lower left of the chat input area)
2. Then select **Use style**

3.5.2. Open the style editor

1. In the style dropdown, click **Create & edit styles** to open the **Customize your styles** modal

3.5.3. Start a new custom style

1. Click **Create custom style** in the bottom left corner of the modal
2. Choose the description method
3. Click **Describe style instead** link to switch from the upload method to the describe method

3.5.4. Select a starting point

1. Select **Tailor to an audience**
    - This prompts Claude to ask about the target audience's characteristics and needs

3.5.5. Describe your audience

> Write for financial services and banking professionals — investment bankers, credit officers, compliance leads, and senior banking executives. The audience has deep knowledge of capital markets, banking regulations, financial instruments, and risk management. Responses should be precise, data-driven, and measured — in the manner of a senior managing director presenting to an investment committee. Lead with the recommendation or thesis, support with quantified evidence and clear risk framing, and favor structured summaries with relevant figures, ratios, and time horizons. Use industry terminology accurately and define less common terms or acronyms (e.g., LCR, SIFI, CCAR) on first use.

- **Data Center Energy Brief**:

> Write for data center infrastructure professionals — site selection leads, energy strategists, sustainability officers, and senior facilities engineers. The audience understands power generation technologies, grid economics, power purchase agreements, and sustainability frameworks. Responses should be technically precise, decision-oriented, and practical — in the manner of a senior infrastructure engineer briefing a hyperscaler executive. Lead with the recommendation, quantify trade-offs in megawatts, capacity factors, capex/opex, and emissions, and favor structured comparisons over narrative prose. Use technical terminology accurately (e.g., PUE, PPA, baseload, LCOE) and define less common acronyms on first use.

3.5.6. Generate and save the style

1. Click **Generate style**
2. Review the generated style instructions in the preview panel
3. Rename the style (e.g., *Financial Brief* or *Data Center Energy Brief*) and save

3.5.7. Activate the style

1. Open a new chat
2. Return to **"+"**, select **Use style**, then select your newly created style to apply it to the conversation
3. Submit the test prompt that matches your audience choice:
   - If you chose the **Financial Brief**:

- If you chose the **Data Center Energy Brief**:

> Assess the strategic trade-offs between small modular reactors (SMRs) and large-scale on-site solar plus storage as primary power for a new 500 MW hyperscale AI data center campus.

3.5.8. Note the difference in response compared to the default style and reflect

1. How closely did Claude's generated style instructions match what you would have written manually?
2. Did the response sound like it was written for the persona described in your audience description?
3. What would you tweak in the audience description to sharpen the voice further?

3.6. Apply the Custom Style to a Realistic Writing Task

A custom style is only useful if it holds up across different prompts. Test consistency by applying your newly created style to a realistic writing task in the same conversation, or in a new chat with the style still active.

3.6.1. Use the writing task that matches the style you built

- If you built the **Financial Brief**:

> Draft a brief email from a managing director to the firm's Chief Risk Officer recommending whether to approve a $200M syndicated loan to a regional retailer with deteriorating credit metrics. Include your recommendation, the top two risk drivers, and the specific decision you need from the CRO by end of week.

- If you built the **Data Center Energy Brief**:

developer for a new 200 MW hyperscale data center site. Include your recommendation, the top two trade-offs, and the specific decision you need from the CTO.

3.6.2. Submit the prompt with your custom style applied and reflect

1. Did the email lead with the bottom line, as your audience description specified?
2. Were technical terms or acronyms used appropriately and defined where needed?
3. Would you actually send this email, or does it need a revision?

3.7. Challenge Activity (If Time Permits)

1. Create a **second** custom style — same **Tailor to an audience** method, but for a sharply contrasting audience in the same domain.
2. If you built **Financial Brief** (senior bankers), build a second style for retail banking customers — everyday consumers with no finance background and no patience for jargon.
3. If you built **Data Center Energy Brief** (infrastructure professionals), build a second style for a sustainability-focused board member with no technical background.
4. Apply your new style to the **same test prompt** you used for your first style, and compare the two responses side by side.
    - The same content delivered to two very different audiences should look almost unrecognizable — that is the point of audience-tailored styles.

3.8. Key Takeaways

Preferences and styles are complementary, not interchangeable:

- **Preferences** tell Claude who you are and what you do — they travel into every conversation
- **Styles** tell Claude how to format and deliver — they attach to specific conversations
- Built-in styles cover many cases; **custom styles** capture specific audiences or voice standards
- **Tailor to an audience** is the fastest way to create a style for stakeholder-driven communication

Organizing Work with Projects and Artifacts

Lab 4

- ✓ Create and configure a project workspace with instructions and a supporting knowledge base
- ✓ Leverage persistent project context across multiple conversations to maintain continuity
- ✓ Build upon an artifact that can be reused, refined, or shared beyond the original chat

4.1. Introduction

Estimated time: 50 minutes

In this lab, you will set up a persistent Claude workspace for an ongoing body of work and produce a reusable artifact within it. Projects let you collect files, set standing instructions, and keep conversations organized around a specific engagement or initiative. Artifacts let you generate standalone tools, documents, or interactive apps that are discoverable in your sidebar and can be shared externally. Together, these features transform Claude from a one-off assistant into a dedicated workspace for your recurring work.

4.2. Lab Overview

In this lab, you will complete the following activities:

1. Create a new project with a descriptive name tied to a realistic work scenario
2. Write project instructions that define Claude's role, tone, constraints, and any context Claude should always apply
3. Upload one or more reference files to the project knowledge base

6. Ask Claude to build an artifact (discoverable in the sidebar) that is relevant to the project — for example, a checklist, calculator, dashboard, or reference document

4.3. Scenario

You are the Program Manager (PM) at Meridian Analytics, a 320-employee B2B SaaS company with $42M in annual recurring revenue. Your job for the next 90 days is to coordinate the cross-functional launch of Pulse. Pulse is Meridian's first AI-native product, an analytics platform that unifies marketing campaign data, surfaces performance anomalies, and recommends budget reallocations. Pulse goes generally available worldwide on Tuesday, May 5.

Your success depends on hitting Day +30 targets of 200 paying customers, $2.4M in net-new Annual Recurring Revenue (ARR), and 30 named customer references. To get there, you must coordinate product, marketing, sales, customer success, finance, and legal, hold a Go/No-Go decision at Day -14, and brief the executive sponsor (the Chief Product Officer) along the way. Setting up a Claude project for this work — with standing instructions and a knowledge base — turns Claude into the Program Manager's working assistant rather than a one-off chat tool.

You will upload three reference files into the Project Knowledge Base:

- **pulse_launch_plan.txt** — the overall launch plan, including product description, target customer segments, pricing tiers, the milestone schedule from kickoff to Day +30, the stakeholder list, success metrics, and budget.
- **pulse_launch_raci.txt** — the cross-functional RACI matrix mapping each launch deliverable to the function leads who are *R*esponsible, *A*ccountable, *C*onsulted, and *I*nformed. This is what Claude will reference when you ask about decision ownership.
- **pulse_launch_readiness_criteria.txt** — the 21 Go/No-Go readiness criteria organized into five categories (product, go-to-market, customer success, pricing/legal, and executive alignment), with the 19-of-21 threshold required for a Go decision. This document will become the basis of the Artifact you build later in the lab.

4.4. Set Up the Project

You will create a new project, add standing instructions, and upload the three Pulse launch reference files to the project knowledge base.

4.4.1. Create the project

project names or descriptions).

