AI Coding Assistants at Wharton: Safe Use Guidance

Last Modified on 08/05/2026 3:52 pm EDT

This article explains how to use **AI coding assistants** safely at Wharton. These tools work inside a project or workspace and may read files, draft and edit work, inspect repositories, and run commands when permission is granted. Wharton and Penn offer tools including **OpenAI Codex** and **Anthropic Claude Code**. This guidance also applies to similar tools, regardless of vendor, interface, or programming language.

You do not need to be a programmer to use these tools or benefit from this guidance. It is intended for students, faculty, researchers, and staff.

The guiding principle is **appropriate reliance**: use these tools while retaining responsibility for their work. A coding assistant can sound polished and confident whether its output is correct or incorrect. You remain responsible for anything you submit, publish, approve, or run.

> **Looking for general AI guidance?**
> For approved services, confidential data, privacy, teaching, research, and authorship guidance, see Generative AI: Best Practices and Resources.

Frequently Asked Questions

What is an AI coding assistant?
It is an AI tool that works within a project or workspace. Depending on the access you provide, it may read files, edit files, inspect repositories, view command output, and run commands.

Can the assistant see my files?
Potentially. It can access files, folders, terminal output, and other information exposed through its workspace and permissions. Use a small, clean project folder rather than a home directory, desktop, or mixed workspace.

What data may I use with a coding assistant?
Use public data, Low data, approved synthetic data, or code that handles Moderate non PII and non FERPA data. Do not place the Moderate data itself in the workspace. For broader data requirements, see Security and Privacy Requirements.

What should I never expose?
Never expose passwords, API keys, tokens, cookies, private keys, login files, PII, FERPA data, High Risk Data, sensitive institutional information, or unauthorized production access.

Can I let the assistant run commands?
Yes, but begin with read only or approval required access. Approve only commands you understand, review every change, and do not allow commands to target production systems.

Can I use a personal Codex, Claude Code, or GitHub Copilot account?
Do not use a personal account for nonpublic Wharton code or data unless it has been separately approved. Use the Penn or Wharton managed access path designated for the tool.

Can the assistant connect to another Wharton system or data source?
Only after review. Connections to University managed applications, servers, endpoints, repositories, APIs, or data sources require an approved enterprise tool and review by Wharton Computing and the Wharton Information Security Office.

Can I use a coding assistant for coursework or research?
For coursework, follow your instructor's AI policy and disclose use when required. For research, keep the work reproducible, record material changes made by the assistant, and confirm that applicable IRB, data use agreement, contract, and research terms permit its use.

Who is responsible for an assistant's actions?
You are responsible for actions taken on your behalf and for anything you submit, publish, approve, or run. Limit access, monitor activity, review changes, and independently verify important results.

How to Obtain Access

Use the Penn or Wharton managed access path designated for the coding assistant. Access requirements and supported services may change, so consult AI Tools and Resources or contact Wharton Computing for current instructions.

Access does not expand the approved use of a tool. Continue to follow the data, workspace, permission, and review requirements on this page after access has been granted.

Notes by Audience

Audience	What to focus on
Students, including undergraduate and MBA students	Integrity and learning. Follow each instructor's AI policy, disclose use when required, and do not allow the tool to replace the learning an assignment is intended to measure.
Faculty	Course design and assessment. State what use is permitted, assume generated work may be fluent regardless of understanding, and design assessments that measure the intended knowledge or skill.
Staff	Operations and data handling. Data requirements apply to every file, command result, log, and connected resource exposed to the assistant. Keep operational records, credentials, and administrative access out of coding assistant workspaces.

Audience	What to focus on
Doctoral students and researchers	Research and reproducibility. Record material changes made by the tool, keep analyses reproducible without it, and confirm IRB, data use agreement, contract, and research terms before it accesses research data.

What an AI Coding Assistant Is

An AI coding assistant is different from ordinary chat. In chat, the main risk is usually what someone types, pastes, uploads, or connects. With a coding assistant, the risk also includes **what the workspace exposes** and **what the tool is permitted to do**.

