

1. COURSE INTRODUCTION

MGMT 638, Fall 2026

Kerry Back

Tue Aug 25, 2026

Course Overview

THE TOOL

Generative AI as it is actually used at work: to write and run code.

THE SUBJECT

Quantitative investing.
Predicting stock returns, and backtesting strategies.

GEN AI SIGNALS

Generative AI + news → trading signals

COURSE MECHANICS

COMPONENT	WEIGHT
Group presentations	60%
Class participation	40%

Group size = 2 or 3. Each group presents twice. Three presentations each class session beginning next week

Class participation includes cold calls and share screen.

Course Grades

An AI Environment for the Course

What runs

Claude Code,
running GLM-4.7.

What is installed

Python, JavaScript,
the analysis
libraries, and a set
of Claude Code
plugins.

What you see

A file browser, a
terminal, and a
browser view for
anything you build.

All of it runs in the cloud and is billed to the school.

Sign In

- 1 Go to `lab638.kerryback.com` — any browser, nothing to install
- 2 Username is your netid, the part before `@rice.edu`
- 3 Password is your student number, including the `S0`

First load takes a moment while your machine starts. If you are asked to sign in again later in the term, nothing is wrong — sessions do not survive a restart of the server.

Right panel

Claude Code. The prompt window is at the bottom.

Left panel

File browser by default. Terminal and View are there too — ignore them today.

Three dots, upper left

Upload from your laptop, download to it.

The Workspace

Things to Remember

Refresh

The file browser does not update itself. When GLM makes a file and you cannot see it, hit Refresh.

No Office

Double-clicking a `.xlsx` or `.pptx` does nothing. Office is not installed.

Download to view

Bring Office documents to your laptop and open them there.

EFFECTIVE PROMPTING

Most of the prompt engineering advice from a few years ago is now either useless or actively harmful.

Say what you want, not how to do it

“I want to compare revenue across countries — how should we show it?”

beats

“Make a bar chart, sorted descending, gridlines off, 12pt

If it does not work, try again

Rephrasing is cheap. Treat it as an infinitely patient colleague with a poor memory.

Good Enough Prompting

Plan Before You Act

Say: plan how to do this before you start.

WHY IT WORKS — KNOWLEDGE

Planning pulls the relevant considerations into the conversation. It then answers in light of what it just retrieved.

WHY IT WORKS — SUBTASKS

It breaks the work into steps you can check one at a time. Failures stay small and local instead of surfacing at the very end.

Confirm It Understood

Ambiguity does not produce a question. It produces a confident answer to a question you did not ask.

TELL GLM, BEFORE IT STARTS

Tell me what you are about to do and what you are assuming.

Correct it in one sentence, then say go ahead. This is the cheapest habit in the course and it will save you the most time.

It Is a Prediction Machine, Not a Colleague

Suppose it told you last week that method A beats method B. Today you ask it to do the task.

WHAT IT WILL DO

If “A beats B” is not in this conversation, and B appeared more often in its training data, it will probably use B.

WHY

Knowledge in the weights is not knowledge in use. Something has to pull it into the conversation before it can act on it.

Prediction also means momentum. Once it has started down a path, continuing that path is the likeliest next step.

You will watch it hit a roadblock, find a way around, hit another, find another way around — tenaciously, exhaustively. What it rarely does is go back and ask whether a different path was better all along.

Restate the goal, not the obstacle. That gives it permission to abandon the path instead of defending it.

Tunnel Vision

What Not to Do

SKIP	WHY
“Think step by step”	Reasoning models already do, in hidden tokens
“Take a deep breath”	Measurably useless now
“CRITICAL! NEVER EVER ...”	Overtriggers, and gets worse results
Stacked personas	“You are an expert X” can help tone; three of them does not

Direct task framing wins. Say what you want, say what done looks like, say what to avoid.