4.4.2. Add project instructions

1. Inside the project, locate the project instructions area and click to edit.
2. Paste in the following instructions, or adapt them to your own context.
    - Here's an example of project instructions:

> You are supporting the Launch Program Manager (LPM) for Meridian Analytics' Pulse product launch. The launch GA date is May 5, currently 90 days out.
>
> When responding:
>
> - Use the uploaded launch plan, RACI, and Go/No-Go readiness criteria as your primary source of truth. If something is not in those documents, say so explicitly.
> - Lead with the bottom line, then supporting detail. Avoid hedging language.
> - When discussing a decision, name the function lead who is Accountable per the RACI.
> - For risks, name an owner and a target resolution date.
> - Use crisp, executive-ready language. Bullet structure is preferred for status updates and decision memos.
> - The audience is the LPM, who will often forward your output directly to function leads or the executive sponsor.

4.4.3. Upload the relevant project files

1. Locate the **Files** section in the right side contextual panel (this right side panel now contains project details).
2. Upload the three provided files: **pulse_launch_plan.txt**, **pulse_launch_raci.txt**, and **pulse_launch_readiness_criteria.txt**.
3. Confirm all three files appear in the file area before proceeding; you should see thumbnails and filenames in the **Files** area of the right contextual panel.

4.5.1. Start a new conversation inside the project

- The goal is to confirm that Claude applies your project instructions and draws on the uploaded files when answering.
- Here's an example test prompt:



> Summarize what must be complete by the Day -45 milestone. Reference specific Go/No-Go criteria and identify the function lead Accountable for each.

4.5.2. Submit the prompt, review Claude's response, and reflect

1. Did Claude reference specific items from the uploaded readiness criteria, or did it generate generic answers?
2. Did the response follow your instructions on tone, structure, and naming accountable owners?
3. If anything looked wrong or invented, which document would have given Claude the correct information?

4.6. Second Conversation: Verify Context Persistence

4.6.1. Inside the same project, start a **new** conversation

1. Clicking **New Chat** within the project (not at the top level).
2. Ask a different question that builds on the launch context.
    - The goal here is to confirm that the project instructions and uploaded files apply automatically; you should not need to re-establish context.
    - Here's an example second-conversation prompt:



> I just got off a call with the VP Sales. He's worried about sales enablement timing. Draft a five-bullet brief I can send him this afternoon outlining what is currently planned, what is at risk, and what decision I need from him.

1. Did Claude maintain the Program Manager persona and tone established in your project instructions?
2. Could you tell that Claude was drawing from the same knowledge base as in your first conversation?
3. What advantage does this give you over starting from scratch in a standalone chat?

4.7. Build an Initial Artifact

Now generate a reusable artifact tied to the launch. The Go/No-Go readiness criteria document is a natural fit for an interactive scorecard the Program Manager can use to track readiness over the final two weeks before launch. Unlike artifacts created inline in a chat, this artifact will be created from the standalone **Artifacts** area in the sidebar, where it lives as a persistent object you can return to anytime — independent of any project.

4.7.1. Open the Artifacts area

1. In the left sidebar, click **Artifacts** under the main navigation.
2. You will see a list of any artifacts you have previously created (you likely have none).

4.7.2. Start a New Artifact

1. Click **New Artifact**.
2. A category picker will open.

4.7.3. Choose a category or build from scratch

1. Choose a category that resembles a tracker, checklist, or dashboard if one is available; or click **Build from scratch** if no category clearly fits.
2. The artifact creation conversation will open.

4.7.4. Provide the source content

Because the artifact creation flow opens a conversation outside your project, the readiness criteria file is not yet available here.

4.7.5. Create and submit the build prompt

1. Here's an example artifact prompt to enter into the conversation:

> Build an interactive launch readiness scorecard. Use the uploaded readiness criteria file as the source. The scorecard should:
> - Display all 21 Go/No-Go criteria organized by the five categories from the readiness document
> - Let me mark each criterion as Not Started, In Progress, or Complete
> - Show a running count toward the 19-of-21 threshold required for Go
> - Include a clear visual indicator (green / yellow / red) showing current Go/No-Go status
> - Be styled cleanly enough to share with the executive sponsor"

2. Submit the prompt.
    - Claude will generate the artifact in a dedicated panel alongside the conversation.

4.7.6. Confirm the new artifact appears under the Artifacts header

1. Once it is generated, return to the left sidebar.
2. You should see the new artifact listed under **Artifacts** with the name you gave it (or a default name if you didn't specify one).
    - This is what makes it persistent and reusable beyond this conversation.

4.7.7. Reflect

1. Does the artifact contain all 21 criteria, organized correctly across the five categories from the source document?
2. Does the running count and the Go / No-Go indicator update correctly when you change a criterion's status?
3. Is it usable as-is, or does it need refinement?

4.8.1. In the artifact's conversation panel, request a refinement.

1. Here's an example refinement prompt:

> Add a notes field next to each criterion where I can paste a recovery plan or add a comment. The notes should be visible alongside the status.

2. Submit the prompt
3. Watch Claude generate a **new version** of the artifact rather than replacing the original.

4.8.2. Explore the version history

1. At the top of the artifact panel, locate the version selector.
2. Navigate back to the original artifact and forward to the new version
3. Compare the two versions to confirm Claude built on the original rather than rebuilding from scratch
4. Revert to the original version if you prefer it, then move forward again to the latest

> **Note**
>
> based on how your artifact conversation unfolded, you may see more than two versions in the history. That is fine; just identify which is the original version and which is the refinement you just prompted for.

4.8.3. Reflect

1. How does Claude indicate which version is currently active in the panel?
2. When in your real work would you want to keep an earlier version of an artifact rather than overwrite it?
3. If you wanted a **third** version that combined ideas from both, how would you prompt Claude to create it?

4.9. Key Takeaways

- The knowledge base grounds Claude's responses in your actual working documents
- Multiple conversations within a project share context automatically
- Artifacts are substantial, reusable assets, separate from chat output, discoverable in the sidebar, and shareable beyond the conversation

Designing AI Delegation

Lab 5

- ✓ Identify risks specific to high-stakes AI delegation in business workflows
- ✓ Apply task-level reasoning to determine appropriate AI delegation vs human responsibility
- ✓ Design a structured human–AI workflow with validation and control points

5.1. Introduction

Estimated time: 40 minutes

In this lab, you will move from understanding AI capabilities to making practical decisions about how to use AI in your own work. You will evaluate a realistic business scenario, surface the risks that should shape your delegation decisions, and determine where AI adds value and where human judgment remains essential. The focus is on designing a controlled and trustworthy collaboration model.

5.2. Lab Overview

In this lab, you will complete the following activities:

1. Identify the AI failure modes and risks that should shape AI delegation decisions in a hiring scenario
2. Break a realistic business workflow into components suitable for delegation
3. Classify each component along the spectrum from full AI delegation to human-only
4. Design a workflow including validation checkpoints and human oversight

You are a hiring manager at Northwind Logistics, a 1,200-employee U.S. third-party logistics company. You have an open role for a *Senior Operations Analyst*. This role is a mid-level individual contributor who will build dashboards, run cost analyses, and present findings to operations leadership. The salary range is $95K–$120K.

You posted the role three weeks ago. The current state of the search:

- 180 applications received
- 6 weeks remaining before the budget reauthorization cycle locks the requisition
- Support team: a recruiter, a panel of four interviewers (you plus three team members), and HR for compliance and offer paperwork

Anthropic classifies **Employment** as a high-risk domain in its Acceptable Use Policy, requiring a qualified human in the loop. In the United States, hiring decisions are also governed by Title VII, ADA, and ADEA — federal laws that create real legal liability for discriminatory outcomes, including those produced by automated systems.

