Received: 18 June 2024 | Accepted: 17 December 2024

DOI: 10.1111/bjet.13558

ORIGINAL ARTICLE

British Journal of Educational Technology | BERA

Assessing student perceptions and use of instructor versus AI-generated feedback

Erkan Er[1] | **Gökhan Akçapınar**[2] | **Alper Bayazıt**[3] | **Omid Noroozi**[4] | **Seyyed Kazem Banihashem**[4,5]

[1]Department of Computer Education and Instructional Technology, Middle East Technical University, Ankara, Turkey

[2]Department of Computer Education and Instructional Technology, Hacettepe University, Ankara, Turkey

[3]Department of Medical Education and Informatics, Ankara University, Ankara, Turkey

[4]Omid Noroozi: Education and Learning Science Group, Wageningen University and Research, Wageningen, The Netherlands

[5]Seyyed Kazem Banihashem: Online Learning and Instruction, Open Universiteit, Heerlen, The Netherlands

Correspondence
Erkan Er, Department of Computer Education and Instructional Technology, Middle East Technical University, Ankara, Turkey.
Email: erkane@metu.edu.tr

Funding information
BAGEP Award of the Science Academy, Türkiye

[Correction added on 7 January 2025, after first online publication: Third author's ORCID has been corrected, in this version.]

Abstract: Despite the growing research interest in the use of large language models for feedback provision, it still remains unknown how students perceive and use AI-generated feedback compared to instructor feedback in authentic settings. To address this gap, this study compared instructor and AI-generated feedback in a Java programming course through an experimental research design where students were randomly assigned to either condition. Both feedback providers used the same assessment rubric, and students were asked to improve their work based on the feedback. The feedback perceptions scale and students' laboratory assignment scores were compared in both conditions. Results showed that students perceived instructor feedback as significantly more useful than AI feedback. While instructor feedback was also perceived as more fair, developmental and encouraging, these differences were not statistically significant. Importantly, students receiving instructor feedback showed significantly greater improvements in their lab scores compared to those receiving AI feedback, even after controlling for their initial knowledge levels. Based on the findings, we posit that AI models potentially need to be trained on data specific to educational contexts and hybrid feedback models that combine AI's and instructors' strengths should be considered for effective feedback practices.

KEYWORDS
AI-generated feedback, artificial intelligence, ChatGPT, higher education, instructor feedback, programming education

Practitioner notes

What is already known about this topic

- Feedback is crucial for student learning in programming education.
- Providing detailed personalised feedback is challenging for instructors.
- AI-powered solutions like ChatGPT can be effective in feedback provision.
- Existing research is limited and shows mixed results about AI-generated feedback.

What this paper adds

- The effectiveness of AI-generated feedback was compared to instructor feedback.
- Both feedback types received positive perceptions, but instructor feedback was seen as more useful.
- Instructor feedback led to greater score improvements in the programming task.

Implications for practice and/or policy

- AI should not be the sole source of feedback, as human expertise is crucial.
- AI models should be trained on context-specific data to improve feedback actionability.
- Hybrid feedback models should be considered for a scalable and effective approach.

INTRODUCTION

It is a widely acknowledged fact that feedback constitutes an essential element of the learning process (Hattie & Timperley, 2007). Decades of educational research have provided consistent evidence that feedback significantly enhances students' learning experiences and leads to improved academic outcomes (Ajjawi & Boud, 2018; Cavalcanti et al., 2019; Gan et al., 2021). Particularly in domains where students' production skills are emphasised, such as programming education, feedback can play a crucial role in addressing the gaps in theoretical understanding and guiding the practical application of knowledge (Fu et al., 2023). Programming courses involve frequent hands-on code-writing activities, and students may struggle with writing correct code, debugging and problem-solving in programming tasks, which may lead to frustration and decreased motivation (McCracken et al., 2001). In this regard, feedback can support students by identifying areas for improvement, highlighting mistakes and providing suggestions for better coding practices. Research has shown that effective feedback practices can significantly guide students towards mastery of programming skills and knowledge and subsequently enhance their confidence and competence in programming (Foster et al., 2024).

Despite its importance, implementing a feedback practice where each student receives tailored guidance on their coding efforts presents a significant challenge for instructors (Nguyen et al., 2014). First, the inherent diversity of programming assignments poses difficulties in effective feedback provision. Because of the problem-solving nature of programming, students may vary in how they approach a programming task with its own set of code structures and styles (Ezeamuzie, 2023; Glassman et al., 2015; Kinnunen & Simon, 2012). This variability makes it difficult for programming instructors to apply a one-size-fits-all approach to assess and provide feedback. In addition, beyond correctness, instructors need to consider factors such as code efficiency, modularity and logic (Keuning et al., 2017). These are essential aspects of programming and require significant instructor time and effort in

assessment. These problems are accentuated when dealing with a large number of students. Over the last decades, the growing interest in computer science-related careers has led to high enrolment rates in programming courses (Camp et al., 2017; Maghsoudi, 2023). As a result, the low instructor-to-student ratios add to teaching workload and make effective feedback practices infeasible on a large scale.

The need for scalable and efficient feedback practices has led to the exploration of various technological solutions to enable automated feedback provision in programming education. While earlier research mostly focused on the development of automated grading systems (Hao et al., 2022), this focus has shifted in later research to tools that rely on dynamic or static approaches to generate automated feedback (Banihashem et al., 2022; Luxton-Reilly et al., 2018). With the recent advancements in large language models (LLMs), the use of AI for feedback provision has attracted much attention lately in the research community (Banihashem et al., 2024; Bhullar et al., 2024). An LLM is a type of advanced deep learning algorithm trained on vast data that can produce human-like text with high accuracy in response to a prompt and engage in dynamic interactive conversations (Chang et al., 2024). ChatGPT is the most popular application based on an LLM, called Generative Pretrained Transformer, developed by OpenAI (Bhullar et al., 2024). Because LLMs have demonstrated the ability to understand code syntax, semantics and common programming errors, there is an increasing, yet few, number of studies investigating their potential as feedback providers in programming education, regardless of the scale. The literature so far does not provide sufficient evidence regarding the impact of feedback generated by LLMs on student learning and progress.

Thus, although instructor feedback is proven to be effective for student learning, it places an unsustainable burden on instructors in large computer science classrooms due to high teaching workload. While the emerging LLMs hold great promise in education, there is insufficient evidence regarding their potential for generating effective feedback in programming education. To bridge this research gap, it is important to investigate both how students perceive AI-generated feedback and the extent to which they benefit from it. While perceptions such as usefulness and fairness (Strijbos et al., 2010) are associated with students' willingness to engage with and act on feedback (Carless, 2006), the actual learning outcomes are essential to capture tangible impact of AI-generated feedback (see Banihashem et al., 2024). Therefore, a comprehensive approach that considers both student perceptions and the actual impact of feedback on learning is necessary to provide robust evidence regarding the potential of LLMs for feedback provision in programming education.