Capability	What it can do	Risk to manage
Learn	Explain code, tools, errors, command output, documentation, and workflows.	Keep all source material appropriate for the service.
Create	Draft documents, scripts, tests, slides, examples, and code changes.	Review and verify generated work before relying on it.
Act	With permission, read files, edit files, install software, or run commands.	Limit permissions and approve actions deliberately.

Where It Shows Up

AI coding assistants appear in several forms. The names and capabilities differ by vendor, but the same safety principles apply.

Surface	Common uses	Safer starting point

Surface	Common uses	Safer starting point
Command line, such as Codex CLI or Claude Code	Learning a codebase, writing examples, running tests, or making focused changes within a selected folder.	Start in a clean project folder. Use read only access when exploring unfamiliar material.
Editor or IDE	Explaining selected files, suggesting changes, revising documentation, and reviewing code within the editor.	Confirm the active project and close files that should not be exposed.
Desktop or chat application	Working with a selected local project, repository, or separate Git worktree.	Treat the selected project as the boundary. Do not use it as a bridge into Penn or Wharton production systems.

Risks to Know

The primary risks are exposing information the assistant should not see, granting more authority than it needs, and relying on output that has not been verified. Coding assistants also create several risks that are less common in ordinary chat.

Prompt Injection

Content the assistant reads may act like instructions. A downloaded repository, shared notebook, package README, issue description, or web page may contain text intended to manipulate the assistant. For example, content could direct it to ignore prior instructions, run an unrelated command, or send information to an external service.

The assistant may not reliably distinguish your instructions from instructions embedded in the material it is processing.

- **Use projects from trusted sources.** Treat an unfamiliar repository or shared folder with the same caution as an unexpected attachment.
- **Limit network access.** Allow access only to destinations required for the task.
- **Watch for unexpected behavior.** An unrequested command, network connection, file access, or sudden change of task is a reason to stop.
- **Keep sensitive information out of the workspace.** This limits the harm that an injected instruction could cause.

Dependency Risk

A coding assistant may recommend installing software packages or dependencies. Malicious packages and packages with misleading names exist. Review each proposed installation, confirm the package name and source,

and avoid installing software you do not understand.

History and Log Risk

Prompts, terminal output, command histories, session logs, and generated files may persist after the task is complete. These records inherit the sensitivity of the information that passed through them. Consider logs and history when preparing and cleaning a workspace.

Use, Use Care, Do Not Use

Good Fit

- Explaining code, data handling logic, errors, or unfamiliar tools.
- Drafting documentation, examples, checklists, tests, or training materials.
- Making small changes in a clean and reviewable project folder.
- Working with public data, Low data, or approved synthetic data.

Use Care

- Running commands, installing packages, or changing dependencies.
- Using network access, browser access, plugins, or connected tools.
- Working with code that handles Moderate non PII and non FERPA data.

Code that handles Moderate non PII and non FERPA data represents the outer boundary of approved coding assistant use. Keep the actual Moderate data out of the workspace, develop against synthetic or sample records, and have a qualified person review the code before it runs against real data.

Do Not Use a Coding Assistant For

- Passwords, API keys, access tokens, cookies, SSO codes, private keys, credentials, or login configuration files.
- PII, FERPA data, or High Risk Data.
- Nonpublic information about identifiable people.
- Production University systems, administrative consoles, identity infrastructure, billing systems, student systems, or regulated workflows.
- Unattended actions, privileged changes, payment approvals, record changes, or other actions that a qualified person cannot review.
- Mixed workspaces such as home directories, browser profiles, desktops, or folders that combine approved content with personal or restricted information.

Stop and ask for guidance
Do not proceed when a task involves PII, FERPA data, High Risk Data, credentials, production systems, contractually restricted data, or work that a qualified person cannot review. Contact the Wharton Information Security Office.

Before You Start: Data and Permission Boundaries

Before granting access or beginning work, identify what the assistant can reach and choose the narrowest permissions that will complete the task.