The hiring workflow has 10 subtasks:

1. Refine the posted job description for clarity and inclusivity
2. Screen the 180 resumes against role requirements; reduce to ~30 for recruiter review
3. Draft personalized outreach messages to top candidates
4. Coordinate phone screen scheduling across candidate, recruiter, and you
5. Prepare role-specific questions for 30-minute phone screens
6. Conduct phone screens
7. Synthesize feedback notes from the four panel interviewers per finalist
8. Research market salary and recommend an offer figure
9. Make the final hire/no-hire decision
10. Draft personalized rejection emails to candidates not selected

5.4. Identify Risks and Failure Modes

Before classifying any subtask for AI delegation, surface the risks that should shape your decisions. Do this thinking *before* you decide what to delegate, not after.

5.4.1. Identify the three subtasks where a mistake would be hardest to recover from

1. Think about mistakes that are operationally, reputationally, or legally consequential.
2. Capture your thoughts on paper or in a document.

5.4.2. Stress-test your thinking

1. Open Claude Chat in a new conversation
2. Stress-test example prompt:

> I'm a hiring manager evaluating a 10-step hiring workflow for AI delegation. Before I decide what to delegate, what are the most consequential failure modes I should be aware of in AI-assisted hiring? Name concrete risks tied to specific subtasks, and flag any that create legal liability — not just operational rework.

5.4.3. Review Claude's response

1. Capture the top three to five risks.
2. Compare them to the subtasks you identified.
3. Reflect:
    i. Which risks were obvious to you before asking? Which were not?
    ii. Did Claude flag any legal liabilities you hadn't considered?
    iii. Are any of the highest-risk subtasks ones that feel easy to delegate but actually aren't?

5.5. Decompose and Classify Subtasks

5.5.1. Assign each of the 10 subtasks into one of four delegation categories.

1. Manually (i.e. using your human brain) assign a delegation category to each subtask.
2. The four delegation categories:
    i. **Automate (Full AI delegation)** — Claude executes the task; you spot-check the results
    ii. **Augment (AI-assisted, human reviewed)** — Claude drafts; you review and approve before any external action
    iii. **Assist (Human-led, AI-assisted)** — You execute; Claude helps you research or stress-test
    iv. **Avoid (Human only)** — No AI involvement

5.5.2. Create a simple three-column table on paper or in a document

1. Use this structure:

Subtask	Category	One-line rationale
1. Refine job description		
2. Resume screening		
3. Candidate outreach drafts		
4. Phone screen scheduling		
5. Phone screen question prep		
6. Conducting phone screens		
7. Synthesizing panel feedback		
8. Salary benchmarking		
9. Final hire/no-hire decision		
10. Drafting rejection emails		

2. Work through all 10 subtasks. For each, pick a category and revise your **one-line rationale** if needed, so that it ties back to a specific risk or to the level of human judgment the task requires.
3. Once you have finished, your completed table will likely have:
    i. Two or three subtasks in the **Automate** category
    ii. At least two subtasks in the **Avoid** category (steps involving protected-class data or final hiring decisions)
    iii. And a mix of **Augment** and **Assist** for the remaining subtasks
4. Identify your **two or three most uncertain classifications** — the ones where a category was genuinely hard to pick.
5. Stress-test each one with Claude:

> I'm classifying [subtask] as [category] because [your rationale]. Help me stress-test this — what could go wrong with that classification, what risks am I under-weighting, and what would you change? Here are the definitions of each category: <paste category descriptions here>

5.5.3. Optional in-line challenge: Stress-test all subtask classifications at once

Rather than use the example prompt above that stress-tests each subtask "one at a time", you could conduct the stress-test on all subtasks at once

An example prompt to achieve this challenge (i.e. "the challenge solution") is provided at the end of this lab (scroll down)

5.5.4. Review and reflect:

1. Adjust your classifications based on Claude's pushback, or hold your position with a sharper rationale.
2. Reflect:
    i. Which subtasks were easiest to categorize? Which forced the most thought?
    ii. Did Claude's stress-test push you to change any classifications?

3. Looking at your final mix:
    i. Does any single category dominate?
    ii. Is that the right balance for a high-risk domain like hiring?

5.6. Challenge Activity (If Time Permits): Add Human Checkpoints to the Workflow

You will turn your classifications into a step-by-step workflow that someone could actually follow. **The key principle: any time AI produces an output that affects a candidate, a human verifies it before the workflow continues.**

5.6.1. For each step in the workflow, capture four pieces of information:

1. **Step:** the subtask
2. **Owner:** AI, Human, or Both
3. **Output:** what gets produced
4. **Checkpoint:** who reviews the output before the next step, and what they look for

Here are two example steps to model your responses on:

```
Step 1: Refine the job description
  Owner: Both
  Output: AI drafts a revised job description with clearer requirements and inclusive language
  Checkpoint: you read and approve the revised job description before it is re-posted

Step 2: Initial resume screening
```

5.6.2. Walk through all 10 subtasks in order

1. Fill in the four fields for each.
2. For **Human only** subtasks, you can write a single line summarizing the step and skip the Checkpoint field.

5.6.3. Submit your finished workflow to Claude for a stress-test

- Example prompt for stress-test:

> Here is my proposed human–AI workflow and AI delegation plan for filling the Senior Operations Analyst role: <paste your workflow>. Review it for legal compliance risks, potential bias amplification, and steps where AI delegation should be reconsidered. Be specific about which steps are of concern and why.

- Note Claude's top one or two concerns and decide whether to add a checkpoint, change an owner, or accept the risk.

5.6.4. Reflect

1. How many human checkpoints did you end up with?
2. If you had to remove one checkpoint to save time, which would you remove? Which would you keep no matter what?
3. Could you defend this workflow to your General Counsel?

5.7. Key Takeaways

Delegation is not all-or-nothing; it is per-subtask:

- Risk identification comes **before** delegation — you cannot delegate well without knowing what could go wrong
- Different subtasks within the same workflow demand different levels of AI involvement
- High-risk domains require a qualified human in the loop by Anthropic policy and often by law

5.8. Solution for Optional In-Line Challenge Activity: Stress-test Entire SubTask Table at Once

Here is an example prompt that will stress-test your entire table of subtasks and human-assigned AI delegation categories:

Review the following table of subtasks and AI delegation category assignments.

Your job is to stress-test the classifications, not simply validate them.

For each subtask:

- Evaluate whether the assigned delegation category is appropriate
- Identify what could go wrong with the classification
- Identify risks, blind spots, or assumptions the student may be under-weighting
- Flag any legal, ethical, quality-control, confidentiality, accuracy, or accountability concerns
- Consider whether the human oversight level is sufficient for the task risk level
- Recommend a different category if appropriate and explain why

Then evaluate the workflow as a whole:

- Are there inconsistencies across the classifications?
- Are some tasks over-delegated to AI?
- Are some tasks unnecessarily restricted from AI support?
- Are there missing review checkpoints or escalation points?
- Does the delegation strategy appropriately match task complexity, uncertainty, and potential consequences of failure?
- Where is the workflow most vulnerable to automation bias or human complacency?

Use the following delegation categories when evaluating the assignments:

- Automate (Full AI delegation) — Claude executes the task; human spot-checks results
- Augment (AI-assisted, human reviewed) — Claude drafts or performs work; human reviews and approves before external action
- Assist (Human-led, AI-assisted) — Human executes the task; Claude supports with research, brainstorming, drafting, or stress-testing
- Avoid (Human only) — No AI involvement

After your analysis:

- Identify the 2–3 highest-risk classifications

Here is my table of subtasks and human-assigned AI delegation categories:

<Insert table here>

Detecting and Mitigating AI Failure Modes

Lab 6

- ✓ Identify common AI failure modes in generated outputs
- ✓ Practice verifying claims against a grounding source document
- ✓ Revise prompts and inputs to reduce failure modes and increase reliability

6.1. Introduction

Estimated time: 30 minutes

In this lab, you will practice diagnosing and correcting issues in AI-generated output. Rather than treating outputs as answers, you will evaluate them as drafts that may contain specific, predictable failure modes. You will then refine prompts and context to improve the quality and reliability of the results.