This study addresses this gap by investigating the effectiveness of AI-generated feedback (using an LLM) in comparison to instructor feedback within an undergraduate Java programming course. To this aim, an experimental research design was conducted, wherein students were randomly assigned to one of the feedback conditions: either receiving instructor feedback or AI-generated feedback. The central focus of the comparison between these conditions was the extent to which feedback yields concrete and measurable improvements in students' performance on a programming task. In other words, students' use of feedback was identified based on the improvements in their work following feedback. This focus on the tangible impact of feedback on student progress distinguishes this study from much of the existing literature, which mostly relies on self-reported students' perceptions of feedback quality or effectiveness (Noroozi et al., 2024). Nevertheless, as students' perceptions may be associated with how they approach and use feedback (Van Der Kleij & Lipnevich, 2021), this study also examines and compares students' perceptions of LLM-generated and instructor-generated feedback, thereby offering a complete perspective on the potential of LLMs as automated feedback providers in programming education.

BACKGROUND

Automated feedback generation

The importance of scalable feedback practices in computer science education has been acknowledged since the early 1960s (Hollingsworth, 1960). Early research in this domain concentrated on the development of automated grading systems (Hao et al., 2022). Later studies have increasingly shifted towards tools that generate automated feedback (Luxton-Reilly et al., 2018). In this area, two dominant approaches have been the dynamic and static analysis of programming assignments (Ala-Mutka, 2005). Dynamic analysis mainly provides feedback on the correctness of a computer program based on test cases and requires that the program compiles without errors (Liang et al., 2009). Therefore, this approach is more scoring-oriented and limited in its capacity to provide effective feedback. On the other hand, static analysis can assess student codes without compiling them and provide feedback on potential errors and code quality issues (Denny et al., 2014). An important limitation of this approach is that it may flag a correct code as problematic, thus leading to high numbers of false positives (Zougari et al., 2016). Inaccurate feedback may damage students' learning process and undermine their confidence and motivation in learning programming (Fu et al., 2023).

Thanks to the recent advancements in LLMs, there has been a rising interest in the research community regarding the use of AI to provide feedback. However, research on LLM-generated feedback in programming education is still in its early stages. To begin with, Zhang et al. (2024) studied students' perceptions and preferences for the feedback generated by ChatGPT in a Java course. Their findings, which rely on self-report data, indicated students' positive perception of ChatGPT-generated feedback, while their preferences for the feedback tone varied. Differently, Jukiewicz (2024), in a Python programming course, compared the scores generated by ChatGPT with teacher scores and evaluated the feedback generated by ChatGPT. He reported a high correlation between AI and human grading and suggested that ChatGPT can generate meaningful feedback for programming assignments, which sometimes may include incorrect information. However, this study did not consider how students perceive or use AI-generated feedback. Similarly, Kiesler et al. (2023) explored the potential of ChatGPT to generate formative feedback for students' code in programming exercises. Based on the qualitative analysis of feedback content, the researchers concluded a good potential of LLMs for generating meaningful feedback while also acknowledging the risk of providing misleading information, as in Jukiewicz's (2024) study. Estévez-Ayres et al. (2024) followed a different approach by comparing the performance of two popular LLMs in providing feedback on student exercises in concurrent programming. The feedback focused on the correctness of the code, aiming to identify and give information about common concurrency errors. The authors found that none of these models could generate accurate feedback.

The emerging literature presents a lack of consensus on the potential of LLMs to provide feedback in programming education. While some studies have demonstrated promising results, highlighting the ability of LLMs to generate relevant feedback, others have reported inconsistencies and limitations in LLM performance. Moreover, current research is limited in both scope and real-world application. Most research studies focused on the ability of LLMs to generate feedback with simple prompts, which do not align with good educational practices in authentic settings where an established assessment rubric is used to assess and provide detailed feedback on student work. Moreover, there is a significant lack of evidence regarding the impact of AI-generated feedback on student learning outcomes and progress. In other words, it remains unknown to what extent students understand and use AI-generated feedback to improve their learning and advance in their work.

Feedback perceptions and impact on learning

Students' engagement with feedback is linked to their perception and interpretation of it (Nicol, 2013). Understanding student perceptions is important, as these perceptions influence whether and how students engage with and utilise feedback (Carless, 2006). A comprehensive literature review by Jonsson (2013) highlighted that students' perception of feedback as having low usefulness is a major factor in their lack of engagement. Similarly, a study by Jonsson and Panadero (2018) emphasised that student perceptions are a key determinant of whether they engage with feedback productively.

Student perceptions of feedback can be considered from various dimensions. A fundamental dimension of feedback perception is its usefulness, which refers to the extent to which students find the feedback easy to understand and apply (Strijbos et al., 2010). Research consistently shows that if feedback is perceived as usable, students are more likely to engage with it and implement the suggestions for improvement (Harks et al., 2013). Students' perception of the fairness of feedback is another crucial factor influencing their engagement (Bazvand & Rasooli, 2022; Strijbos et al., 2010). When students perceive feedback as just, unbiased and reflective of their actual performance, they are more likely to accept and utilise it constructively (Panadero et al., 2023).

The perceived developmental value of feedback, or the extent to which students believe it contributes to their learning and growth, is an important motivator for feedback engagement (Lizzio & Wilson, 2008). When students see feedback as a tool for improvement and development, they are more likely to actively seek it out and use it to refine their skills and knowledge (Boud & Molloy, 2012). Lastly, students' engagement with feedback can be positively affected by their perception of how encouraging the feedback is (Hattie & Timperley, 2007). Encouragement refers to the degree to which students feel motivated and supported by the feedback (Lizzio & Wilson, 2008).

It is important to acknowledge that how students perceive may not always relate with the desired learning outcomes (Kerman et al., 2023; Smits et al., 2008). While research suggests a positive correlation between student perceptions of feedback and their subsequent performance (Hattie & Timperley, 2007), some studies underline the fact that what students perceive may deviate from their actual learning gains (Barzilai & Blau, 2014; Porat et al., 2018). Therefore, to accurately capture the impact of feedback on learning, student perceptions should be supported with evidence about the actual impact of feedback on performance outcomes (Noroozi et al., 2024; Smits et al., 2008). When examining the effectiveness of AI-generated feedback, we should not only consider student perceptions but also investigate the direct impact the feedback has on actual learning outcomes and performance.

THE SIGNIFICANCE OF THE STUDY

While research on using LLMs to generate feedback for students is increasing, the existing literature primarily focuses on experimentation and tends to overlook the practical impact of AI-generated feedback in authentic educational contexts. Prior research, while promising, leaves a significant gap in our understanding of the efficacy of AI-generated feedback in fostering student improvement. Although LLMs, as natural language processing tools, can produce seemingly reasonable feedback, this does not guarantee that students will effectively engage with or implement the suggestions provided. It is crucial to determine whether such feedback is truly actionable and facilitates meaningful student learning. Additionally, it is important to measure students' perceptions of AI-generated feedback, as their feelings and views of feedback can significantly influence how they engage with it.

To that end, we conducted an experimental study in an undergraduate Java programming course. Students were randomly assigned to receive either AI-generated feedback or instructor feedback on their lab submissions, and their subsequent improvement in resubmissions was measured. Feedback perception was assessed across four dimensions including usefulness, fairness, development value and encouragement. Additionally, we collected data on students' perceptions of the feedback they received. In this way, this study offers a comprehensive understanding of the actionable nature of AI-generated feedback along with how it is perceived by students. The findings of this study will contribute to the growing body of knowledge about AI's true potential in educational settings.

The research questions of this study are provided below:

- Is there a statistically significant difference in the mean feedback perception scores (usefulness, fairness, developmental value and encouragement) between students receiving instructor feedback and those receiving AI feedback?
- Is there a statistically significant difference in the mean improvement of lab assignment scores between students receiving instructor feedback and those receiving AI feedback?

