

Vibe Coding from Classroom to Industry

AI Utilization in Higher Education Settings for Students

DD

Dr. Dewayne A. Dixon

Faculty Fellow | AUC DSI

Visiting Assistant Professor, Morehouse College

CB

Prof. Cory Brooks

ex SWE (Big Tech, Wall Street)

CodePath Instructor, Morehouse College

🎯 Workshop Focus

● Why AI matters now

Data-driven urgency, adoption stats, trust paradox

● Hands-on Agent Building

Create personal study agents

● Industry coding workflows

Real-world development practices

Workshop roadmap

Total duration: 110–120 minutes

Today's journey from theory to practice



Audience check-in

Quick poll: where are you today with AI?

Raise your hand for the option that best describes your current usage.

1

Never used

I haven't tried AI tools yet, or I have not found the time.

2

Dabbled a bit

I have used ChatGPT once or twice, but it is not part of my routine.

3

Use weekly

I use AI for a few targeted tasks such as drafting, outlining, or feedback.

4

Use daily

AI is integrated into my workflow and changes how I work.

Please raise your hand when your option is called.

AI is already part of real-world tech jobs

This is the environment you're being prepared for

84%

Developers use or plan to use AI tools

Source: Stack Overflow 2025

51%

Professional developers use AI daily

Source: Stack Overflow 2025

91%

Adoption in active developer cohorts

Source: DX sample (135k+ devs)

What this means for you

When you graduate, you won't be asked whether you've used AI. You'll be expected to know how to use it responsibly, critically, and effectively. This workshop helps you build those skills, on purpose.

AI tools are now industry infrastructure

Not a trend — a baseline expectation

~90%

Fortune 500 adoption

Microsoft FY26 Q1 Earnings

20M+

**Total GitHub Copilot
users**

Mid-2025 user base

4.7M

**Paid subscribers (75%
YoY)**

Jan 2026

Why this matters: AI tools shape how work gets done, not just how fast. Knowing how to collaborate with AI — and catch mistakes — matters more than speed alone.

AI now generates a lot of code — but humans still decide

Practice collaborating with AI-generated output

22%

Merged code that is AI-authored

Source: DX Analytics

26.9%

AI-authored code of all production code

Source: DX Research, Q1 2026, 4.2M developers

Important skill shift

Your job is not just to produce code. Your job is to read, test, debug, and improve code — including code written by AI. This workshop treats AI output as a draft, not an answer.

AI can speed you up — if you stay in control

The upside is real, but it requires judgment and review

3.6h/week

Average time saved

Source: DX Analytics 2025

+60%

PR volume increase

Source: Daily AI users vs light users

10–30%

Overall productivity lift

Source: Across varied tasks

Caution

Teams that use AI consistently often ship more, but gains depend on task type, developer skill, and verification habits. Mistakes happen when verification is skipped.

The trust paradox: high usage does not mean high trust

Adoption is high, but confidence remains limited

1.7x

More issues in unverified AI code

Source: CodeRabbit Dec 2025

75%

Developers still turn to humans when they doubt AI

Source: Stack Overflow 2025

Trust Breakdown

46%

Distrust or skeptical

33% Trust

3% Highly trust

Critical takeaway

AI literacy does not mean blind trust. It means better prompting, stronger review, and routine verification.

AI literacy affects your career

Institutional competitiveness increasingly depends on AI literacy

70%

CS teachers already include AI topics

Source: 2025 teacher landscape

Hiring reality

Employers increasingly expect AI literacy across the full development lifecycle, not just in isolated tasks.

Funding pressures

Graduate outcomes are being watched more closely, and AI skill-readiness is part of that story.

Big picture

Knowing how to work with AI, without relying on it blindly, is becoming part of professional competence.

Common questions students ask

Four concerns that usually come up first

1 **Academic integrity** | Is this just cheating?

2 **Skill loss** | Will AI make me worse at problem-solving?

3 **Accuracy** | What if the AI is wrong?

4 **Access and cost** | What if I don't have premium tools?

Our approach

We won't ignore these concerns. We'll address each one with clear rules and practical safeguards.

