Programming assessment using AI

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

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. A key objective of this module is to ensure all students achieve competency in using Python and GitHub, regardless of their starting point. The module's intended learning outcomes (ILOs) are:

1. Design software for sustainable and reproducible research.
2. Build software using test-driven development and continuous integration.
3. Develop software using a version control system.
4. Create software using the Python programming language.
5. Assess different sources of errors in science and engineering software and create tests for these errors.
6. Utilize and deploy cloud computing technologies for computational and data science.

The assessment discussed in this case study is the first in the module and serves as an introduction to key programming concepts and practices. Delivered during the first two weeks of the course, the module operates within a block-teaching format, with this being the sole focus of the students during this period. The assessment is due in the second half of the first week, providing students with 2–3 days to complete it. This tight timeline is balanced by the dedicated time available for this module and the structured support provided.

An important thing to consider with such tight deadlines is allowing for cases of mitigation so that students with extenuating circumstances who will be granted an extension can participate in the assessment. Same applies to students with disabilities, who also should have reasonable adjustments to be able to take the module and successfully complete their assessment.

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,

Programming assessment using AI

algorithmic level, then translate these abstractions into functional Python code.

To achieve this, students were provided with a framework containing a skeleton Python model and tasked with completing the following:

1. Code Development: Use ChatGPT to implement functionality within the skeleton code. Students were required to critically evaluate the AI-generated code, identifying shortcomings and iteratively refining it.
2. Code Profiling: Test and profile their code to measure performance, focusing on runtime and computational efficiency.
3. Code Optimization: Improve the code's performance based on profiling insights. Students could use their own knowledge, module content, ChatGPT, or a combination of these resources to achieve optimizations.
4. Reflection and Discussion: Write a discussion outlining how ChatGPT contributed to the task, what improvements they made to the AI-generated output, and provide details of the profiling results, including specific optimizations, performance gains, and rationale for changes.

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. Policing AI usage would necessitate a shift toward strictly proctored exams without internet access—an approach that overlooks the reality of AI's pervasive role in professional environments. Regardless of their future career paths, students will likely integrate AI into their daily workflows, making it essential to prepare them for its responsible and critical use.

Challenges of Policing AI Usage

While authenticity interviews were trialled in previous assignments to deter AI misuse, they proved ineffective for reliably detecting generative AI usage. Although the possibility of such interviews served as a deterrent, they did not provide conclusive evidence of misconduct. Furthermore, in cases where AI misuse was suspected across a cohort, authenticity interviews lacked the scalability and clarity to address such issues comprehensively.

Evolving the Assessment

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.

This is definitely a more innovative way of assessing. When approaching new assessment designs it is important to ensure that whatever we do adheres to principles of good practice with assessment

Employers currently don't have a blanket approach to the use of generative AI in business however many are adopting elements of it in varying ways. What employers do agree on is that generative AI uses and developments need to be constantly monitored and evaluated so assessments that help students understand how to do this are useful. Some industries, especially those with regulatory bodies such as some areas of finance, are very cautious about generative AI. As educators we are not going to be able to teach our students about every technical advance they will encounter in their future however we can teach them how to critically evaluate and communicate their evaluation regarding new products. This is a transferable skill that is very attractive to employers.

Anatomy of Assessment

Programming assessment using AI

Selection of ChatGPT

ChatGPT 3.5 was chosen for its widespread familiarity and accessibility at the time of the assessment. It was also recognized as one of the most capable AI tools for generating Python code, outperforming alternatives like Google Bard and earlier iterations of GitHub Copilot (which was not yet broadly available during this period). Requiring students to use the free-tier ChatGPT 3.5 ensured equitable access and consistency across submissions.

To ensure all students have equal access to the tool make sure you choose technology that is free and easily accessible to all, especially if it's such an integral part of the assignment. Wen choosing the technology you should also consider whether using College supported tools will be more beneficial to students, if they help you achieve the same result.

To ensure transparency, students were instructed to include links to their ChatGPT conversation histories as part of their submissions, alongside other references. This requirement reinforced the importance of acknowledging AI contributions, mirroring professional practices for citing tools and resources.

Goals of AI Integration

While the module's overarching purpose remained unchanged, integrating ChatGPT added a critical new dimension: the ability to evaluate and refine AI-generated output. The assessment aimed to:

- Foster an understanding of how the quality of AI output depends on the clarity and structure of prompts.
- Emphasize the importance of critically analysing AI-generated solutions for correctness and appropriateness.
- Encourage responsible and transparent use of AI tools, including proper acknowledgment and citation of generative AI contributions.

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.