METHOD

Context and participants

The study context was a Java programming course offered to second-year undergraduate students during the Spring semester of 2024 at a public university in Türkiye. This context was chosen as it represents a scenario where frequent feedback would be beneficial due to numerous practice activities and the students' lack of experience with object-oriented programming languages like Java. The course was conducted face-to-face, with two hours allocated for lectures on Tuesdays and another two hours for lab practice on Fridays each week, under the guidance of the course instructor. All course materials were distributed using the university's Moodle-based learning management system (LMS). Students were required to complete practice quizzes on lecture and lab days and to submit the homework activities before each lab day. The course included three lab exams, one midterm and one final exam. By the time of this study, the first lab exam, which is the focus of this research, and the midterm exam had been administered. The lab exam required students to write code hands-on to develop a number-guessing Java program based on detailed instructions. During the exam, students used their own laptops and were prohibited from seeking help from the internet or others.

A total of 54 students were enrolled in this course, which was part of the curriculum of the Department of Educational Technology. Among those were 24 female and 30 male students. Forty-five were second-year students, while nine were third-year students (retaking the course). All students had some familiarity with programming as they had taken a Python programming course in the previous semester.

Experimental design and procedure

This study employed a randomised controlled trial with a between-subjects design to assess students' use and perceptions of feedback received from an instructor (control group) and AI (experimental group). Prior to the study, ethical approval was obtained from the Institutional Review Board of Human Research at the first author's university. The study procedure unfolded as follows:

1. Lab exam administration: All 54 enrolled students participated in a Java programming lab exam.
2. Random assignment: One day after the exam, students were informed about an optional feedback session scheduled for one week later. Subsequently, students were randomly assigned to either the control group or the experimental group using a Python script to ensure an unbiased allocation process.
3. Feedback generation and delivery: The control group received feedback on their lab exam submissions generated by the course instructor, while the experimental group received feedback generated by ChatGPT. This feedback was delivered through the LMS. Students were unaware of the source of the feedback (i.e. whether it was generated by AI or the instructor). This was intended to minimise potential biases in students' perceptions and responses to the feedback, ensuring a fair and objective assessment of the feedback's effectiveness on student performance. At this point, students were not granted access to their submissions to prevent them from viewing feedback before the feedback session.
4. Feedback session and revision: One week after the lab exam, 35 students (18 control, 17 experimental) attended the face-to-face feedback session. This represents a notable decrease from the initial 54 students who took the exam. Students signed a consent form at the beginning. During this session, both control and experimental groups simultaneously worked on improving their lab exam submissions based solely on the feedback provided, without utilising any external resources. Students were informed that their exam scores would not be affected, but they would receive a bonus point for participating and submitting their revised work. The instructor monitored the session to ensure adherence to instructions but did not provide any individual assistance.
5. Questionnaire: At the end of the feedback session, students completed an online questionnaire aimed to capture their perceptions of the feedback they received.

Feedback generation

In the control group, the instructor provided feedback on student submissions based on a rubric refined over several years. This rubric evaluated code across multiple dimensions: Completeness, Modularity, Design of Logic, Syntax and Definition Errors, Code Efficiency, and Readability. Each dimension was graded on a five-point scale, providing a comprehensive assessment of code quality. The instructor reviewed the Java file submitted by the student, cross-referenced it with the lab instructions and used the rubric to provide detailed comments. The instructor's feedback comments typically used plain language, offering hints and explanations on how students could improve their work without directly providing solutions. For the repeating errors across student submissions, the instructor provided standardised explanations to ensure consistency in feedback between different students.

ChatGPT 4 was used as the AI technology to produce feedback on each student submission for the experimental group. The reason for choosing ChatGPT 4 was that it leverages GPT-4, which was the state-of-the-art LLM at the time of this study. In addition, ChatGPT is widely recognised as the most popular generative AI tool in the education. Therefore, demonstrating its effectiveness in this research could stimulate considerable interest within the educational community. ChatGPT was provided with the same rubric in an Excel file, the complete lab instructions in a PDF file and the student submission in a Java file, all uploaded using the file-uploading feature of ChatGPT 4. The same prompt, provided in Figure 1, was used to generate feedback for all students. The prompt was designed to mirror the instructor's strategy when writing his feedback, characterised by using

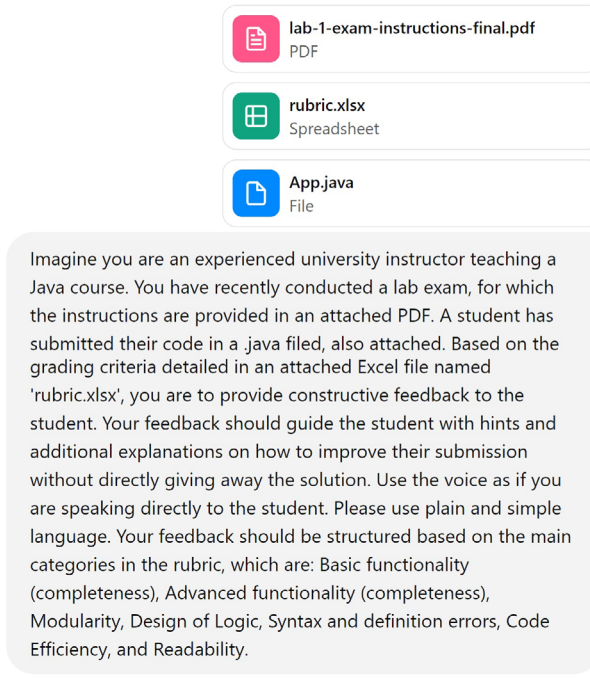

FIGURE 1 ChatGPT prompt window to generate AI feedback.

plain language and providing hints and explanations on how students can improve their work without including code that reveals the solution or part of it. This approach aimed to achieve equality in content and coverage between the instructor and AI feedback, ensuring both used the same rubric.

Measurement tools

An online questionnaire employing 5-point Likert subscales (1: Fully Disagree, 5: Fully Agree) from Strijbos et al.'s (2010) feedback perceptions scales was administered to gauge student perceptions of feedback fairness, usefulness and acceptance. Each subscale contained three items as in the original scale. The fairness items (eg, 'I would consider this feedback fair') measure the degree to which students consider the feedback fair; the usefulness items (eg, 'The feedback would provide me a lot of support') assess how useful students found the feedback; and last, the acceptance items (eg, 'I would dispute this feedback') measure the degree to which students accept the feedback. In the original study (Strijbos et al., 2010), Cronbach's alpha values for these subscales were reported as 0.80 for fairness, 0.82 for usefulness and 0.69 for acceptance, indicating acceptable to good internal consistency.

Additionally, the instrument included two 5-point Likert subscales (1: Fully Disagree, 5: Fully Agree) to assess students' perceptions of the encouragement and developmental focus inherent in the feedback. Encouragement in feedback refers to the positive comments that acknowledge the good aspects of student work and recognise the effort invested, while the developmental nature is associated with feedback's capacity to guide students on how to improve. These items were adapted from Lizzio and Wilson (2008) scale to measure the extent to which students perceive the feedback as encouraging and developmental. Students' perceptions of encouragement in feedback were measured using four items (eg,

'The feedback recognised the effort I had made') from the original scale. In comparison, the developmental value of feedback was measured using six items (eg, 'Commented on not just what was wrong, but also what to do about it'). One item (ie, 'Marker offered opportunities to clarify their comments') was excluded since it was irrelevant to the context of this study. The Cronbach's alpha values reported by Lizzio and Wilson (2008) for these subscales were 0.85 for encouragement and 0.82 for developmental feedback, indicating high internal consistency.

