

Context Engineering Exercise

The Challenge You've been hired by an insurance company to design an AI capability that flags potentially fraudulent claims for human review.

You're not a claims adjuster. You've never processed a claim. But you need to define what "suspicious" looks like well enough for AI to help—without making the AI the final decision-maker on fraud.

Your Task: Design a Skill for an AI agent to follow

Using the **Claude Skills** format, draft a reusable specification. This is the actual format used by Claude Code. Work with your neighbor(s). Fill in what you can. Mark what you're uncertain about.

name: claim-fraud-flagger

description: This skill should be used when _____

Purpose

What does this skill accomplish? (1-2 sentences)

When to Use

What triggers this skill? What types of claims should it analyze?

- _____
- _____

How It Works

What patterns suggest possible fraud? List 3 red flags.

1. _____
2. _____
3. _____

Human Handoff

When should AI stop and escalate to a human? What must it NEVER decide alone?

Escalate when: _____

Never: _____

Reflection

After the exercise, consider:

Question	Your Response
How did you define "suspicious" without being a fraud expert?	
What was harder: the domain gap or the ethical constraints?	

The Point: You just worked through three stages:

Stage	What You Did	Skill Section
UNDERSTAND	Grasped a problem you're not expert in	Purpose, When to Use
DECIDE	Choose what to flag vs. escalate	Human Handoff
SPECIFY	Wrote it in a format AI could execute	How It Works

At every stage, you practiced **context engineering**—figuring out what information matters, how to structure it, what to include and exclude.

The format you used is real. Claude Skills (SKILL.md files) are how developers teach AI domain expertise. The sections—Purpose, When to Use, How It Works, Human Handoff—are the actual structure.

Business schools already teach understanding and decision-making. What's new:

- All three stages can be AI-enabled
- The output of Stage 3 is a reusable capability that encodes business logic
- Context engineering is the meta-skill; Claude Skills are the artifact

Beyond SKILL.md: The Full Architecture

A Claude Skill is more than instructions—it's a **capability architecture**:

```
claim-fraud-flagger/
├── SKILL.md                         ← Instructions (what you just wrote)
├── references/                       ← Domain knowledge loaded into context
│   └── fraud-patterns.md           (e.g., common fraud indicators by claim
type)
├── scripts/                          ← Deterministic steps (no AI judgment
needed)
│   └── calculate-risk-score.py    (e.g., rule-based scoring algorithm)
└── output-style.md                  ← How AI formats its responses
```

What else would YOUR skill need?

Component	Your Design
-----------	-------------

References — domain docs AI can consult

Scripts — deterministic logic, no judgment

Output format — how results should look

Your Next Step: Become a builder

You just did in 15 minutes what most faculty haven't tried: you designed an AI capability.

The best way to teach context engineering is to practice it yourself.

What could you build for YOUR domain?

If You Teach... You Could Build...

Accounting A skill that flags unusual journal entries

If You Teach... You Could Build...

Marketing	A skill that analyzes campaign copy for brand consistency
Finance	A skill that stress-tests DCF assumptions
Operations	A skill that identifies supply chain bottlenecks
Strategy	A skill that maps competitive positioning

Let's Continue the Conversation

Vishal Sachdev University of Illinois at Urbana-Champaign Gies College of Business

- GitHub: github.com/vishalsachdev
- Substack: chatwithgpt.substack.com
- Claude Skills Library: github.com/vishalsachdev/clause-skills

What will YOU build? I'd love to learn from you.

ICIS 2025 | Created: 2025-12-15