The HE sector, as well as the College encourages embracing the use of AI as a more authentic practice better aligned with where the future of many industries lies. Since there is no reliable method to detect AI use, integrating it into assessment eliminates some of the worries around plagiarism and inappropriate use. Such redesign of assessment also helps students develop much needed skills for the future.

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.

Assessments that link to and build up the skills that are then further in other years presents a more connected assessment strategy.

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

1. 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.
2. 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.

While AI is permitted for the second assessment, it is not an integral part of the task as it is in the first assessment. The two assessments are designed to complement each other, with the first honing

Programming assessment using AI

students' ability to write efficient, functional code and the second extending their skills to encompass the broader workflows and processes required for sustainable software development.

To ensure equitable skill development, both assessments are individual tasks. This approach ensures that each student acquires a solid foundation in core programming and software development practices. These individual skills are then applied and further refined in group projects later in the programme, enabling balanced contributions and smoother collaboration.

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. These exercises include feedback mechanisms to guide students as they work through the problems, ensuring that they build the foundational skills needed for the assessment
2. Assessment Briefing
   Students receive a detailed briefing that explains the task, its objectives, and the specific deliverables required. This session covers both the coding and AI components of the assessment, helping students understand expectations and navigate the integration of AI tools like ChatGPT into their workflow.
3. AI Familiarity
   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. Students are expected to already have a basic understanding of ChatGPT, its capabilities, and how to use it effectively. Modules on machine learning, data science, and deep learning within the programme mean that most students are not only familiar with AI but also have an interest in its applications.

By relying on their existing familiarity and providing structured opportunities to practice, the approach emphasizes self-directed learning while ensuring students have sufficient support to succeed in the assessment.

*It is worth noting that, given the widespread use of AI among the MSc cohort, there have been discussions about introducing a more formal session on the ethical and responsible use of AI. However, a consensus on the best approach has not yet been reached. For now, the strategy has been to integrate examples of effective AI use into the lecture sessions, allowing students to see practical applications and consider responsible practices in context.

Where appropriate for the cohort it is useful to introduce a level of briefing around AI in addition to a thorough briefing around assessment expectations. This briefing should be designed with AI literacy in mind – helping students develop necessary knowledge and skills to productively engage with AI tools.

It is important to clearly outline to the students what is permitted and what is not permitted when it comes to AI. The College encourages staff to create opportunities where students' AI literacy can be developed. College's approach to the use of AI for assessment is outlined in "AI tools in teaching and assessment" document.

The main consideration from the Quality Assurance perspective is making sure that the rules of engagement are clear to the students from the start. This is where a good explanation about what the learning outcomes are for that module and how students are expected to demonstrate them is extremely useful as well.

Programming assessment using AI

Marking arrangements

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

1. **Code Functionality and Testing (12 Marks)**
   - Marks are awarded for the functionality of the students' code, including its correctness, robustness, and clarity.
   - Testing and code comments are also assessed, with credit given for thoroughness and adherence to good coding practices.
2. **Critical Discussion and Analysis (18 Marks)**
   - Students are required to write a discussion evaluating their use of ChatGPT.
   - This includes:
     - Assessing how effectively ChatGPT performed.
     - Identifying issues in the initial AI-generated output
     - Documenting how these issues were addressed and resolved, using knowledge gained during the module.
     - Profiling the code and discussing improvements, such as optimizations applied to enhance performance.
   - Marks are awarded for demonstrating a critical approach to analysing, refining, and optimizing the code, with particular emphasis on using ChatGPT effectively as part of the process.

ChatGPT Interaction Review

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.

The length and depth of conversations varied, with some students using only a few prompts and others engaging in extensive dialogue.

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.

When asking students to use new tools and redesigning assessment to include new approaches (such as the use of AI) it is important to have a clear marking scheme that is clearly communicate to the students. Especially when it comes to assessing critical use of AI it needs to be well explained to the students what the expectations around criticality in this context are. Exemplars could be a good way of communicating expectations.

Provision of feedback

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

1. **Group Feedback**
   - A summary of common issues observed across the submissions is shared with the entire cohort.
   - This feedback highlights frequent coding challenges, misconceptions, or areas for improvement, providing a collective learning opportunity.
2. **Individual Feedback**
   - Students receive tailored comments if there were notable aspects of their submission, either particularly strong or requiring significant improvement.

- Individual feedback includes specific suggestions to guide further development and refinement of their coding skills.

3. **ChatGPT Usage Feedback**
   - Comments are provided on students' interactions with ChatGPT, including suggestions on how to enhance their use of AI in the future.
   - This feedback emphasizes critical engagement, highlighting effective strategies and identifying areas where a more thoughtful approach could improve outcomes.