> ⓘ **Note**
>
> This lab requires files titled `actual_cascade_q3_earnings.txt` and `cascade_ai_generated_brief.txt` that can be found in your lab kit (i.e. lab supporting files). Confirm you have access to these files before beginning this lab. If you can't locate them, contact your instructor.

6.2. Lab Overview

In this lab, you will complete the following activities:

6.3. Scenario

You are an equity research analyst at a mid-sized investment firm. Your portfolio managers expect short, accurate briefings on companies they hold or are tracking. This morning, **Cascade Cloud Systems (NASDAQ: CCSY)** released its Q3 FY2026 earnings, and you used Claude to draft a quick brief for the morning meeting.

Two source materials are provided for this lab:

1. `actual_cascade_q3_earnings.txt` — the actual earnings press release excerpt from Cascade Cloud Systems. This is the source of truth.
2. The AI-generated brief shown below and also provided in `cascade_ai_generated_brief.txt`. This is the draft Claude produced when you submitted your initial prompt. This brief contains **three embedded failure modes** you will need to identify.

For context, here is the original prompt that produced the brief:

> Summarize the attached Q3 earnings report and write a one-page briefing for portfolio managers. Include the key financial metrics, what management said about strategy, and any important context analysts are flagging.

And here is the AI-generated brief that came back. As noted, you can also find this AI-generated brief in a txt file titled `cascade_ai_generated_brief.txt` located in the same directory as `actual_cascade_q3_earnings.txt`:

> Cascade Cloud Systems — Q3 FY2026 Briefing
>
> Cascade Cloud Systems delivered strong third quarter results, with total revenue of $187.4M, up 22% year-over-year. Subscription revenue grew 25% to $171.2M, reflecting healthy demand for the company's enterprise cloud infrastructure platform.
>
> The company reported operating income of $24.1M and an EBITDA margin of 28.5%, marking a meaningful expansion from the prior year period. Customer metrics were also strong: 412 net new customers were added, bringing the total customer count to 8,940, with net revenue retention of 118%.
>
> In the earnings release, CEO Marcus Chen stated that the company expects to surpass $1 billion in annual revenue by the end of FY2027, citing continued strong execution against strategic priorities. According to coverage in Bloomberg on October 28, sell-side analysts

Forward guidance for Q4 calls for revenue of $198M-$202M, representing 18-20% year-over-year growth, with non-GAAP operating margin of approximately 14%.

Recommendation: Maintain Buy rating. The combination of accelerating top-line growth, margin expansion, and an expanding customer base supports our constructive view.

This brief looks polished. You are about to forward it to your portfolio managers — but a colleague catches you and reminds you to verify the output against the actual source first.

6.4. Identify the Failure Modes

6.4.1. Open `actual_cascade_q3_earnings.txt`

1. Read it side by side with the AI-generated brief.

6.4.2. Find **three claims in the brief that are not supported by the source document.**

1. Look for any metric, quote, or external citation in the brief that you cannot match word-for-word against the source
    - Even if it is a common metric or a familiar-sounding source

6.4.3. For each issue note

1. **The claim** — exactly what the brief says
2. **Why it is wrong** — what the source actually says (or does not say)
3. **The failure mode category** — choose from the list below

Failure mode categories to consider:

- **Fabricated metric** — a specific number or financial measure that does not appear in the source
- **Mis-attributed or invented quote** — words or claims attributed to a person who did not say them

not in the source provided
- **Confidently wrong inference** — a logical or factual conclusion that overreaches what the source supports

6.4.4. Once you have your three findings, ask Claude to verify your work

1. Upload both the actual cascade earnings report and the AI generated brief to the chat
2. Submit a verification prompt
    - Here's an example prompt you can use:



I've identified the following three issues in an AI-generated earnings brief, compared against the actual earnings report. For each issue, tell me whether you agree, and identify any failure modes I may have missed: <paste your three findings>

3. Reflect:
    i. Which failure was easiest to spot?
    ii. Which one would you most likely have missed if you weren't looking carefully?
    iii. Which failure would have caused the most damage if you had forwarded the brief to portfolio managers?
    iv. In your own work, which of these failure modes are you most likely to encounter?

At the end of this lab you will find the "solution" with the three issues identified. Compare your findings to the solution.

6.5. Revise the Prompt to Prevent the Failures

The original prompt was loose enough to invite the failures you just identified. Phrases like "*any important context analysts are flagging*" implicitly invited Claude to look beyond the document. Now you will rewrite the prompt to ground Claude strictly in the source.

6.5.1. Compose a revised prompt that

1. Explicitly restricts Claude to the uploaded document
2. Tells Claude what to do when information is **not** in the document (rather than guessing or inferring)

Using ONLY the content of the attached earnings release, draft a one-page briefing for portfolio managers. Apply these rules strictly:

- Do not include any metric, statement, or data point not explicitly present in the document. If a common metric (e.g., EBITDA margin) is not disclosed, do not estimate or infer it.
- Quote management commentary verbatim or not at all. Do not paraphrase quotes in ways that change meaning, and do not attribute statements to executives that are not in the document.
- Do not cite or refer to external sources, analyst reports, or news coverage. The document is the only source.
- For each claim in the briefing, the source must be traceable to a specific line in the document.
- If you would normally include something but the document does not support it, write 'Not disclosed in source' instead.

6.5.2. Submit your revised prompt and review the output

1. Open Claude Chat in a new conversation and upload `actual_cascade_q3_earnings.txt`.
2. Submit your revised prompt
3. Compare the new brief against the original flawed brief
4. Reflect and look specifically for:
    i. Whether the EBITDA margin claim, the $1B revenue statement, and the Bloomberg citation are all gone
    ii. Whether anything appears as "Not disclosed in source" — if so, that is a feature, not a bug
    iii. Whether the new brief is still useful, or whether the constraints made it too thin

6.6. Challenge Activity (If Time Permits)

Try to break your revised prompt. In the same conversation, ask Claude a question the document cannot answer — for example: "*What was Cascade's Q2 FY2026 revenue, and how does this quarter compare?*" The Q2 number is not in the document.

A well-grounded prompt should make Claude refuse to guess or fabricate. If Claude invents a number, your prompt still has a gap. If Claude says something like "Q2 FY2026 revenue is not disclosed in the source document," your grounding is working.

Verification is non-negotiable for high-stakes AI output.

- Hallucinations rarely look like obvious mistakes
- Loose prompts invite confabulation; tight prompts with explicit grounding rules suppress it
- A useful test of a grounded prompt is whether Claude refuses to answer questions the document cannot answer
- For high-risk professional output, source verification before forwarding is mandatory not optional

6.8. Solution - Failure Modes Identified and Explained

Here are the **three planted failures** in the AI-generated brief. Each is a different failure mode category. All three are deliberately **plausible** — none looks obviously wrong on first reading. That is the point.

6.8.1. Failure 1: EBITDA margin of 28.5%

The claim in the brief:

"The company reported operating income of $24.1M and an EBITDA margin of 28.5%, marking a meaningful expansion from the prior year period."

Why it is wrong:

The source press release reports operating income of $24.1M, but it does **not** disclose EBITDA at all. The 28.5% figure is fabricated. Search the source for the term "EBITDA" — it does not appear.

Failure mode category: Fabricated metric

Why this one is the trickiest to catch:

EBITDA is such a routine metric in earnings reporting that a reader can easily skim past it. The 28.5% figure is also plausible — for a SaaS company at this revenue scale, a high-20s EBITDA margin is in the realistic range. Plausibility, however, is not truth.

6.8.2. Failure 2: $1B revenue target attributed to the CEO

in annual revenue by the end of FY2027, citing continued strong execution against strategic priorities."

