

KK01 — Emerging trends in programming using artificial intelligence

Programming · Comprehensive exam study guide

STUDY DESIGN — YOU MUST KNOW

- using prompts to generate code
- automated debugging and testing of modules
- code optimisation
- responsible and ethical use of artificial intelligence tools

Artificial intelligence (AI) tools such as code assistants are now part of professional programming. For the exam you must be able to describe **how** they help at each stage of development and, just as importantly, discuss the **responsible and ethical** issues they raise. Marks are usually lost by students who only say AI 'writes code' without explaining the programmer's continuing responsibility.

Where AI assists during programming

AI assistants do not replace the programmer; they accelerate parts of the process while the developer stays accountable for the final solution.

Stage	How AI helps	What the programmer must still do
Generating code	A natural-language prompt (e.g. "write a function that validates an email") returns candidate code.	Read, understand, adapt and test the code; confirm it meets the actual requirements.
Debugging & testing	Suggests likely causes of errors, explains messages, and generates test data and test cases .	Verify the diagnosis, run the tests, and confirm expected = actual output.
Code optimisation	Suggests more efficient or more readable alternatives (e.g. removing repetition, better algorithms).	Judge whether the change is correct, maintainable and appropriate for the context.

Prompts that generate code

A **prompt** is the instruction given to the AI in natural language. Clear, specific prompts (stating inputs, outputs, data types and constraints) produce better candidate code. The output is a starting point only — it may contain logic errors, insecure patterns or code the student does not understand.

EXAMPLE

Prompt: "In Python, write a function that takes a list of integers and returns the average as a float, returning 0 if the list is empty." — the AI returns code, but the student must test the empty-list case and confirm the return type.

Responsible and ethical use

- **Accountability** — the developer is responsible for AI-generated code, so it must be understood, tested and validated, not trusted blindly.
- **Correctness & security** — AI can produce subtly wrong or insecure code (e.g. no validation); always review it.
- **Bias** — AI output reflects its training data and can embed bias or out-of-date practices.
- **Privacy** — do not paste private, personal or confidential data into prompts; it may be stored or reused.

- **Intellectual property & academic honesty** — acknowledge AI assistance; do not submit AI work as wholly your own; AI may reproduce others' copyrighted code.

Key definitions to memorise

Prompt	A natural-language instruction given to an AI tool describing the code or task required.
Code optimisation	Improving existing code so it runs more efficiently or is clearer/easier to maintain, without changing what it does.
Responsible use	Using AI tools while remaining accountable for verifying, understanding, testing and ethically sourcing the output.

Exam technique

EXAM TIP

- If a question says '**explain**', give a point + a reason. For responsible use, name the issue (e.g. privacy) AND say what the developer should do about it.
- Always mention that AI output must be **tested and understood** — this is the single most-rewarded idea in this KK.

COMMON MISTAKES

- Saying AI 'does the work for you' — examiners want the programmer's continuing responsibility.
- Confusing optimisation (making working code better) with debugging (fixing broken code).
- Listing ethical words (bias, privacy) with no explanation of why they matter.

Quick revision checklist

- I can describe how AI helps at three stages: generating, debugging/testing, optimising.
- I can explain what a prompt is and why prompt quality matters.
- I can discuss four responsible-use issues and what the developer should do about each.

Exam practice — questions & model answers

Q1. State one way an AI tool can support a programmer during development.

State · 1 mark

Model answer

- Any one: generate code from a prompt; suggest fixes during debugging; generate test data/cases; or recommend optimisations (1).

Q2. Describe how an AI assistant can support debugging and testing.

Describe · 2 marks

Model answer

- It can suggest likely causes of an error and explain the error message (1),
- and generate relevant test data and test cases so the programmer can compare expected and actual output (1).

Q3. Explain two responsible-use considerations when a student uses an AI assistant to generate code for a SAC.

Explain · 4 marks

Model answer

- Accountability — the student remains responsible, so must read, understand and test the generated code rather than trusting it (1) to ensure it is correct and secure (1).
- Privacy/academic honesty — they must not paste private data into prompts and must acknowledge AI assistance (1) rather than submit AI work as wholly their own (1).

Q4. Distinguish between using AI for code optimisation and using it for debugging.

Distinguish · 2 marks

Model answer

- Optimisation improves code that already works so it is faster or clearer (1);
- debugging finds and fixes code that is currently broken/producing errors (1).