Previous research has shown that pre-existing knowledge can impact how students interpret and apply feedback (Fyfe & Rittle-Johnson, 2016; Shirah & Sidney, 2023). Accordingly, this study considers students' knowledge levels as a confounding variable that may affect how students use feedback to improve their work, potentially obscuring the true effect of the independent variable (ie, feedback source). In particular, students' midterm scores were used as a measure of their knowledge levels. The midterm exam conducted one week after the experiment consisted of 20 multiple-choice questions and 10 short-answer questions. Midterm exam was considered a valid measure of students' knowledge levels, as both the midterm and the lab exam were administered in close temporal proximity and focused on the same Java programming topics.

The change in lab assignment scores was computed to measure the actionability of feedback. First, students' initial lab submissions were graded. After they had the opportunity to improve their work based on the feedback received, their revised submissions were graded. Critically, the same rubric used by both the instructor and AI to generate feedback was also employed to grade all submissions. This consistent approach ensured a standardised evaluation process, minimising potential grader bias. The change in scores for each student was calculated by subtracting the initial scores from the latest scores. This approach, commonly used in educational research (Zimbardi et al., 2017), measures the extent to which students effectively use feedback to improve their work, thus reflecting the actionability of the feedback provided.

Data analysis

Before the analysis, preprocessing the data was necessary. This process involved identifying students who had submitted both the revised version of their lab submissions and completed the survey. As a result, three students from Group 1 (instructor feedback) and two students from Group 2 (AI feedback) were excluded. In addition, one student from each group was removed for resubmitting the identical source code file, verified by the file's last modified date. Consequently, in the data analysis, a total of 35 students were included (18 in group 1 and 17 in group 2).

The data analysis can be divided into two parts. In the first part, which involves the comparison of students' feedback perceptions across conditions, Cronbach's alpha values were initially computed to ensure the internal consistency of the survey construct, which was followed by performing descriptive statistical analysis for each construct measured. Afterwards, the Shapiro–Wilk test was conducted to assess the normality of the mean scores for each construct. Depending on the results of the normality tests, appropriate statistical tests were chosen (ie, independent *t*-test if data were normally distributed, and otherwise, Mann–Whitney *U* test).

In the second part, which focuses on the comparison of feedback actionability, an ANCOVA was applied to compare the changes in lab scores between the two groups, with the feedback source as the independent variable, the change in lab scores as the dependent variable and midterm scores as the covariate. All statistical analyses were conducted using the statsmodels library in Python (Seabold & Perktold, 2010).

RESULTS

Comparison of feedback perceptions

The survey instrument used in this study included items to assess students' perceptions of feedback. Perceptions were measured using four constructs, each referring to a particular feedback aspect: fairness, usefulness, development and encouragement. Cronbach's alpha values were computed to check the reliability of the survey items. Results indicated a high level of internal consistency for all constructs measured: fairness of feedback ($\alpha = 0.80$), usefulness of feedback ($\alpha = 0.94$), development capacity ($\alpha = 0.90$) and encouragement ($\alpha = 0.80$). After ensuring internal consistency, a single score was computed for each construct by averaging the scores. Descriptive statistics for each construct, as measured by the survey, are provided in Table 1. It is important to note that one student from each group did not complete the survey, resulting in 17 students in the control group and 16 in the experimental group for the final analysis.

According to descriptive results, both instructor and AI feedback were perceived positively by students. Students who received instructor feedback (ie, group 1) consistently considered feedback more fair ($M = 4.59$, $SD = 0.46$), useful ($M = 4.39$, $SD = 0.79$), developmental ($M = 4.17$, $SD = 0.84$) and encouraging ($M = 4.53$, $SD = 0.65$) compared to students who received AI feedback (Group 2: $M = 4.25$, $SD = 0.66$ for fairness; $M = 3.77$, $SD = 0.94$ for usefulness; $M = 3.99$, $SD = 0.65$ for development; and $M = 4.22$, $SD = 0.58$ for encouragement). This suggests that students perceived instructor feedback to be of higher quality across all four dimensions compared to feedback from the AI.

The normality of the mean scores was tested using the Shapiro–Wilk test. The results indicated that the scores for fairness (Group 1: $W = 0.816$, $p = 0.003$; Group 2: $W = 0.895$, $p = 0.067$), usefulness (Group 1: $W = 0.770$, $p < 0.001$; Group 2: $W = 0.938$, $p = 0.322$), development (Group 1: $W = 0.883$, $p = 0.036$; Group 2: $W = 0.896$, $p = 0.069$) and encouraging (Group 1: $W = 0.754$, $p < 0.001$; Group 2: $W = 0.941$, $p = 0.357$) were mostly not normally distributed in Group 1 but were normally distributed in Group 2. Given the results of the normality tests, Mann–Whitney U tests were used to compare the scores between the two groups. The results are presented in Table 1.

The Mann–Whitney U test revealed a significant difference in usefulness scores between the groups ($U = 191.0$, $p = 0.04$), suggesting that students perceived the usefulness of instructor feedback differently from AI feedback. No significant differences were found for fairness ($U = 179.0$, $p = 0.11$), development ($U = 163.0$, $p = 0.34$) or encouragement ($U = 180.5$, $p = 0.10$) scores.

TABLE 1 Mann–Whitney *U* test results for students' perceptions of instructor versus AI-generated feedback regarding fairness, usefulness, development and encouragement.

	Group 1: Instructor feedback ($n = 17$)		Group 2: AI feedback ($n = 16$)		Mann–Whitney *U*	
	Mean	Std	Mean	Std	Stat	*p*-Value
Fairness	4.59	0.46	4.25	0.66	179.0	0.11
Usefulness	4.39	0.79	3.77	0.94	191.0	0.04
Development	4.17	0.84	3.99	0.65	163.0	0.34
Encouragement	4.53	0.65	4.22	0.58	180.5	0.10

TABLE 2 Descriptive statistics for the changes in scores, midterm scores and laboratory scores before feedback (BF) and after feedback (AF) for each group.

	Group 1: Instructor feedback ($n=18$)				Group 2: AI feedback ($n=17$)			
	Lab BF	Lab AF	Change	Midterm	Lab BF	Lab AF	Change	Midterm
Mean	71.17	80.28	9.11	47.61	59.71	63.71	4	35.35
Std	17.85	18.59	7.28	23.56	23.91	21.87	3.34	19.05
Median	72	83	9	48	54	59	4	31
Min	33	37	0	8	15	20	0	8
Max	93	100	28	86	97	97	12	86

Comparison of score improvements after feedback use

The descriptive statistics for the midterm scores, lab assignment scores before feedback (BF) and after feedback (AF), and the change in the lab scores are provided for each group in Table 2. Both lab and midterm exams were assessed on a 100-point scale. Results indicate that students who received instructor feedback outperformed those who received AI feedback on various metrics. The mean midterm score was higher for the instructor-feedback group ($M=47.61$, $SD=23.56$) than for the AI-feedback group ($M=35.35$, $SD=19.05$). Moreover, the instructor-feedback group had higher mean lab scores both BF ($M=71.17$, $SD=17.85$) and AF ($M=80.28$, $SD=18.59$) compared to the AI-feedback group (BF: $M=59.71$, $SD=23.91$; AF: $M=63.71$, $SD=21.87$). Furthermore, students in the instructor-feedback group exhibited a greater mean increase in lab scores ($M=9.11$, $SD=7.28$) than those in the AI-feedback group ($M=4$, $SD=3.34$). The median values support this trend. Additionally, the maximum increase in lab scores was notably higher in the instructor-feedback group (28) compared to the AI-feedback group (12).