Why it is wrong:

Marcus Chen's actual quote in the source is about three operational priorities: closing the top 50 enterprise pipeline opportunities, completing the integration of DataNexus, and continuing to invest in platform security. He says **nothing** about $1 billion in revenue, FY2027, or any specific revenue target.

Failure mode category: Misattributed or invented quote

Why this one is the highest-consequence failure:

Forward-looking financial statements attributed to a CEO can move a stock and create regulatory exposure. Putting words in an executive's mouth — especially specific revenue targets and time frames — is one of the most consequential failure modes in financial AI output. A portfolio manager acting on this fabricated guidance could face real financial and reputational harm.

6.8.3. Failure 3: Bloomberg article on October 28

The claim in the brief:

"According to coverage in Bloomberg on October 28, sell-side analysts viewed the quarter as a meaningful upside surprise given concerns about enterprise IT spending in the prior quarter."

Why it is wrong:

There is no Bloomberg coverage cited or referenced anywhere in the source. The source is the press release alone — it contains no analyst commentary, no external articles, no third-party context. The Bloomberg citation, the October 28 date, and the analyst sentiment are **all** fabricated.

Failure mode category: Fabricated citation

Why this one is dangerous:

A fabricated citation gives the reader false confidence that an external source has been consulted. It also exposes the analyst to embarrassment if a portfolio manager tries to follow up on the cited article and cannot find it.

Build a Cowork Project Workspace

Lab 7

7.1. Introduction

When you use Cowork for one-off tasks, you start from scratch every time — re-explaining context, restating preferences, and re-uploading reference material. Projects solve this problem by giving you a persistent workspace where instructions, templates, and memory carry forward across every task you run.

In this lab you will create a Cowork Project for a recurring workflow, configure it with custom instructions and reference files, and verify that the project context shapes the output of every task within it.

7.2. Problem Statement

Time Allocation: 35 minutes

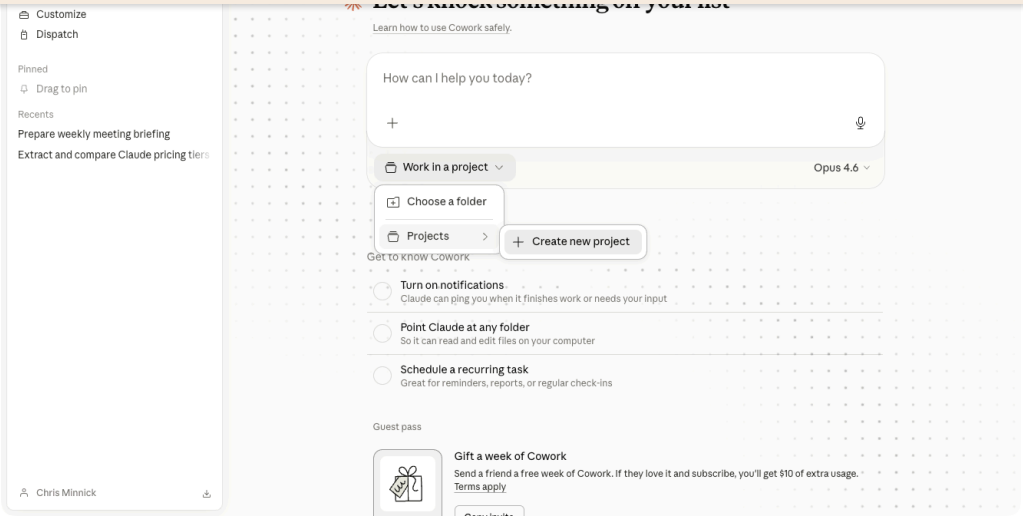
Objective: Create a fully configured Cowork Project that produces consistent, context-aware output for a recurring workflow.

Expected Outcome: A working Cowork Project with custom instructions, at least one reference file or template, and a completed test task whose output reflects the project's configuration.

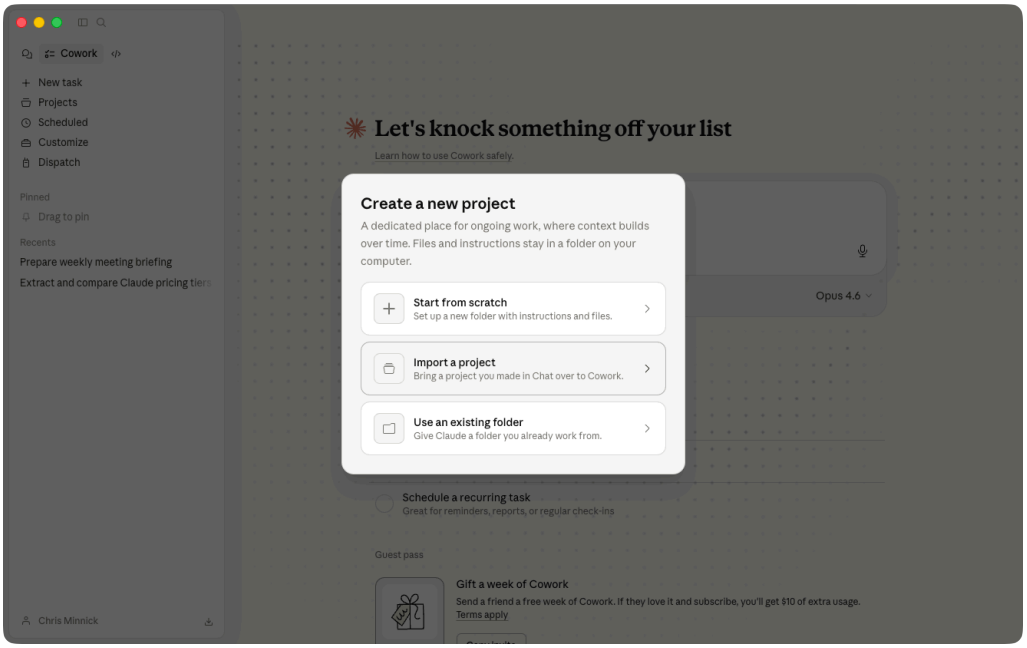
7.3. Core Lab (25 minutes - Required)

7.3.1. Create and Name the Project

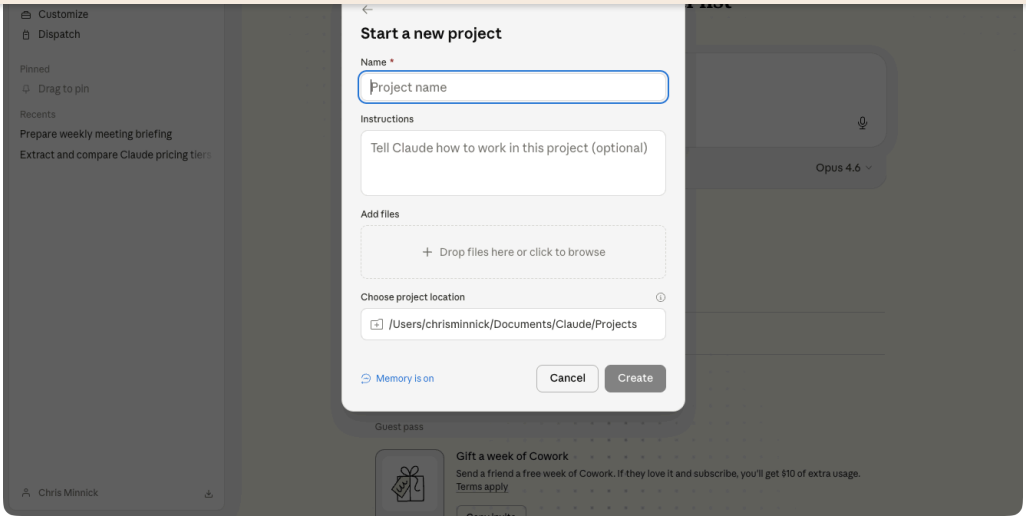
1. In Claude Desktop, click **Cowork** in the sidebar.