How to use AI well

Focus on learning, not shortcuts

AI is part of the workflow

You will see AI integrated into real development today. This is normal in industry.

You are responsible

You must understand and review any AI-assisted work before using or submitting it.

We focus on reasoning

Reasoning, verification, and decision-making matter more than perfection.

Rule of thumb

If you can explain it, you are using it well. Next up: the **Use • Verify • Explain** framework.

Your student framework: Use • Verify • Explain

The three habits that make AI assistance productive and honest

USE

Use AI for brainstorming, outlining, debugging hints, draft ideas, and practice help.

AI is a starting point, not the finish line.

VERIFY

Check outputs with tests, tracing, examples, source checks, and human review.

Do not blindly accept output.

EXPLAIN

Be able to explain final logic, decisions, and corrections in your own words.

If you cannot explain it, keep working to understand the given code.

AI is a supervised assistant, not an authority. **Your thinking is still the product.**

Smart, allowed uses of AI in your courses

Practical ways you can use AI right away

- **Brainstorming and outlines** Generate study plans, concept maps, topic outlines, and project ideas.
- **Personal study aid** Clarify criteria, get first-pass feedback on drafts, review your own notes.
- **Exam practice problems** Generate varied practice questions, coding challenges, and edge cases.
- **Scientific writing** Draft proposals, summaries, and alternative abstracts — then revise and verify.
- **Explore alternate approaches** Find alternate solutions, identify misconceptions, and test your reasoning.

Next step: We will practice these workflows together step-by-step.

AI use zones: know your boundaries

A quick reference for safer, cautious, and risky AI use

✓ GREEN — Safer

- Brainstorming
- Practice questions
- Concept explanations
- Debugging hints
- Test generation
- Outline drafting

⚠ YELLOW — Use with caution

- Code generation
- Custom GPTs / agents
- Citation suggestions
- Study agents
- Draft writing
- Generated study materials


⛔ RED — Do not do

- Submit AI output unchanged
- Use code you can't explain
- Upload private info to public tools
- Ask AI to complete graded work
- Merge agent code without review

Privacy and custom tool awareness

Be smart, not scared — but know the boundaries

 **Know the boundaries** — Different AI tools have different data policies. Read before you paste.

 **Agents and custom GPTs** — May connect to external apps or APIs. Review what permissions you are granting.

 **Protect your data** — Be cautious of sharing personal information, especially with the growing number of tools being released.

 **Use approved tools** — Prefer institution-approved or instructor-approved tools when possible.

This is not about fear. It is about making informed choices with your information.

05:00

Stretch, hydrate, and get ready for the coding segment

Setup checklist

Install VS Code if needed

Sign in to GitHub or create a free account

Install the Copilot Extension in VS Code

code.visualstudio.com • github.com

Presenters are circulating — raise your hand if you need help.

ADMI 2026 WORKSHOP

Segment 2

Industry vibe coding

Bridging classroom theory and modern software engineering practice with live demos.

We are moving from insight to practice — building a personal artifact you keep after today.

Term of the day: Vibe Coding

A useful definition plus the classroom implication

Definition

The process of generating software primarily through iterative LLM prompting and review rather than purely manual line-by-line coding.

Industry reality

Professional developers work hybrid: AI for acceleration, humans for judgment and review.

The focus is productivity and speed.

Student reality

Students with strong AI literacy outpace those without it.

The focus is knowledge transfer through determined pedagogy & policy.

Guardrail

AI can scaffold and accelerate, but it never replaces understanding. If you cannot explain the code, you do not want to claim it as your own.

Passing the interview: U.M.P.I.R.E. framework

Interview prep strategies are totally up to the student, but structure still needs to be implemented when using AI for interview prep.

Understand

Restate problem, constraints, I/O.

Match

Identify patterns, data structures, algorithms.

Plan

Pseudocode before asking AI for syntax.

Implement

AI for boilerplate; keep authorship.

Review

Run tests. Ask "why did this fail?" not "fix it."

Explain

Defend logic in your own words.

