

// AI-Assisted Programming

Module Introduction;

Two questions;

- ▶ Who used an AI tool to write code in the last week?
- ▶ Who could **explain every line** it gave you?
- ▶ `// the gap between those two answers is this module`

What this module is;

- ▶ Not "how to use Copilot" — tools change every few months
- ▶ How to **direct** an AI assistant, and how to **judge** what comes back
- ▶ Prompting, retrieval, protocols, agents, deployment, review
- ▶ **The uncomfortable part.** You are accountable for code you did not write. The assistant is fast; your name is on the commit.

Where this actually is, in 2026;

US developers using AI coding tools daily	92%
...who trust the code it produces	29%
...who always review it before committing	48%
Major issues vs human-written code	1.7×
AI samples with an OWASP Top-10 vulnerability	~45%

// they don't trust it – and they ship it anyway

Industry surveys of varying rigour: read the direction, not the decimals.

So – must you understand every line?;

- ▶ **No.** Almost nobody does, and pretending otherwise is dishonest
- ▶ But the review didn't disappear — it **moved**
- ▶ From *reading every line* → to *tests, types, and CI that must pass*
- ▶ From *"looks right"* → to *"prove it behaves right"*
- ▶ **The trade only works if the verification is real.** Skip the tests and you have not moved up a level — you have just stopped checking.

Predict: whose code is it?;

The assistant gives you a 40-line function. It is character-for-character identical to one in a well-known open-source project, released under a licence that requires anything built on it to be open-sourced too.

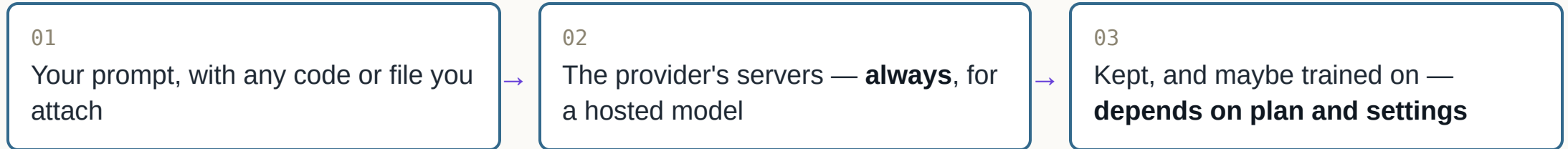
You paste it into your employer's closed-source product.

- ▶ It is yours — the tool wrote it, so no licence applies
- ▶ The licence may apply — it follows the code, not the typist
- ▶ Nobody can tell, so it does not matter

The licence follows the code;

- ▶ A licence attaches to the **code**, not to whoever typed it. "The tool wrote it" is not a defence anyone has established
- ▶ Word-for-word copies of well-known code are rare, but real. Most assistants can **block suggestions that match public code** — find out whether yours does
- ▶ The law on training data is still being argued in court. Do not build your habits on how it comes out
- ▶ Treat generated code as code of **unknown origin**: you can stand over it once you have read and tested it — or you cannot ship it.

Where what you paste goes;



- ▶ It is not a text editor. It is closer to **emailing the text to a company** — read the data setting on the plan you use
- ▶ Never in a prompt: **credentials**, other people's **personal data**, code you have **no right to share**. Whatever the plan says
- ▶ Employer code goes only into the tool the employer licensed. A local model keeps everything on your machine — at a cost in capability
- ▶ `// nothing in this module ever needs a secret in a prompt`

The words you'll hear this year;

- ▶ **Vibe coding** – prompt it, run it, ship it, barely read it
- ▶ **Comprehension debt** – the future cost of understanding code a machine wrote and nobody read
- ▶ **Haunted codebase** – a working system the team no longer understands
- ▶ **Context engineering** – the shift from *how you ask* to *what you put in front of the model*
- ▶ **Spec-driven development** – the backlash: write the spec, let the agent implement it
- ▶ *// half of these did not exist two years ago*

Module Delivery;

- ▶ Act 1 — how the module runs: schedule, assessment, effort
- ▶ Act 2 — the tools you need set up before next week

Duration and contact time;

- ▶ **12 teaching weeks**, plus a reading week
- ▶ Reading week is the October bank-holiday week — it sits between week 6 and MCQ 1
- ▶ Each week: **2 hour lecture + 2 hour lab**
- ▶ **No lab in week 1** — labs start next week
- ▶ The class is split into groups for labs; check your timetable

// your lab group and room are on your timetable

Enrol on the VLE;

1. Go to your college's **VLE** — the link is on your timetable
2. Search for **AI-Assisted Programming**
3. Find out which lab group you are in (A, B, C or D)
4. Click **Enrol** and use your group's enrolment password

Passwords are given out in this lecture, not published. If you miss them, email me — this deck is on a public site.

Module learning outcomes;

- ▶ **Identify and evaluate** AI-powered coding tools — generation, completion, debugging
- ▶ **Integrate** them into a real development workflow
- ▶ **Critically analyse** their limits: code quality, over-reliance, bias
- ▶ **Explore** emerging trends in the field

Assessment ;

Component	Weight	When
MCQ 1	32%	Week 7, in the lab
MCQ 2	32%	Week 12, in the lab
Practical Assessments	9 × 4%	One per week, open all that week

Nine small continuous assessments are 36% of the module. A missed one counts as zero, and each closes at the end of its week.

The two MCQs;

- ▶ Multiple choice, sat **in person** in the lab slot
- ▶ Drawn from the lectures **and the labs** of previous weeks
- ▶ MCQ 1 covers weeks 1–6 · MCQ 2 covers weeks 8–11
- ▶ **To practise:** use the practice app on the module site, or upload the week's material to a tool like NotebookLM and ask it to generate questions.

The Practical Assessments (PAs);

- ▶ **One per week**, on the VLE, worth **4%** each
- ▶ Open **all week** — do it when it suits you
- ▶ You **may** use AI tools
- ▶ // the first one opens week 2

Tools you need;

Tool	What for
GitHub	Where your work lives
Codespaces	A full dev environment in the browser
GitHub Copilot	AI assistance inside the editor
CLI coding agents	Covered in week 9

// nothing to install – Codespaces runs in a browser

Before next week;

- ▶ Sign up for the **GitHub Student Developer Pack** — it is free and unlocks Copilot
- ▶ Change your GitHub username to **your actual name**
- ▶ Your GitHub account is the one an employer will look at. Start it as you mean to continue.

Where everything lives;

- ▶ **The module site** — every lecture, every lab and the MCQ practice, in a browser

<https://danielcregg.is-a.dev/ai-assisted-programming>

- ▶ **Your own copy** — click below to create a private copy of the module repository for your lab work

<https://github.com/new?>

[template_owner=danielcregg&template_name=ai-assisted-programming&visibility=private&name=ai-assisted-programming-module](https://github.com/new?template_owner=danielcregg&template_name=ai-assisted-programming&visibility=private&name=ai-assisted-programming-module)

Summary;

- ▶ **12 weeks**, 2 hour lecture + 2 hour lab, no lab this week
- ▶ **32% + 32% + 9 × 4%** — two MCQs and nine practical assessments
- ▶ The first practical assessment opens **next week**, with the first lab
- ▶ Set up GitHub and the Student Developer Pack **before** next week

The one idea to keep: you don't have to read every line — but something has to check it, and if that something isn't a test, it's you.

Next week: what AI-assisted programming actually is — and the first lab.