Area	Pass condition	If the condition is not met
Data	The workspace contains public data, Low data, approved synthetic data, or code that handles Moderate non PII and non FERPA data. It contains no PII, FERPA data, or High Risk Data.	Stop. Classify the data and contact the Wharton Information Security Office when regulated data, a contract, a data use agreement, or a University system is involved.
Workspace	The assistant can reach only the files and folders required for the task.	Create a clean project copy, use a separate worktree, remove unrelated files, or switch to read only access.
Commands	Commands require approval, are understood by the user, and operate only within the intended environment.	Do not approve the command. Ask for an explanation or complete the action manually.
Network	Network access is limited to specific trusted services required for the task.	Disable network access or restrict it before continuing.
Secrets	No credentials appear in files, prompts, clipboard contents, environment variables, terminal output, logs, or session history.	Remove the credentials. Rotate any credential that may have been exposed and restart from a clean environment.
Evidence	Changes, tests, commands, citations, and results can be reviewed independently.	Do not rely on the result until appropriate testing, source review, or subject matter review is available.

Recommended permission sequence
Begin with view only or read only access. Permit edits only after the workspace is clean, the task is clear, and changes can be reviewed. Use broad access only in a disposable test environment containing Low data.

Appropriate Reliance

Safe use is not limited to permissions and data. It also requires deciding how much to trust the output.

Human automation research describes the goal as **appropriate reliance**: trusting a tool in proportion to its

demonstrated performance on the task being completed, rather than according to how confident or polished it sounds.

Two failure modes matter:

- **Overreliance:** Accepting output because it is fluent, fast, or usually correct. A plausible script, formula, citation, or explanation may still contain serious errors.
- **Underreliance:** Refusing to use a capable tool for low risk work that it performs well. The alternative to excessive trust is calibrated trust, not zero trust.

Use these practices to calibrate reliance:

- **Trust by task, not by product.** A tool that performs well when explaining an error may perform poorly when evaluating statistics, security, or licensing.
- **Verify through an independent route.** Run the code, inspect the source, open the cited document, reproduce the calculation, or consult a qualified person.
- **Start with small tasks.** Begin where mistakes are inexpensive and visible. Expand the scope only after the tool demonstrates reliable performance in your context.
- **Guard against automation bias.** When a tool is usually correct, reviewers may begin to skim. Continue to inspect changed files, commands, tests, and sources.
- **Do not rely on work you cannot evaluate.** When neither you nor another qualified person can review the result, do not submit, publish, or run it.

Reference: Lee, J.D., & See, K.A. (2004). Trust in automation: Designing for appropriate reliance. Human Factors, 46(1), 50–80.

Review Habit

- Start with one narrow folder, one clear task, no production credentials, and a specific definition of completion.
- Ask the assistant to summarize what it changed, what commands it ran, what passed or failed, and which sources it used.
- Review every changed file, new dependency, command approval, generated citation, and external connection.
- Run tests and checks independently rather than relying only on the assistant's report.
- Stop for normal security, privacy, research, and IT review whenever the work touches protected data or Penn systems.

Source Links

Wharton and Penn Guidance

- Generative AI: Best Practices and Resources
- AI Tools and Resources
- Wharton Data Classification and Management Standard
- Penn ISC Guidance for Using Generative AI at Penn
- Penn Data Risk Classification
- Penn Generative AI Tools and Resources
- Penn ChatGPT Edu FAQ

Vendor Documentation

- OpenAI Codex Agent Approvals and Security
- OpenAI Codex Documentation
- Claude Code Security Documentation
- Claude Code Documentation

Training

- AI Fluency for Students, a free self paced course about collaborating with AI responsibly.

Research

- Lee and See (2004), Trust in Automation: Designing for Appropriate Reliance

Questions?

Contact your Wharton Computing representative with questions about coding assistants, access, or whether a task fits within this guidance.

For questions involving data classification, protected information, University systems, contracts, or security requirements, contact the Wharton Information Security Office before beginning the work.