Software development lifecycle (industry)

AI is not replacing the process — it is embedded across it. Here are some examples of where AI is used throughout the SDLC (Software Development Lifecycle).

1 Requirements

Define goals, clarify specs, generate user stories.

Requires human negotiation

2 Architecture

Propose patterns, analyze trade-offs, create blueprint.

Involves team conversations

3 Implementation

Write actual code. AI scaffolds — you must understand.

4 Documentation

Add docs for engineers and clients.

5 Testing

Unit tests and integration tests. Verify quality.

6 Deploy & Maintain

Release safely. Monitor health post-deploy.

Environment setup check

Making sure everyone can follow the live coding segment

VS Code installed

Confirm that Visual Studio Code launches correctly on your machine.

GitHub account ready

Sign in to GitHub and make sure you can access repositories.

Copilot extension

Install the GitHub Copilot extension within VS Code. You will not need a paid subscription for today.

If you are unable to install VS Code today, please complete the non-coding portions individually. You will be able to implement your project from the browser towards the end.

Copilot: What exactly is it?

The AI pair programmer reshaping daily development workflows

What it does

- Context-aware suggestions from your files
- Generate unit tests, documentation, boilerplate
- Chat-based explanation and refactoring
- Define agent instructions and use custom agents

20M+

Users (Mid-2025)

4.7M

Paid subs (Jan 2026)

~90%

Fortune 100 use

Critical caveat: AI output is a prediction, not a fact. It can hallucinate APIs, introduce security flaws, or suggest deprecated patterns. AI-generated code can be wrong, insecure, outdated, or misleading.

Best practice: Always review line by line, pair with tests, and start with a good prompt. Treat AI-generated code as a draft to inspect, not a finished answer to submit.

Prompt Engineering: Getting more out of AI

Specificity is the difference between toy output and production-ready output

✗ Weak prompt

"Make a portfolio website for me. I'm a CS student from Atlanta with a 3.5 GPA"

Likely result: basic design, hallucinated info, missing identity

✓ Strong prompt

"I am creating a customized portfolio. Review my design, then ask follow-up questions. [Design prompt from previous slide]. Please ask follow up questions to ensure the design is aligned with my intentions."

Result: tailored output you can verify and improve

Pro tip: Define requirements before asking for work. Convenience does not remove responsibility.

Prompt creation workflow (5min)

Think before we prompt. Let's put as much detail into our ideal product. We will use this in a few slides.

Workshop Goal:

Implement a personal portfolio website using Copilot. Type your responses of the below topics into a notes app / txt file.

1. Consider UI/UX questions:
 - a. Color palette?
 - b. Animations and layout?
 - c. Background images (need to be downloaded and saved locally).
2. What content do you want included? You might need to collect various PDFs or other files that you want to include.
 - a. Resume/Cover Letter PDF
 - b. LinkedIn URL
 - c. Samples of work that you have completed (photo, audio, etc)

1. Design — Plan your website mentally. Be explicit.

2. Draft Notes — Write specifications step by step.

3. Prompt — Translate notes into a solid prompt.

4. Review & Iterate — Have the LLM ask you follow-up questions.

Copilot Agent: Roles and Instructions

Users should define agent “roles” to fine tune their behavior. This is accomplished by providing **agent instructions**.

The task:

1. Open the Copilot Panel in VS Code (top right)
 2. Click settings icon → Generate Instructions → Instructions (Workspace)
 3. Paste your full prompt at the bottom of the file
 4. Add additional instructions and rules as needed
- Example Role: “You are a veteran web designer who focuses on showcasing individual strengths, tailored towards a technical reader”

1. Identify the “real world” persona

Think about who you would want to build this in real life.

2. Preventing “free reign”

Be specific about things that the agent absolutely can never do.

3. Lasting guardrails

Instruction files are read before every user prompt is executed. For example: “confirm that any photos found online are copyright friendly”

4. Paste your rules into the instructions file

Now you have a local file that you can edit at anytime for immediate behavioral changes that last.

Define rules based on the task at hand. Think about what persona would be most beneficial for the product.