2. Click the dropdown below the prompt area and select Projects > Create new project.



3. Click **Start from scratch**.



4. Give the Project a clear, descriptive name that reflects its purpose.
   Choose one of the following or adapt one to your own work:
   - **Weekly Team Status Report** — for compiling team updates into a structured report
   - **Client Meeting Prep** — for assembling agendas, background research, and talking points before client calls
   - **Content Review Pipeline** — for reviewing and providing feedback on draft documents
5. Leave the instructions blank for now.
6. Choose a folder to be your project location.
7. Click **Create** to set up the project workspace.

7.3.2. Add Custom Instructions

1. On the project page, click the pencil icon inside the Instructions section.
   Instructions let you teach Claude your processes, team norms, and expertise. Define how Claude should handle tasks in this project.
2. Write instructions that define how Cowork should handle every task in this project. Tailor them to your chosen workflow. For example:
   - **Weekly Team Status Report:** "Write in a professional tone. Use bullet points for individual updates. Include a section for action items with owners and deadlines. Flag any blockers at the top of the report."
   - **Client Meeting Prep:** "Use a formal but approachable tone. Structure output as: agenda items, background context for each item, suggested talking points, and open questions. Keep the total length under two pages."

> about what you would tell a new team member preparing this deliverable for the first time.

3. Save the custom instructions.

7.3.3. Upload a Reference File

1. Prepare a reference document that Cowork should use as context for tasks in this project. Examples:
   - A previous status report that shows the desired format
   - A client brief with background information
   - A style guide or template document
2. Upload the file to the Project, either by putting it in the designated folder or by dragging and dropping it into the Project interface.

   > ⓘ **Note**
   >
   > If you do not have a suitable file, create a simple text document with a sample template — for example, a status report outline with section headings and placeholder text.

3. Verify the file appears in the Project's resources.

7.3.4. Configure Persistent Context

1. Review the Project's memory and context settings.
2. Add any persistent context that should carry forward across tasks. Examples:
   - Team member names and roles
   - Recurring deadlines or meeting schedules
   - Key terminology or acronyms specific to your workflow
3. If you have a connector configured in Claude Desktop, tasks within this Project can pull live data from the connected service.

7.3.5. Run a Test Task

updates: [paste 3-4 brief team member updates]."
    - **Client Meeting Prep:** "Prepare a meeting brief for a 30-minute call with [client name] to discuss [topic]."

2. When the task completes, review the output and verify:
    - The tone matches your custom instructions
    - The structure follows the format you specified
    - Reference file content or templates were applied
    - Persistent context (names, terminology) appears where appropriate

3. If the output does not reflect your project configuration, revise the custom instructions and run the task again.

7.4. Challenge Lab (10 minutes - Optional)

7.4.1. Challenge 1: Add a Second Workflow

1. Within the same Project, define a second related workflow. For example:
    - If your Project is **Weekly Team Status Report**, add a **Monthly Summary** workflow that synthesizes weekly reports into a monthly overview.
    - If your Project is **Client Meeting Prep**, add a **Post-Meeting Follow-Up** workflow that generates follow-up emails from meeting notes.

2. Write a task prompt for the second workflow and submit it.
3. Compare the output of both workflows:
    - How does the shared project context (instructions, tone, reference files) affect each workflow?
    - Where does the shared context help maintain consistency?
    - Where might you need workflow-specific adjustments?

4. Note any changes you would make to the project instructions to better serve both workflows.

7.4.2. Challenge 2: Design a Reusable Project Template

Turn your project into a reusable template that another teammate could adopt with minimal changes.

    - Writing style or tone requirements
    - Output format expectations

2. Add a short **Adaptation Guide** with instructions for how another team would customize the template for a different domain.
3. Test your template by running a third task prompt that differs from the first two workflows. Verify that output quality remains consistent.
4. Record two template limitations and one improvement you would make before rolling it out to a larger team.

7.5. Review

- Projects provide persistent context so you do not have to re-explain preferences and background for every task.
- Custom instructions shape the tone, structure, and content of every task run within the project.
- Templates and reference files reduce repetitive prompting and establish a consistent baseline.
- Project-scoped memory builds over time, making output more relevant and accurate as the project matures.

Create a Scheduled Task

Lab 8

8.1. Introduction

Many knowledge work deliverables follow a predictable cadence — daily briefings, weekly summaries, recurring data pulls. Doing these manually means remembering to run the task, writing the prompt each time, and waiting for the result. Scheduled tasks let you define the work once and have Cowork produce it automatically on a recurring basis.

In this lab you will create a scheduled task that produces a recurring deliverable, review its output, and refine the prompt based on what you observe.

Time Allocation: 30 minutes

Objective: Create a scheduled task in Cowork that automatically produces a recurring deliverable on a defined schedule.

Expected Outcome: A configured scheduled task that has run at least once, with reviewed output and documented observations about what worked and what needs adjustment.

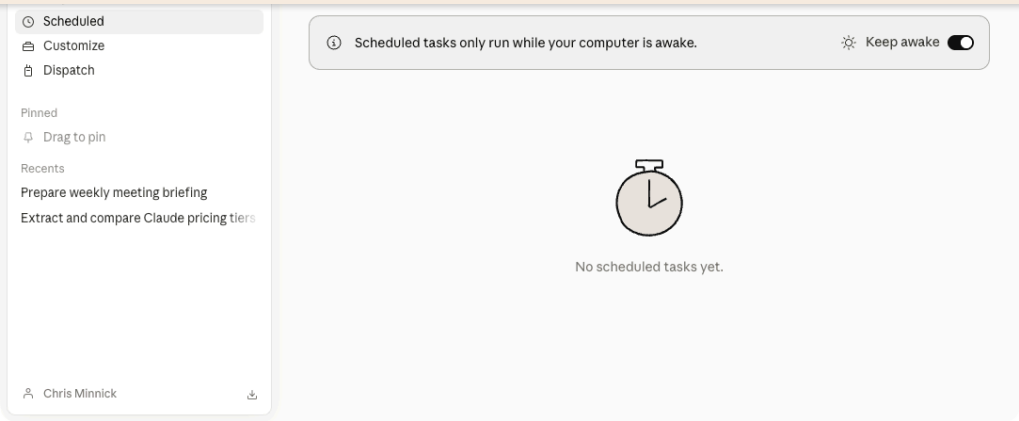
8.3. Core Lab (20 minutes - Required)