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

This multidimensional approach to feedback coming in individual as well as group form has the potential to help students better learn from feedback. It is also usefl to highlight to students other less obvious forms of feedback that they encounter such as opportunity to self-reflect in light of AI interactions, opportunities for peer discussions and feedback from formative assessment – all of that forms a feedback strategy on the course and is a useful tool for learning.

Online adaptations

The assessment can run online in its current form.

Advantages of the assessment type

Promotes Critical Thinking in AI Usage

- The assignment teaches students how to effectively use AI (specifically ChatGPT) and emphasizes the importance of critically evaluating its outputs.
- Students learn to not only rely on AI but also recognize its limitations, improving their overall problem-solving skills.

Strengthens Python Programming Skills

- Students gain practical experience in writing, optimizing, and profiling Python code, which are essential skills in both academic and professional settings.
- The focus on optimization and performance helps students develop an understanding of writing efficient code early in their academic journey.

Encourages Sustainable Coding Practices

- The module encourages students to approach coding with sustainability in mind, ensuring that their solutions are not only functional but also maintainable and optimized for long-term use.
- This is a foundational skill that will benefit students in group projects and research, where collaboration and code longevity are crucial.

Prepares for Real-World Application

- AI is increasingly integrated into professional workflows, and this assessment helps students understand its role in modern software development.
- Students are introduced to using AI in a practical context, preparing them for its use in future careers where AI tools are commonplace.

Supports Early Skill Development for Future Group Work

- By learning essential coding and AI usage skills early on, students are better equipped for the collaborative nature of later group projects, ensuring smoother teamwork and more effective contributions.

Limitations of the assessment type

Limitations in Reporting Interactions with AI

- Some students encountered challenges in submitting valid or accessible links to their ChatGPT conversation histories, which hindered the ability to verify their AI usage and the reasoning behind their code adjustments. Additionally, students had the option to engage in private chats and later submit refined versions of their conversations, which could complicate tracking the full scope of their AI interactions and understanding their thought process.

Access Issues for Some Students

- In one instance, a student was unable to create a ChatGPT account due to timing constraints, as the assessment was due shortly after the briefing. Unfortunately, the student did not proactively report this issue, which hindered their ability to complete the assessment as intended.

Dependence on AI Availability and Consistency

- The reliance on ChatGPT for part of the assessment introduces the risk of technical disruptions, such as outages or slowdowns, which could hinder students' ability to fully engage with the tool and complete their work on time.

Uneven Usage of AI Across the Cohort

- The varying levels of familiarity and skill with using ChatGPT among students led to

Programming assessment using AI

inconsistencies in how critically they approached AI-generated content. This disparity in AI usage impacted the overall quality of submissions and their critical engagement with the tool.

Advice for implementation

Ensure a Clear Method for Submitting AI Interactions

- Establish a clear process for students to submit their interactions with AI, ensuring that all conversations are accessible for review. Consider requiring students to retain their AI history and submit it as part of their assessment.

Define Consequences for Non-Compliance with Submission Guidelines

- Clearly communicate the consequences if students fail to follow submission guidelines, such as deleting AI chat history or submitting incomplete records. Decide whether deductions will apply, or alternative actions will be taken, and be transparent about this policy.

Ensure Equal Access to AI Tools for All Students

- Make sure that all students have access to the required AI tools. Set aside time for students to familiarize themselves with the platform before the assessment, to avoid issues with access or usage during the task.

Test AI Tools and Assessments Before Launch

- Test the AI tools and assessment tasks yourself before rolling them out. This will help you understand the types of responses AI provides and ensure that the assessment remains appropriately challenging. If AI can produce “perfect” answers, consider adjusting the task to maintain its focus on critical thinking.

Monitor and Limit the Impact of Your AI Interactions

- Be cautious of your own AI interactions when testing responses to ensure your queries don’t influence future AI outputs. Limit the number of prompts you provide, and test tools well in advance of the assessment.

Encourage Critical Engagement with AI Output

- Focus on encouraging students to critically evaluate AI-generated content. Ensure that the assessment encourages students to not only use AI but also to assess its strengths and limitations and improve upon its outputs where needed.

Provide Guidelines for Ethical AI Use

- Set clear expectations for ethical AI use. Ensure students understand how to properly attribute AI-generated content and the importance of transparency when incorporating AI tools in their work.

Clarify the Role of AI in the Assessment Process

- Make it clear how AI should be used in the assessment. Define AI’s role as a tool for assisting students with tasks rather than providing complete solutions and emphasize that students should use it as a learning aid, not a replacement for their own problem-solving.