The actionability of the feedback was measured in terms of the change in the lab scores after students applied the improvements based on the feedback received, whether it was instructor-generated (Group 1) or AI-generated (Group 2). Since the students' knowledge levels could impact how they respond to feedback (Shirah & Sidney, 2023), midterm scores were introduced as a confounding variable that measures their knowledge levels. An ANCOVA was conducted to identify significant differences in feedback actionability between the groups, with the independent variable being the feedback source or provider, the dependent variable being the change in the lab scores and the confounding variable being midterm scores.

Before the analysis, the assumptions for ANCOVA were checked. The Shapiro–Wilk test was conducted to check if the change in the scores was normally distributed within each group. The Shapiro–Wilk tests for both groups suggested that the data do not significantly deviate from normality (Group 1: $W=0.900$, $p=0.057$, Group 2: $W=0.898$, $p=0.062$). Next, the homogeneity of regression slopes, which refers to the equality of the regression lines' slopes between the dependent variable and the covariate across groups, was checked. The interaction term was not significant ($p=0.128$), indicating that this assumption was met. Finally, Levene's test was used to check if the variances of the change in the scores were equal across the two groups, thus assessing the homogeneity of variance. Levene's test indicated a significant difference in variances between the two groups, $F(1, 33)=4.501$, $p=0.041$, thus violating the homogeneity of variance assumption.

To ensure an analysis that is robust to the violation of homogeneity of variance (aka, homoscedasticity), ordinary least squares with robust standard errors were conducted. This analysis involves computing robust standard errors using the HC3 estimator to account for heteroscedasticity (MacKinnon & White, 1985). The adjustment for heteroscedasticity using

TABLE 3 The results of robust ANCOVA to compare the improvements in lab exam scores between control and experimental groups.

	Coefficient	Std. error	*t*	$p > \|t\|$
Intercept	8.57	3.77	2.28	0.03
Group	−4.97	2.40	−2.07	0.04
Midterm	0.01	0.05	0.21	0.83

robust standard errors (HC3) helps make valid inferences even when the variance is not constant across groups. The robust ANCOVA results (see Table 3) indicate a statistically significant difference between the two groups regarding the change in student scores after controlling their knowledge levels measured by the midterm scores. Given the coefficient for the group variable (−4.97) and its statistical significance ($p = 0.04$), students who received AI feedback had a significantly lower average change in scores compared to those who received instructor feedback after accounting for their midterm scores. The coefficient for the midterm variable, 0.01, is not statistically significant ($p = 0.83$), indicating that midterm scores do not significantly influence the changes in scores. These results indicate that students who received instructor feedback improved scores significantly more than those who received AI feedback. In addition, the midterm scores, used as a control variable to account for initial knowledge level, did not significantly impact the improvement in scores, indicating that students' initial knowledge level does not confound the difference in feedback use.

DISCUSSION

The recent emergence of powerful LLMs in the field of AI has led to significant interest in their potential applications in education. One area of particular focus is the use of AI to generate automatic feedback for students. Contributing to the rapidly evolving literature, this study reported the results of an experimental investigation that compared student perceptions of feedback and the actionability of feedback in two conditions: AI-generated feedback and instructor feedback.

Regarding the first research question focusing on feedback perception comparison, while descriptive statistics revealed that students perceived instructor feedback as significantly more useful than AI feedback. While descriptive statistics hinted that instructor feedback was also perceived as more fair, developmental and encouraging, these differences were not statistically significant. This finding is consistent with existing literature suggesting that personalised, human-generated feedback is often perceived as more beneficial and usable compared to automated feedback systems (Lipnevich & Smith, 2009). Regarding the second research question focusing on the lab score improvement after incorporating feedback, the findings revealed an important discrepancy between the instructor and AI feedback groups. In particular, robust ANCOVA analysis indicated that students receiving AI feedback showed significantly lower score improvements compared to those receiving instructor feedback, even after controlling for their knowledge levels. These findings are supported by research highlighting the superior effectiveness of instructor feedback in promoting deeper learning and performance on specific tasks compared to automated feedback (Tian & Zhou, 2020; Zhang & Hyland, 2018). While recent research by Banihashem et al. (2024) suggests that AI, such as ChatGPT, can indeed provide valuable feedback, this AI-generated feedback was more descriptive than peer feedback, which was directly focused on identifying problems. This study highlighted a potential complementary role for AI in the feedback process. This potential synergy could also be applied to programming, where AI might address general syntax errors, freeing instructors to focus on more nuanced or context-specific issues.

However, the current study's findings emphasise that AI feedback, at least in its present form, may not be as effective as instructor feedback in fostering learning and performance improvement, particularly in programming contexts.

Several factors may contribute to the observed differences. First, instructors often have a deep understanding of the course content, learning objectives and assessment criteria, enabling them to provide contextually relevant and accurate feedback. On the other hand, AI, despite its computational power, lacks this contextual awareness and might struggle to grasp the nuances of specific assignments or assessment rubrics, potentially leading to less precise feedback (Liu et al., 2023). This limitation might worsen due to hallucination—a known weakness of LLMs, wherein they generate false or misleading information (Xu et al., 2024). This phenomenon is particularly relevant in feedback scenarios, as LLMs are trained on vast datasets possibly containing a wide variety of rubrics and assessment criteria (Kiesler et al., 2023). Consequently, they may inadvertently provide feedback on criteria that are not included in the specific rubric they are instructed to use, which may confuse students and further diminish the actionability of their feedback.

Another key factor contributing to the superiority of instructor feedback in this study might be the instructor's rich contextual knowledge and awareness of individual students and classroom dynamics obtained through continuous interactions throughout the semester. This unique awareness, which is challenging to replicate in AI, provides instructors with a distinct advantage in the feedback provision process. Therefore, feedback comments from instructors are inherently aligned with the specific content and teaching methods used in the classroom (Gardner et al., 2016), which may ensure that feedback is relevant and easily digestible for students. In contrast, AI models lack information about students and classroom dynamics if they are not trained specifically to learn the context, and as a result, they may tend to produce rather generic feedback that lacks the personal connection and understanding often found in instructor feedback.