AI AND CODING

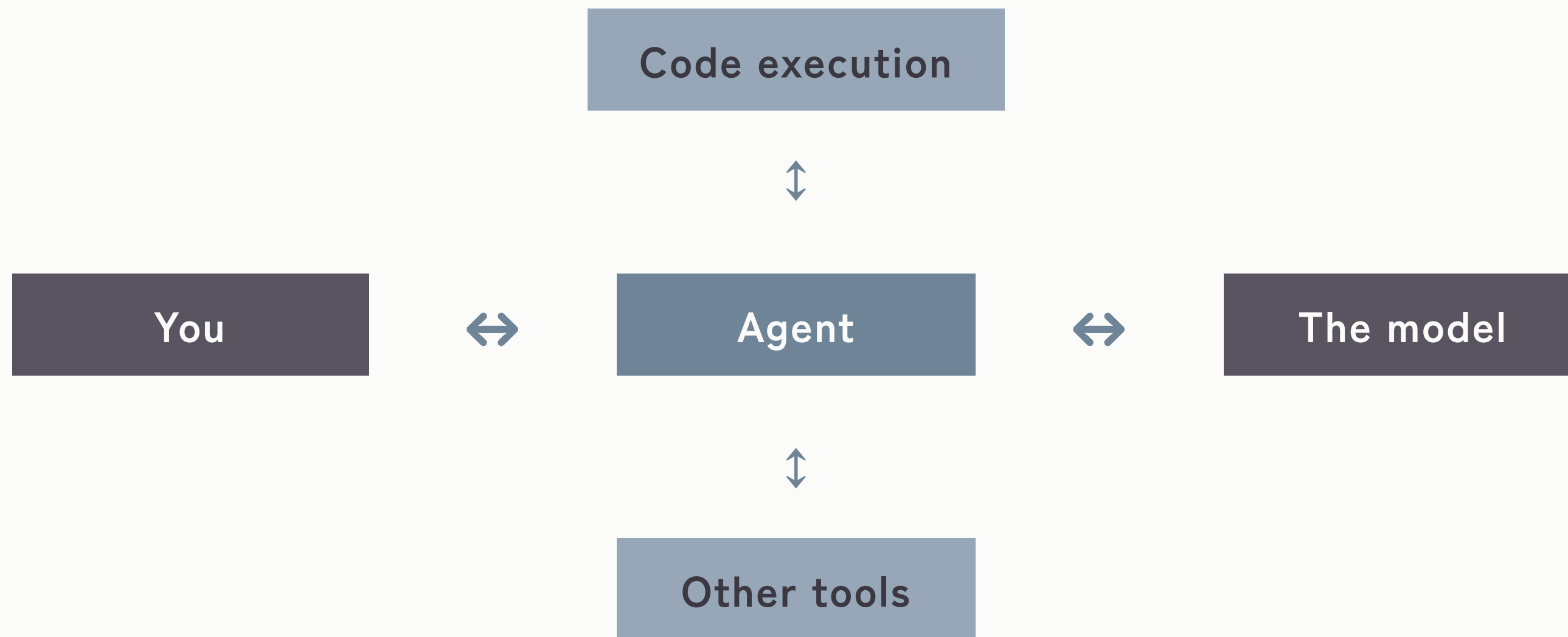


The model reads text and writes text. That is the entire capability.

It cannot open a file, run a calculation, look up today's price, or check whether what it just told you is true.

Anatomy of a Chatbot

An Agent Is a Chatbot with Tools



The model never runs any of it — it only emits requests.

- 1 The agent sends the prompt, the history, and a list of tools it has
- 2 The model answers: use tool X, with these arguments
- 3 The agent runs tool X and reports the result back, with instructions for reading it
- 4 The model decides what to do next — another tool, or an answer for you

The model emits requests; the agent executes them. Those step-3 instructions are the tool harness, and some are far better than others.

The Loop, Step by Step

Writing and Running Code

Almost everything useful the model does, it does by writing code and running it, including terminal commands.

On our AI lab, GLM will write code, save it to a file (a tool use), then run it with a terminal command (another tool use), read the output (another tool use), and revise and repeat as necessary.

All of this is orchestrated by the Claude Code “harness.”