Hands-on challenge

15:00 remaining

15 minutes to build using the supervised coding workflow

The assignment

Using the methods discussed today, work with Copilot to implement your website.

- Test out different models with each request (Claude, Sonnet, GPT, etc.)
- Always Start in **Plan** mode — research and outline before writing a line of code

Requirements

- The website launches and renders successfully
- You added one unique element that you came up with on your own
- You are proud of what you made today!

Tip: Use 'Restore Checkpoint' in Copilot chat if you need to go back. Do not blindly accept — test before trusting.

GitHub agents preview

Beyond VS Code: How to use Copilot from the browser.

Example workflow

- 1) Initialize a new Repository (enable the ReadMe option)
- 2) Open an issue with a clear request
- 3) Assign it to Copilot
- 4) Agent implements changes
- 5) Human reviews the branch before merging

High-value use cases

- Issue triage
- Boilerplate & scaffolding
- Test generation for existing code
- Organizing large scale projects that involve large features to be implemented in pieces

The human loop: By default, GitHub will not allow Copilot to merge changes without your approval.

Segment 3

Build your personal AI study agent

A personalized AI agent that helps you study, practice, review, and self-test

Your agent lives inside your chosen LLM (ChatGPT, Claude, Copilot, Gemini, etc.)

Strong students don't just ask questions — they design systems

How to use your study agent responsibly

✓ Allowed

- Explaining concepts
- Generating practice problems
- Quiz yourself
- Debugging after you attempt
- Summarizing notes
- Ask for hints before answers

✗ Not Allowed

- Submitting AI answers as your own
- Completing graded work
- Copy-paste without understanding
- Pasting full assignments
- Confusing studying with outsourcing

Golden rule: If you can't explain the answer out loud, you can't submit it. A good study agent helps you think more — it should not think instead of you.

Choose your LLM

Pick the tool that fits your task — and ask the right safety questions

GenSpark

Persistent agents with stable instructions

Best for: persistent agents

Gemini

Clear explanations, multimodal reasoning

Best for: concept exploration

ChatGPT / Claude

Fast drafting, prompt iteration

Best for: prompt iteration

Copilot

In-context coding inside VS Code

Best for: in-context coding

Before you choose, ask:

What task am I doing? Does this tool connect to outside apps? Is it safe to paste this info? Can I verify the output? Convenience does not remove responsibility.

Step 1. Set the agent's role

Tell the AI who it is and what it is not

Copy & adapt prompt

"You are my personal study agent for [COURSE NAME]. Your job is to help me understand concepts, practice problems, and review mistakes. You must explain reasoning step by step. You must NOT complete graded assignments for me."

Why this matters

Agents without roles default to answer machine. Clear roles = better explanations and safer use.

Refinement tips

Add course level.

Forbid final answers without explanation.

Revisit if agent does too much.

Step 2. Define your learning goals

Tell the agent what you want to improve

COPY & ADAPT PROMPT

"In this course, I struggle most with: [topic 1], [topic 2]. I want to get better at: solving problems on my own, understanding why solutions work."

Good goals are

Specific ('recursion base cases' > 'recursion').
Skill-focused, not grade-focused.

Refinement tips

Avoid vague goals. Update after exams. Add 'explain as if I'm new' if lost.

Your task: Describe topics you find difficult, skills to strengthen, where you get stuck.

Step 3. Set your teaching style

Tell the agent how to teach you

Copy & adapt prompt

"When you explain things: Start simple then add complexity, Use examples and analogies, Pause to ask me questions, Check my understanding before moving on."

Why this matters

AI defaults to dense explanations. Teaching style matching improves retention.

Refinement tips

If rushed, ask to slow down. Say 'ask me before explaining' for struggle-first. Change styles per topic.

You can ask for: step-by-step • visual/analogy-based • Socratic questioning • slower pacing

Step 4. Build practice problem behavior

Make the agent help you practice, not just read

Copy & adapt prompt

"Generate practice problems on [TOPIC]. Start easy and increase difficulty. Do not give full solutions unless I ask. Offer hints first, then explanations."