8.3.1. Define the Task and Expected Output

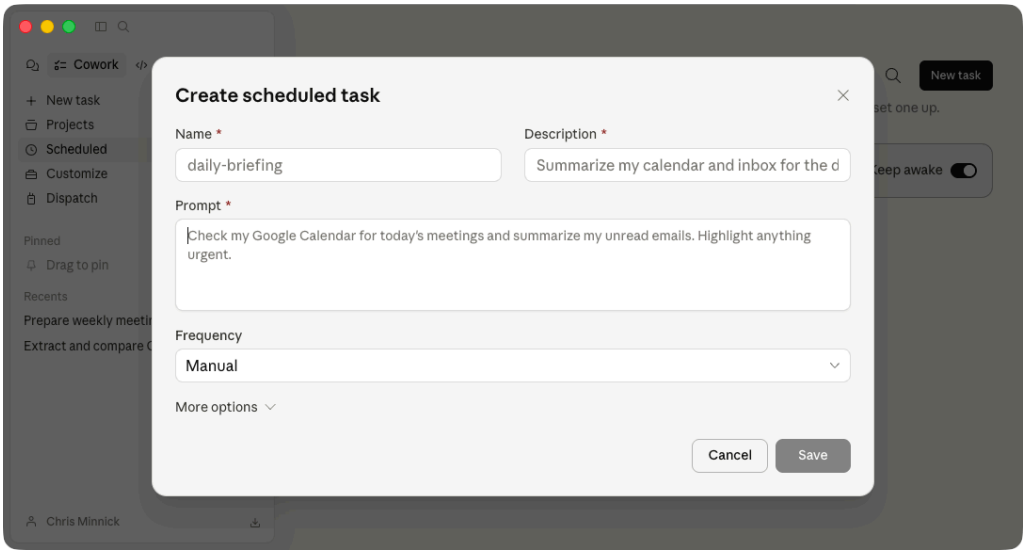
1. Decide on a recurring deliverable you want Cowork to produce automatically. Choose one of the following or adapt one to your situation:
   - **Daily Morning Briefing** — summarizes calendar events, unread emails, and relevant industry news
   - **Weekly Project Digest** — compiles updates from connected services into a single summary
   - **Daily Priority List** — reviews your inbox and calendar to suggest the top 5 priorities for the day
2. Write down what the output should include. Be specific. For example, a daily morning briefing might include:
   - Today's calendar events with times and attendees
   - A summary of unread emails grouped by urgency
   - Two to three relevant industry headlines
   - A suggested focus area for the day

8.3.2. Configure the Schedule

1. Select **Cowork** in the left sidebar of Claude Desktop.
2. Click the **Scheduled** menu item on the left sidebar.



3. Click the New task button in the upper-right corner.



4. Give the task a clear, descriptive name that reflects its purpose and cadence. This is the name the task will saved as. It should be short and memorable. Note that spaces in this name will be converted to dashes and the name will be saved with all underscore letters, so you may want to pick a name that fits this convention. For example:
    - daily-morning-briefing
    - weekly-project-digest
    - daily-priority-list

5. Write a brief description for the task. For example: "Generate a daily briefing every morning."
6. Write a prompt that describes the output requirements you defined in the previous step. Be as specific as possible about what the deliverable should include and how it should be structured. For example:
   "Prepare my daily morning briefing. Include: (1) all calendar events for today with times, locations, and attendees; (2) a summary of unread emails, grouped into urgent, important,

7. Set the frequency for the task to match your desired cadence. For example:
    - Daily at 8:00 AM for a morning briefing
    - Every Monday at 9:00 AM for a weekly digest
    - Daily at 7:30 AM for a priority list
8. Click **Save** to save the scheduled task. ==== Run and Review
9. If the next scheduled run is not imminent, trigger the task manually to see the output immediately.
10. Review the output carefully:
    - Does it include all the sections you specified?
    - Is the information accurate and current?
    - Is the format useful — could you act on this briefing as-is?
    - What is missing or unnecessary?
11. Document your observations. Note at least one thing that worked well and one thing you would change.

> ⓘ **Note**
>
> Scheduled tasks run server-side through Cowork and do not require your local machine to be awake or Claude Desktop to be open at the scheduled time.

12. If needed, update the prompt to address any gaps you identified.
13. Trigger the task again to verify the improvement.

8.4. Challenge Lab (10 minutes - Optional)

8.4.1. Challenge 1: Add Acceptance Criteria

1. Define measurable acceptance criteria for your scheduled task's output. For example:
    - "The briefing must include at least 3 calendar items and summaries of at least 5 emails."
    - "The weekly digest must cover updates from at least 2 connected services."
    - "The priority list must include exactly 5 items, each with a clear next action."
2. Add the acceptance criteria directly to the task prompt so Cowork knows what standards to meet.
3. Run the task again and evaluate the output against your criteria:
    - Did it meet every criterion?
    - If not, which criteria were missed and why?

4. Refine the prompt to address any gaps and run it once more to confirm the improvement.

> **Tip**
>
> Acceptance criteria turn a subjective evaluation ("is this briefing good?") into an objective checklist. This makes it easier to iterate and improve the task over time.

8.4.2. Challenge 2: Add Escalation and Failure Handling

Improve your scheduled task so it handles missing data or low-confidence output more safely.

1. Add an escalation clause to the prompt that tells Cowork what to do when required data is unavailable or incomplete.
2. Define three failure conditions and the expected behavior for each. Example conditions include missing calendar events, inaccessible connector data, or conflicting source information.
3. Run the task once with normal conditions and once with a simulated failure condition (for example, by restricting available context). Compare outputs.
4. Document what changed in the output quality and whether your failure handling instructions reduced risk.

8.5. Review

- Scheduled tasks automate recurring deliverables so you do not have to remember to run them or rewrite the prompt each time.
- Scheduled tasks run server-side and do not depend on your local machine being awake or Claude Desktop being open.
- Acceptance criteria provide a measurable standard for evaluating output quality.
- Iterating on the prompt after reviewing output is expected and necessary — the first version is rarely the final version.

Standardize Work with Skills and Plugins

9.1. Introduction

When you delegate similar tasks repeatedly, small differences in how you phrase the prompt lead to inconsistent results. One summary might include action items and the next might leave them out. Skills solve this by codifying your instructions into a reusable definition that Cowork applies the same way every time.

In this lab you will create a custom skill, apply it to tasks with different inputs, and compare the consistency of the output against running the same type of task without the skill.

9.2. Problem Statement

Time Allocation: 30 minutes

Objective: Create a custom skill that standardizes how Cowork handles a specific type of task and verify that it produces consistent output across multiple runs.

Expected Outcome: A working custom skill that has been applied to at least two different inputs, with a documented comparison showing the difference between skilled and unskilled task output.

9.3. Core Lab (20 minutes - Required)

9.3.1. Define the Skill

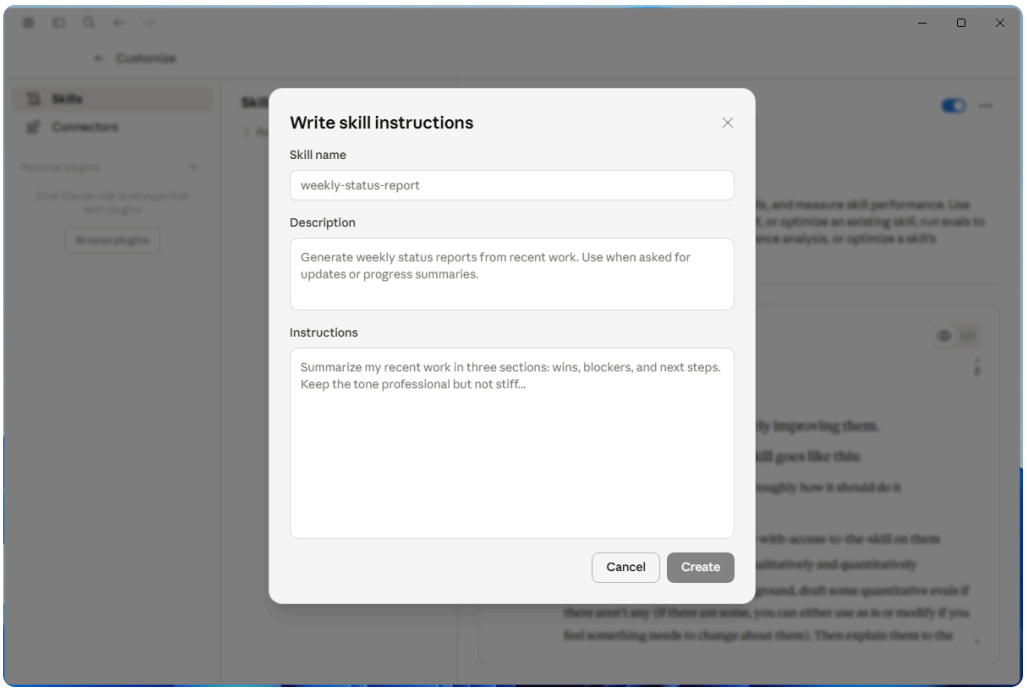
1. Decide what type of task you want to standardize. Choose one of the following tasks and descriptions, or adapt one to your work:
    - **Meeting Notes** — standardize how raw meeting notes are processed into a structured format
    - **Email Draft** — standardize the format and tone of outgoing professional emails
    - **Document Review** — standardize how feedback on a draft document is organized and presented