What Code Produces

Excel workbooks
with live formulas

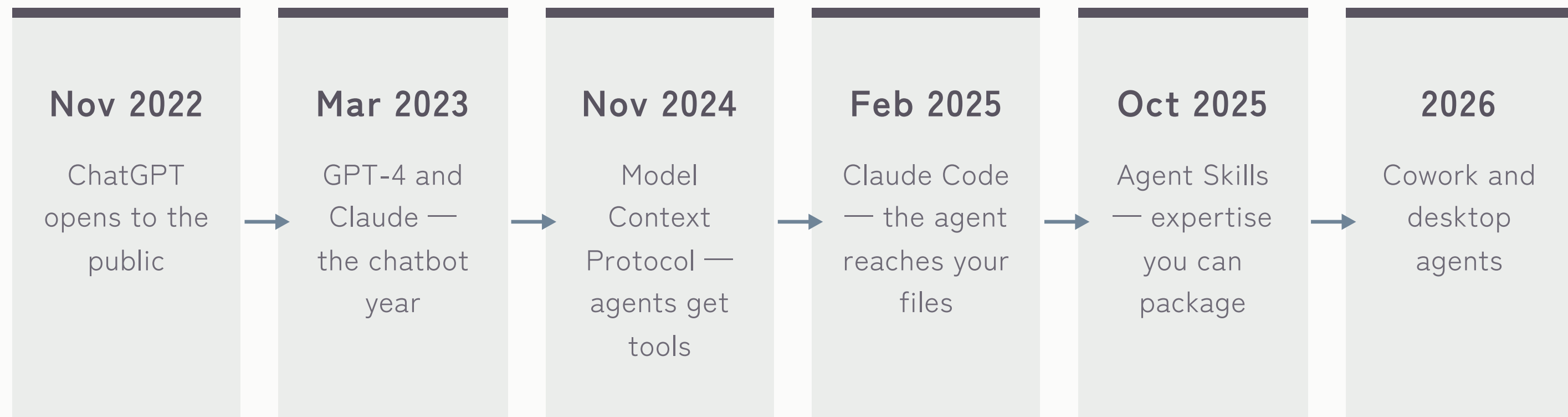
Charts and figures

Word and
PowerPoint files

Data analysis and
statistics

Web apps and
dashboards

A Brief Timeline



OPENAI CO-FOUNDER, FORMER TESLA AI CHIEF, NOW AT ANTHROPIC

October 2025

“They just don’t work. The industry is trying to pretend like this is amazing, and it’s not. It’s slop.”

December 2025

“I really am mostly programming in English now, a bit sheepishly telling the LLM what code to write ... in words.”

February 2026

“Coding agents basically didn’t work before December. I don’t think I’ve typed like a line of code probably since December.”

Andrej Karpathy

GIT AND GITHUB

Git Is a Version Control System

A commit is a snapshot of the whole folder, dated and labelled. Nothing is ever overwritten — it is added to.

Every commit is a point you can go back to, and Git can tell you what changed between any two of them, in which file, and on which line.

Git is not an AI tool. It is twenty years older than any of this. GLM simply knows how to drive it.

Git vs GitHub

GIT

On your machine. The history lives in the folder itself — no network, no account, yours alone.

GITHUB

A hosted copy of that same history for backup and sharing.

Create an Account and Log In from the Lab

- 1 Create a free account at `github.com`, and note your username
- 2 Open the Terminal tab, not the GLM pane — this step waits for you to type
- 3 Run `gh auth login` and answer: GitHub.com, HTTPS, yes to authenticating Git, browser
- 4 It prints a code — open the link on your laptop and paste it there

One Person in Each Group Does This

TELL GLM — ONE PERSON IN THE GROUP, ONCE

- Make a folder in my workspace called [whatever you want to call it] and make it a git repository.
- Add a .gitignore for csv, parquet, and .env files.
- Commit, then create a private GitHub repo of the same name and push to it.

Then Add Your Groupmates

ON GITHUB.COM, IN THE NEW REPOSITORY

- 1 Settings, then Collaborators, then Add people
- 2 Search each groupmate by their GitHub username
- 3 They get an email invitation, and have to accept it

TELL GLM

Clone the GitHub repo [username]/[repo name] into my workspace.

username should be the username of the group member who created it. repo name is the name they used for the folder

Everyone Else Joins

Later: Push and Pull

TELL GLM, AT THE START OF EVERY SESSION

Pull the latest from GitHub before we start.

THEN, AT THE END OF THE SESSION

Commit what we did and push it.

When Git Pushes Back

WHAT YOU SEE

`rejected --- non-fast-forward`

WHAT TO SAY

Someone pushed while you were working. “Pull their work, merge it, then push.”

`CONFLICT: automatic merge failed`

→ You both edited the same lines. “Show me both versions and wait for me to choose.”

Refuse these if asked: `git push --force` and `git reset --hard`