Student responsibility

Attempt the problem before asking for help. Struggle is part of learning.

Refinement tips

- Ask for exam-style or real-world problems.
- Request time/space constraints.
- Ask for multiple variations.

What to ask for: practice problems, gradual difficulty, edge cases, hints (not answers) first.

Step 5. Add self-testing mode

Turn the agent into a quiz partner

Copy & adapt prompt

"Quiz me on [TOPIC]. Ask one question at a time. After my answer, explain what I got right or wrong. Then ask a follow-up question."

Why this works

- Retrieval practice strengthens memory.
- Immediate feedback improves correction.

Refinement tips

- Mix old and new topics.
- Request explanations even when correct.
- Say "be strict like an exam grader" if needed.

Goal: Use AI to check understanding, not replace thinking.

Step 6. Use mistakes as learning data

Teach the agent how to respond when you are wrong

Copy & adapt prompt

"Explain why my answer is incorrect. Show the correct reasoning. Then give me a similar problem to retry."

Why this matters

Learning comes from debugging thinking.

Mistake analysis is more powerful than repetition.

Refinement tips

- Ask agent to highlight where reasoning failed
- Request a simpler version before retrying
- Keep a running list of common mistakes

Step 7. Create your final agent prompt

Lock in a reusable study agent

Copy & adapt prompt

"You are my personal study agent for [COURSE]. You help me understand concepts, generate practice problems, quiz me, and analyze mistakes. You explain reasoning step by step. You never complete graded assignments for me. If I am confused, slow down and clarify."

Save the output

Re-paste at the start of each study session. Update as your needs change.

Final reminder

A good agent is designed, not discovered. Your thinking is still the product.

Your task: Combine everything into one master instruction you can reuse each session.

Agent build debrief & reflection

Share outputs, compare prompts, and capture what worked

01 What worked (and why)?

Identify one agent instruction that noticeably improved your understanding.

02 Where did the agent struggle?

Describe one moment where the agent misunderstood you, gave too much help, or needed refinement.

03 How would you redesign it?

Name one change to your agent's role, rules, or prompts to improve it for future study.

Practical integrity strategies

Using AI without crossing the line

01

Make your thinking visible

Use your agent to explain, practice, and check understanding — but always submit work you can explain in your own words.

02

Separate practice from submission

AI is for studying, drafting, and feedback. Not for completing or submitting graded assignments on your behalf.

03

Design for verification

Ask for explanations, hints, and self-testing — not final answers. Verify everything you use.

Your thinking is the product. AI assistance must stay visible and explainable. Responsible AI use means learning with support, not avoiding effort.

Debrief & reflection

Move from passive observation to active strategy

01

What surprised you?

Reflect on AI capabilities, limitations, and moments when the tools behaved differently than expected.

02

Where did AI help or hinder?

Identify moments of acceleration versus friction or confusion in today's workflow.

03

What will you try next week?

Name your first small change: one study habit, one new tool, or one coding workflow improvement.

Key takeaways

Three things to remember from today

01

AI is now part of real workflows

Students graduating without AI collaboration skills face a real disadvantage. Integrate it now — responsibly and on purpose.

Action: start using AI with intention today.

02

Verification is part of the assignment

Do not just prompt. Test, review, and verify. Make verification a habit, not an afterthought.

Action: always test before trusting.

03

If you cannot explain it, you do not own it

AI-assisted work must be work you understand. Your thinking is still the product.

Action: explain every line you submit.

AI is useful, but only when you stay in control.

Resources & next steps

Keep the vibe going with tools, contacts, and sources

Contacts

Dr. Dewayne Dixon

ddixon@aucenter.edu

Prof. Cory Brooks

cory.brooks@morehouse.edu

Essential tools

GenSpark — persistent workflows

Gemini — AI tools

GitHub Copilot — industry assistant

VS Code Live Share — collaboration

Sources cited

Stack Overflow Developer Survey 2025

DX Analytics Report Q4 2025

CodeRabbit Analysis Dec 2025

Microsoft FY26 Q2 / Gartner 2025