We consider two important implications of this research for the use of AI in feedback provision. First, the research findings strongly suggest that AI models should not be treated as one-size-fits-all solutions. Instead, they need to be trained on data that is specific to the educational context so that they can learn the particularities of a specific context wherein they provide feedback. These data must be comprehensive and include data about students, course content and instructors. Student data may incorporate information about students' strengths, weaknesses and progress over time, which can be measured through past assignments and quiz results. An important part of student data might be the instructor's input about the student based on his/her classroom observations and interactions, which may provide important insights to AI that cannot be captured from performance and progress data. Course content data may contain the syllabus, lecture notes and specific materials used in the course (such as assessment rubrics) to ensure their feedback aligns with the course content and learning objectives. The instructor data may include examples of the instructor's feedback (from previous semesters) and teaching methods to enable the AI to emulate the instructor's voice and approach, making the feedback more familiar and relatable to students. Thus, we suggest moving beyond generic language models and developing AI systems that can ingest and understand the unique characteristics of a course, students and course instructors. By immersing the AI in this contextual knowledge, we can potentially equip it to offer feedback that is more aligned with the specific needs and expectations of the students and the course itself.

As the second implication, the findings of this study suggest the development of hybrid feedback models that enable human–AI collaboration in feedback provision. Such models can bring together complementary strengths of both AI and instructors, thus representing a promising avenue for a scalable yet effective feedback practice. Several approaches can be adopted to leverage this synergistic connection between AI and instructors. For example, AI

can be used to provide an initial round of feedback, quickly identifying areas where students can improve. Instructors can then review this feedback, adding their own insights, clarifications and personalised guidance, ensuring its accuracy, relevance and alignment with the assessment criteria. Alternatively, AI can be employed to focus on specific aspects of student work, such as syntax errors and formatting, while instructors can then dedicate their expertise to providing higher-order feedback on program logic and code efficiency. Previous research has highlighted the importance of human–AI collaboration in education to foster a more engaging learning experience for students (Holstein et al., 2020).

CONCLUSION

This study investigated the effectiveness of AI-generated feedback compared to instructor feedback in an undergraduate Java programming course. While students perceived instructor feedback as more useful, there was no significant difference in other perception dimensions. Importantly, students receiving AI feedback showed significantly lower improvement in their lab assignment scores. This discrepancy is attributed to instructors' deeper contextual understanding and ability to tailor feedback, highlighting AI's current limitations in grasping assignment nuances and potential for hallucination. The study emphasises the need to train AI models on context-specific data and suggests the development of hybrid feedback models that leverage the strengths of both AI and human instructors. These findings contribute valuable insights into the evolving literature of generative AI in education, suggesting that LLMs, while promising, require further refinement and integration with human expertise to truly enhance the feedback process and maximise student learning.

This study has some limitations that need to be addressed in future research. First, we assumed that the feedback generated by the large language model was accurate and did not check its content. Although the large language model used in the study is known to work with programming languages, there is a possibility of hallucination or error in the feedback it generates. Second, in this study, the feedback generated by the AI is limited by the LLM and the prompt used. However, new models are emerging every day. In future studies, the performance of different language models or different prompt engineering methods in generating feedback can be examined. Models specifically fine-tuned for giving feedback can be compared with existing models. Moreover, this study relies on the data from 35 students, as not all students participated in all research activities. Although the research design was rigorous, a larger sample size would enhance the generalisability of the findings. Therefore, this study should be replicated with multiple programming assignments in different languages with a larger sample to enhance the generalisability of the findings.

FUNDING INFORMATION
This work was supported by the BAGEP Award of the Science Academy, Türkiye.

CONFLICT OF INTEREST STATEMENT
The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT
The datasets generated during and/or analysed during the current study are available from the corresponding author upon reasonable request.

ETHICS STATEMENT
This research was approved by the Institutional Review Board of Human Research at first author's university (Reference number: 0367-ODTUIAEK-2023).

ORCID

Erkan Er https://orcid.org/0000-0002-9624-4055
Gökhan Akçapınar https://orcid.org/0000-0002-0742-1612
Alper Bayazıt https://orcid.org/0000-0003-4369-587X
Omid Noroozi https://orcid.org/0000-0002-0622-289X
Seyyed Kazem Banihashem https://orcid.org/0000-0002-9978-3783

REFERENCES

Ajjawi, R., & Boud, D. (2018). Examining the nature and effects of feedback dialogue. *Assessment and Evaluation in Higher Education*, *43*(7), 1106–1119. https://doi.org/10.1080/02602938.2018.1434128

Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education*, *15*(2), 83–102. https://doi.org/10.1080/08993400500150747

Banihashem, S. K., Kerman, N. T., Noroozi, O., Moon, J., & Drachsler, H. (2024). Feedback sources in essay writing: Peer-generated or AI-generated feedback? *International Journal of Educational Technology in Higher Education*, *21*(1), 23. https://doi.org/10.1186/s41239-024-00455-4

Banihashem, S. K., Noroozi, O., Van Ginkel, S., Macfadyen, L. P., & Biemans, H. J. A. (2022). A systematic review of the role of learning analytics in enhancing feedback practices in higher education. *Educational Research Review*, *37*, 100489. https://doi.org/10.1016/j.edurev.2022.100489

Barzilai, S., & Blau, I. (2014). Scaffolding game-based learning: Impact on learning achievements, perceived learning, and game experiences. *Computers & Education*, *70*(January), 65–79. https://doi.org/10.1016/j.compedu.2013.08.003

Bazvand, A. D., & Rasooli, A. (2022). Students' experiences of fairness in summative assessment: A study in a higher education context. *Studies in Educational Evaluation*, *72*, 101118. https://doi.org/10.1016/j.stueduc.2021.101118

Bhullar, P. S., Joshi, M., & Chugh, R. (2024). ChatGPT in higher education—A synthesis of the literature and a future research agenda. *Education and Information Technologies*, *29*, 21501–21522. https://doi.org/10.1007/s10639-024-12723-x

Boud, D., & Molloy, E. (2012). Rethinking models of feedback for learning: The challenge of design. *Assessment & Evaluation in Higher Education*, *38*(6), 698–712. https://doi.org/10.1080/02602938.2012.691462

Camp, T., Adrion, W. R., Bizot, B., Davidson, S., Hall, M., Hambrusch, S., Walker, E., & Zweben, S. (2017). Generation CS: The growth of computer science. *ACM Inroads*, *8*(2), 44–50. https://doi.org/10.1145/3084362

Carless, D. (2006). Differing perceptions in the feedback process. *Studies in Higher Education*, *31*(2), 219–233. https://doi.org/10.1080/03075070600572132

Cavalcanti, A. P., de Mello, R. F. L., Rolim, V., André, M., Freitas, F., & Gašević, D. (2019). An analysis of the use of good feedback practices in online learning courses. In Proceedings of 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil (pp. 153–157).