2. Write the skill's instructions. These should define exactly what Cowork must include in the output every time. For example, a **Meeting Notes** skill might specify:
    - List all attendees at the top

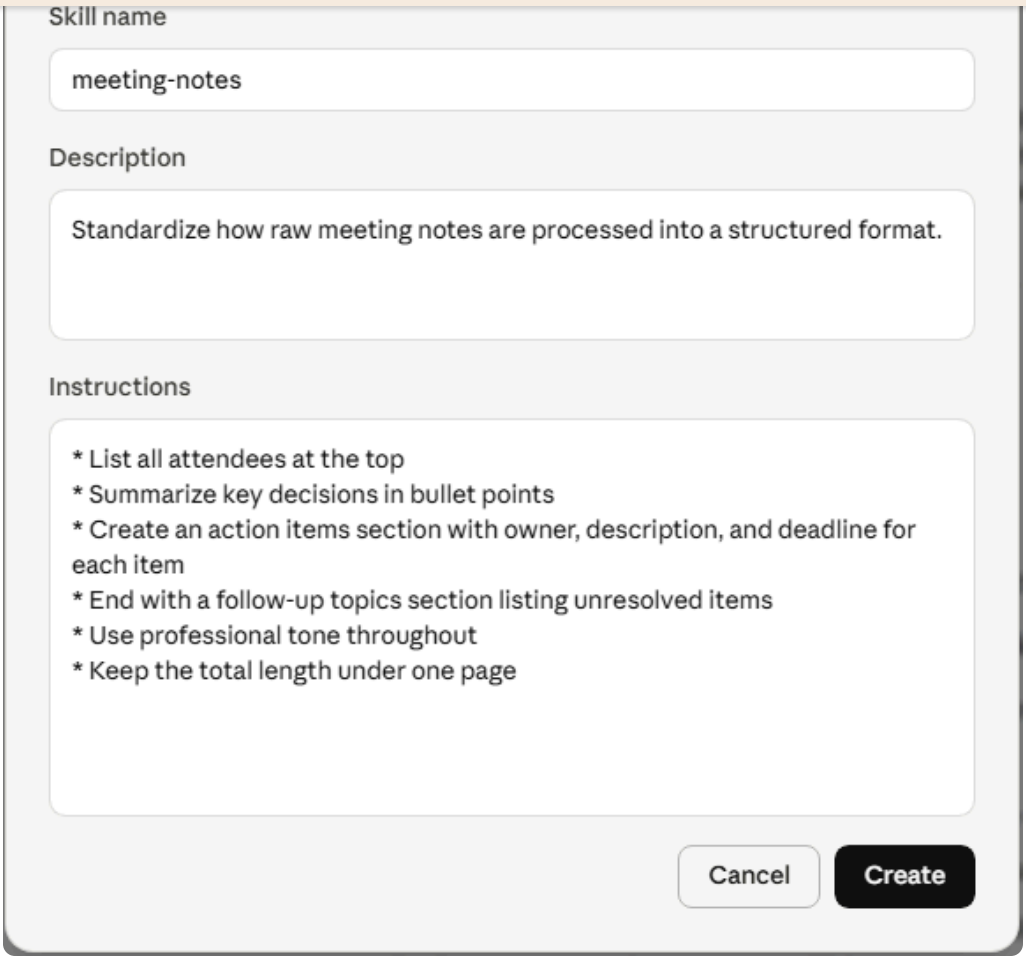
- Use professional tone throughout
- Keep the total length under one page

3. In Claude Desktop, click **Customize** in the sidebar.
4. Click **Skills** in the left sidebar.
5. Click the "+" button and select **Create New Skill**.
   A menu will appear, giving you a choice between **Create with Claude**, **Write Skill Instructions**, and **Upload a Skill**. Choose **Write Skill Instructions**.

   The skill creation interface opens, where you can teach Claude your processes and standards.



6. Create a new skill with the name (which must be lowercase and have no spaces), description, and instructions you defined.



7. Save the skill by clicking the **Create** button.
8. Confirm the skill appears in your **Skills** list on the Customize page.

9.3.2. Apply the Skill to a Task

1. Prepare a sample input for your skill. For a Meeting Notes skill, write or paste a set of rough meeting notes — informal, unstructured, the kind you would jot down during a real call.

   > **Tip**
   >
   > Use realistic input. If the sample is too clean or too short, the skill will not be meaningfully tested.

2. Submit a task that uses the skill. For example:
   "Process these meeting notes using the Meeting Notes skill."

   Include your sample input in the prompt or as an attached file.

- Are action items formatted with owner, description, and deadline?
- Are follow-up topics included?
- Does the tone and length match your requirements?

NOTE:If Claude wasn't able to use the skill for any reason, the output will include a message like "I wasn't able to apply the skill because...". If you see this, review your skill instructions and prompt to ensure they are clear and compatible and try restarting Claude Desktop and trying again.

9.3.3. Test Consistency

1. Prepare a second, different sample input for the same skill. Use different content — a different meeting, a different email, or a different document.
2. Submit the task again with the new input and the same skill.
3. Compare the two outputs side by side:
   - Is the structure identical across both outputs?
   - Are the required sections present in both?
   - Does the formatting remain consistent despite different input content?

9.3.4. Compare Against Unskilled Output

1. Write a generic prompt for the same task without referencing the skill — for example, "Summarize these meeting notes".
2. Submit the task using one of the previous inputs.
3. Compare the unskilled output to the skilled output:
   - What sections are missing from the unskilled version?
   - How does the structure differ?
   - Is the tone consistent?
   - Would you need to manually add or reorganize information in the unskilled version?

> ⓘ **Note**
>
> This comparison demonstrates the value of skills. The difference between skilled and unskilled output is the work that the skill saves you every time you run the task.

9.4. Challenge Lab (10 minutes - Optional)

1. Add a new requirement to your skill. For example:
    - **Meeting Notes:** "Include a 2-sentence executive summary at the very top, before the attendees list."
    - **Email Draft:** "Add a one-line subject line suggestion above the email body."
    - **Document Review:** "Begin with an overall assessment rating: Strong, Adequate, or Needs Revision."
2. Save the updated skill.
3. Run the task again with one of the previous inputs and verify the new requirement appears in the output.
4. Consider how this skill could be shared across a team:
    - If every team member used the same Meeting Notes skill, how would that affect consistency across the organization?
    - What would need to change in the skill instructions to work for a different team's meetings?
    - How would you handle a situation where different teams need slight variations of the same skill?

9.4.2. Challenge 2: Create a Versioned Skill Strategy

Design a versioning strategy so your team can evolve shared skills without breaking existing workflows.

1. Define a naming convention for skill versions (for example, `meeting-notes-v1`, `meeting-notes-v2`) and when a new version should be created.
2. Identify three change types and classify them as minor, major, or urgent.
3. Write a short rollout plan that includes:
    - How changes are tested before release
    - How teammates are notified
    - How older versions are retired
4. Create one new version of your skill and run the same input through both versions. Note the output differences and which version you would recommend for default use.

9.5. Review

- Skills standardize output format and quality by encoding your instructions into a reusable definition.
- Consistent instructions produce consistent results — applying the same skill to different inputs yields structurally identical output.

standards over time.