



Faculty: Engineering

Department: ESE

**Module name: Modern
Programming Methods
(MPM)**

**Programme name: Applied
Computational Science and
Engineering**

Level: 7

**Approximate number of
students: 100**

Weighting: 50%

Module ECTS: 5

Module Type: Core

Programming assessment using AI

Assessment overview

This individual coursework assessment evaluates students' ability to develop, optimize, and profile code effectively. The first part of the assessment explicitly integrates the use of AI, requiring students to utilize ChatGPT to generate a solution for the 'Lorenz System'. Students must prompt ChatGPT to produce an initial code snippet, critically analyse its output and shortcomings, and iteratively refine the code using well-structured prompts. They are expected to reflect on the limitations of AI-generated output and demonstrate how their code improved through their interactions with ChatGPT.

Design decisions

Rationale for the assessment type

Students enter this Master's-level course with varied programming backgrounds, ranging from strong coding proficiency to limited experience. The goal of this assessment is to help students develop skills in Python programming, code testing, profiling, and optimization while fostering algorithmic thinking. It centres on cellular automata—mathematical models where each cell evolves based on specific rules that depend on the state of its neighbours. The focus is on having students think at an abstract, algorithmic level, then translate these abstractions into functional Python code.

This assessment is designed not only to test students' technical skills but also to encourage critical thinking about AI-generated solutions, fostering their ability to abstractly conceptualize algorithms and then bring those concepts to life in Python.

Adapting to Emerging Trends in Generative AI

In response to rapid developments in AI, students were explicitly required to use AI tools, specifically ChatGPT, for this assessment. This decision aligns with departmental policy across all Master's-level courses, grounded in the recognition that banning AI use is neither practical nor forward-thinking.

In earlier iterations, students were tasked with creating simple models in Python and verifying their functionality through testing. However, advancements in AI, particularly with tools like ChatGPT, have rendered these tasks trivial to complete with minimal effort. Consequently, the assessment was redesigned to actively incorporate AI use, with a specific focus on developing students' critical thinking and algorithmic design skills.

By redesigning the assessment in this way, students were challenged to think critically about the limitations of generative AI, reflect on its potential for enhancing code quality, and engage deeply with algorithmic abstractions. This approach aligns with the module's intended learning outcomes, which did not require revision as a result of this change.



Fit with other assessments on the module and the programme

The content of the Modern Programming Methods module forms the foundation for all subsequent modules in the programme. The skills students acquire in Python programming and GitHub are crucial not only for their independent research projects but also for collaborative group work, such as developing software packages.

This module includes two assessments, each contributing 50% to the final grade:

First Assessment: Focuses on the fundamentals of writing Python code. This assessment emphasizes the development, testing, profiling, and optimization of code, with a specific focus on algorithmic design and critical use of AI tools like ChatGPT.

Second Assessment: Builds on the first by focusing on software sustainability. Students are required to create a repository, implement tests for the code, include appropriate licensing, and set up continuous integration workflows in GitHub. The emphasis is on using GitHub Actions to ensure that testing is automated for every commit, thereby instilling best practices in collaborative software development.

Practicalities

Preparing students for assessment

Preparation for the assessment is embedded into the teaching materials and the briefing session, with the following key elements:

1. Embedded Exercises in Lectures

During lectures, students are provided with Jupyter notebooks containing exercises that align with the assessment tasks.

2. Assessment Briefing

Students receive a detailed briefing that explains the task, its objectives, and the specific deliverables required.

3. AI Familiarity

Programming assessment using AI

No formal AI-specific training is provided as part of the module. However, the nature of the cohort and programme makes this a reasonable approach.

Marking arrangements

The AI-related assessment question is worth 30 marks, divided into two key components:

- 1. Code Functionality and Testing (12 Marks)**
- 2. Critical Discussion and Analysis (18 Marks)**

As part of the submission, students are required to provide a record of their ChatGPT conversation history. This allows staff to:

Review how students interacted with ChatGPT.

Verify that AI was used ethically and responsibly.

Marks were also allocated based on the effectiveness of students' interactions with ChatGPT:

Students who demonstrated a critical approach, such as identifying and correcting errors in ChatGPT's output, received additional credit.

This grading criterion incentivized students to actively engage with the AI, rather than passively copying and pasting its responses, and provided insight into their problem-solving processes.

Provision of feedback

Feedback is delivered both at the group and individual levels to ensure students receive actionable insights:

- 1. Group Feedback**
- 2. Individual Feedback**
- 3. ChatGPT Usage Feedback**

By combining group-level insights with personalized advice, the feedback strategy ensures students gain a clear understanding of their performance and areas for growth.