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, *15*(3), 1–45. https://doi.org/10.1145/3641289

Denny, P., Luxton-Reilly, A., & Carpenter, D. (2014). Enhancing syntax error messages appears ineffectual. In Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education—ITiCSE '14, Uppsala, Sweden (pp. 273–278). https://doi.org/10.1145/2591708.2591748

Estévez-Ayres, I., Callejo, P., Hombrados-Herrera, M. Á., Alario-Hoyos, C., & Delgado Kloos, C. (2024). Evaluation of LLM tools for feedback generation in a course on concurrent programming. *International Journal of Artificial Intelligence in Education*, (2024) https://doi.org/10.1007/s40593-024-00406-0

Ezeamuzie, N. O. (2023). Project-first approach to programming in K–12: Tracking the development of novice programmers in technology-deprived environments. *Education and Information Technologies*, *28*(1), 407–437. https://doi.org/10.1007/s10639-022-11180-8

Foster, R. A., Wood, K. L., & Evans, M. H. (2024). The impact of multimedia feedback in blended learning environments on university students' programming skills. *Research and Advances in Education*, *3*(5), 42–52. doi:10.56397/RAE.2024.05.05

Fu, Q., Zheng, Y., Zhang, M., Zheng, L., Zhou, J., & Xie, B. (2023). Effects of different feedback strategies on academic achievements, learning motivations, and self-efficacy for novice programmers. *Educational Technology Research and Development*, *71*(3), 1013–1032. https://doi.org/10.1007/s11423-023-10223-2

Fyfe, E. R., & Rittle-Johnson, B. (2016). Feedback both helps and hinders learning: The causal role of prior knowledge. *Journal of Educational Psychology*, *108*(1), 82–97. https://doi.org/10.1037/edu0000053

Gan, Z., An, Z., & Liu, F. (2021). Teacher feedback practices, student feedback motivation, and feedback behavior: How are they associated with learning outcomes? *Frontiers in Psychology*, *12*, 697045. https://doi.org/10.3389/fpsyg.2021.697045

Gardner, E. E., Anderson, L. B., & Wolvin, A. D. (2016). Understanding instructor immediacy, credibility, and face-work strategies through a qualitative analysis of written instructor feedback. *Qualitative Research Reports in Communication*, *18*(1), 27–35. https://doi.org/10.1080/17459435.2016.1247113

Glassman, E. L., Scott, J., Singh, R., Guo, P. J., & Miller, R. C. (2015). OverCode: Visualizing variation in student solutions to programming problems at scale. *ACM Transactions on Computer-Human Interaction*, *22*(2), 1–35. https://doi.org/10.1145/2699751

Hao, Q., Smith Iv, D. H., Ding, L., Ko, A., Ottaway, C., Wilson, J., Arakawa, K. H., Turcan, A., Poehlman, T., & Greer, T. (2022). Towards understanding the effective design of automated formative feedback for programming assignments. *Computer Science Education*, *32*(1), 105–127. https://doi.org/10.1080/08993408.2020.1860408

Harks, B., Rakoczy, K., Hattie, J., Besser, M., & Klieme, E. (2013). The effects of feedback on achievement, interest and self-evaluation: The role of feedback's perceived usefulness. *Educational Psychology*, *34*(3), 269–290. https://doi.org/10.1080/01443410.2013.785384

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, *77*(1), 81–112. https://doi.org/10.3102/003465430298487

Hollingsworth, J. (1960). Automatic graders for programming classes. *Communications of the ACM*, *3*(10), 528–529. https://doi.org/10.1145/367415.367422

Holstein, K., Aleven, V., & Rummel, N. (2020). A conceptual framework for human–AI hybrid adaptivity in education. In I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, & E. Millán (Eds.), *Artificial intelligence in education* (Vol. *12163*, pp. 240–254). Springer International Publishing. https://doi.org/10.1007/978-3-030-52237-7_20

Jonsson, A. (2013). Facilitating productive use of feedback in higher education. *Active Learning in Higher Education*, *14*, 63–76. https://doi.org/10.1177/1469787412467125

Jonsson, A., & Panadero, E. (2018). Facilitating students' active engagement with feedback. In A. Lipnevich & J. Smith (Eds.), *The Cambridge handbook of instructional feedback (Cambridge handbooks in psychology)* (pp. 531–553). Cambridge University Press. https://doi.org/10.1017/9781316832134.026

Jukiewicz, M. (2024). The future of grading programming assignments in education: The role of ChatGPT in automating the assessment and feedback process. *Thinking Skills and Creativity*, *52*, 101522. https://doi.org/10.1016/j.tsc.2024.101522

Kerman, N. T., Banihashem, S. K., & Noroozi, O. (2023). The relationship among students' attitude towards peer feedback, peer feedback performance, and uptake. In O. Noroozi & B. De Wever (Eds.), *The power of peer learning* (pp. 347–371). Springer International Publishing. https://doi.org/10.1007/978-3-031-29411-2_16

Keuning, H., Heeren, B., & Jeuring, J. (2017). Code quality issues in student programs. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, Bologna, Italy, (pp. 110–115). https://doi.org/10.1145/3059009.3059061

Kiesler, N., Lohr, D., & Keuning, H. (2023). Exploring the potential of large language models to generate formative programming feedback. In 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA (pp. 1–5). https://doi.org/10.1109/FIE58773.2023.10343457

Kinnunen, P., & Simon, B. (2012). My program is ok—am I? Computing freshmen's experiences of doing programming assignments. *Computer Science Education*, *22*(1), 1–28. https://doi.org/10.1080/08993408.2012.655091

Liang, Y., Liu, Q., Xu, J., & Wang, D. (2009). The recent development of automated programming assessment. In 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, China (pp. 1–5). https://doi.org/10.1109/CISE.2009.5365307

Lipnevich, A. A., & Smith, J. K. (2009). Effects of differential feedback on students' examination performance. *Journal of Experimental Psychology: Applied*, *15*(4), 319–333. https://doi.org/10.1037/a0017841

Liu, Z., He, X., Liu, L., Liu, T., & Zhai, X. (2023). Context matters: A strategy to pre-train language model for science education. In N. Wang, G. Rebolledo-Mendez, V. Dimitrova, N. Matsuda, & O. C. Santos (Eds.), *Artificial intelligence in education. Posters and late breaking results, workshops and tutorials, industry and innovation tracks, practitioners, doctoral consortium and blue sky* (Vol. *1831*, pp. 666–674). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36336-8_103

Lizzio, A., & Wilson, K. (2008). Feedback on assessment: Students' perceptions of quality and effectiveness. *Assessment & Evaluation in Higher Education*, *33*(3), 263–275. https://doi.org/10.1080/02602930701292548

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, M. J., Sheard, J., & Szabo, C. (2018). Introductory programming: A systematic literature review. In Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, Larnaca, Cyprus, (pp. 55–106). https://doi.org/10.1145/3293881.3295779

MacKinnon, J. G., & White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, *29*(3), 305–325. https://doi.org/10.1016/0304-4076(85)90158-7

Maghsoudi, M. (2023). Uncovering the skillsets required in computer science jobs using social network analysis. *Education and Information Technologies*, *29*, 12759–12780. https://doi.org/10.1007/s10639-023-12304-4

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-. D., Laxer, C., Thomas, L., Utting, I., & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education, Canterbury, UK, (pp. 125–180). https://doi.org/10.1145/572133.572137

Nguyen, A., Piech, C., Huang, J., & Guibas, L. (2014). Codewebs: Scalable homework search for massive open online programming courses. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, (pp. 491–502). https://doi.org/10.1145/2566486.2568023

Nicol, D. (2013). Resituating feedback from the reactive to the proactive. In D. Boud & E. Molloy (Eds.), *Feedback in higher and professional education* (pp. 34–49). Routledge.

Noroozi, O., Alqassab, M., Taghizadeh Kerman, N., Banihashem, S. K., & Panadero, E. (2024). Does perception mean learning? Insights from an online peer feedback setting. *Assessment & Evaluation in Higher Education*, 1–15. https://doi.org/10.1080/02602938.2024.2345669

Panadero, E., Alqassab, M., Fernández Ruiz, J., & Ocampo, J. C. (2023). A systematic review on peer assessment: Intrapersonal and interpersonal factors. *Assessment & Evaluation in Higher Education*, *48*(8), 1053–1075. https://doi.org/10.1080/02602938.2023.2164884

Porat, E., Blau, I., & Barak, A. (2018). Measuring digital literacies: Junior high-school students' perceived competencies versus actual performance. *Computers & Education*, *126*(November), 23–36. https://doi.org/10.1016/j.compedu.2018.06.030

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference, 57*, Austin, TX (pp. 92–96).

Shirah, J. F., & Sidney, P. G. (2023). Computer-based feedback matters when relevant prior knowledge is not activated. *Learning and Instruction*, *87*, 101796. https://doi.org/10.1016/j.learninstruc.2023.101796

Smits, M. H. S. B., Boon, J., Sluijsmans, D. M. A., & Van Gog, T. (2008). Content and timing of feedback in a web-based learning environment: Effects on learning as a function of prior knowledge. *Interactive Learning Environments*, *16*(2), 183–193. https://doi.org/10.1080/10494820701365952

Strijbos, J.-W., Narciss, S., & Dünnebier, K. (2010). Peer feedback content and sender's competence level in academic writing revision tasks: Are they critical for feedback perceptions and efficiency? *Learning and Instruction*, *20*(4), 291–303. https://doi.org/10.1016/j.learninstruc.2009.08.008

Tian, L., & Zhou, Y. (2020). Learner engagement with automated feedback, peer feedback and teacher feedback in an online EFL writing context. *System*, *91*, 102247. https://doi.org/10.1016/j.system.2020.102247

Van Der Kleij, F. M., & Lipnevich, A. A. (2021). Student perceptions of assessment feedback: A critical scoping review and call for research. *Educational Assessment, Evaluation and Accountability*, *33*(2), 345–373. https://doi.org/10.1007/s11092-020-09331-x

Xu, Z., Jain, S., & Kankanhalli, M. (2024). *Hallucination is inevitable: An innate limitation of large language models (version 1). arXiv:2401.11817.* https://doi.org/10.48550/ARXIV.2401.11817

Zhang, Z. (V.), & Hyland, K. (2018). Student engagement with teacher and automated feedback on L2 writing. *Assessing Writing*, *36*, 90–102. https://doi.org/10.1016/j.asw.2018.02.004

Zhang, Z., Dong, Z., Shi, Y., Price, T., Matsuda, N., & Xu, D. (2024). Students' perceptions and preferences of generative artificial intelligence feedback for programming. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(21), 23250–23258. https://doi.org/10.1609/aaai.v38i21.30372

Zimbardi, K., Colthorpe, K., Dekker, A., Engstrom, C., Bugarcic, A., Worthy, P., Victor, R., Chunduri, P., Lluka, L., & Long, P. (2017). Are they using my feedback? The extent of students' feedback use has a large impact on subsequent academic performance. *Assessment & Evaluation in Higher Education*, *42*(4), 625–644. https://doi.org/10.1080/02602938.2016.1174187

Zougari, S., Tanana, M., & Lyhyaoui, A. (2016). Hybrid assessment method for programming assignments. In 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, Morocco, (pp. 564–569).

How to cite this article: Er, E., Akçapınar, G., Bayazıt, A., Noroozi, O., & Banihashem, S. K. (2025). Assessing student perceptions and use of instructor versus AI-generated feedback. *British Journal of Educational Technology*, *56*, 1074–1091. https://doi.org/10.1111/bjet.13558

APPENDIX

AN EXAMPLE OF INSTRUCTOR AND AI FEEDBACK

An example of instructor feedback	An example of ChatGPT feedback
Basic functionality (completeness) It is great that your program gets user input for lower and upper limits. Your program does not repeatedly ask for upper limit when the range is less than 20. Instead of 'if statement', you could use do/while or while loop to keep asking the user to enter the upper limit until the range (the difference between upper and lower limit) is 20 or more. Random number was generated properly based on the final upper and lower limits. The program prints a proper feedback message. The program prints a congrats message with correct number of trials. **Advance functionality (completeness)** You used the loop nicely to limit the number of attempts to 5. The program nicely saves the results and also prints the previous game results. Good job on that! The program partially handles the 'pressing the Enter key condition' for ending the game. Before the for loop, and even before the parseInt statement, you should have checked if the userInput is empty. The string objects have empty() methods for this purpose. If the userGuess is not empty, then the parseInt and for statements should be executed. Otherwise, the game should have ended. The program nicely prints an end message when the game is lost. **Modularity** displayGameResults method is implemented properly. saveResults method was implemented partially. This method should receive the content as a string parameter (such as '5,1'). Then this content variable should be passed to the println statement. You have it empty in your code: outputFile.println(). generateNumber method is implemented nicely. **Design of logic** The for loop should move outside the do-while loop. Actually, there is even no need for a do-while loop just for checking if the userInput is empty. You could do this check inside the for loop as well. Inside the for loop, you should still use nextLine instead of nextInt. If it is empty, then you could do the same thing as you do with guessCount = 4. Syntax and Definition Errors No syntax or definition errors are found. Great job on this! **Code efficiency** Because of nested loops and asking for the user guess at two different places, code efficiency is not excellent. But still a quite good job! **Readability** Code is clean, understandable and well-organised. **Overall** In overall, very good job, congrats! You have done a great job. You have an advance understanding of the concepts we have learned so far. With a bit more push, you could get an excellent grade for this assignment.	**Basic functionality (completeness)** Your submission does a good job with the basic functionalities: You correctly implemented the functionality for setting the lower and upper limits with proper validation for the range. This is fundamental and you have handled it well. The random number generation appears to function correctly within the specified limits, which is crucial for the gameplay. However, improvements can be made in the following areas: It seems that your game allows the user to guess indefinitely. Remember, the basic functionality should allow the user to guess until the number is correctly guessed, but there should be an upper limit on guesses as specified under the advanced functionality. **Advanced functionality (completeness)** The implementation of limiting the guesses to five attempts before revealing the answer is missing. This is an important part of the game logic to ensure it does not continue indefinitely under the advanced requirements. There is no functionality to handle the scenario when a user presses Enter without entering a guess, which should end the game. Make sure to add this to handle all user inputs correctly. **Modularity** You have some methods like generateNumber well defined, which is great for code organisation. Consider adding more methods for other functionalities like input validation, guessing logic and result handling to make your code more modular. **Design of logic** The logical flow of your application needs some attention. For example, ensure that your guessing logic is robust and correctly handles all possible scenarios, such as incorrect inputs or unexpected user actions. Consider refining the logic for handling the number of guesses. A loop with a clearly defined exit condition would be more appropriate here. **Syntax and definition errors** There are some minor syntax issues in your code that need correction. Ensure that all your loops and conditionals are syntactically correct to avoid compilation errors. Check your variable declarations and ensure they are defined at appropriate scope levels to avoid issues during runtime. **Code efficiency** To enhance code efficiency, ensure that your methods and logic are optimised to perform their tasks with the least amount of processing required. For example, the number generation method should be called only once per game unless a re-generation is required by game rules. **Readability** Your code would benefit from improved formatting and commenting. Proper indentation, consistent naming conventions and comments explaining the purpose of complex blocks of code will greatly improve the readability and maintainability of your code. Overall, you have made a commendable effort in implementing the number-guessing game. Please consider the above points to refine your code further. These improvements will not only make your game more robust and user-friendly but also enhance your coding skills. Keep up the good work!