Ordu Üniversitesi Bilim ve Teknoloji Dergisi
Ordu Üniv. Bil. Tek. Derg.

Ordu University Journal of Science and Technology
Ordu Univ. J. Sci. Tech.

E-ISSN: 2146-6459 2025, 15 (1), 128-141 Araştırma Makalesi / Research Article

https://doi.org/10.54370/ordubtd.1691860

When Class Time Falls Short: An Alternative Path to Application-Based Learning with GenAI

Necati Taşkın[1]

[1] Ordu University, Vocational School of Technical Sciences, Computer Technologies, Ordu

Geliş Tarihi / Received Date: 05.05.2025 Kabul Tarihi / Accepted Date: 21.06.2025

Abstract

This study examines the impact of GenAI (Generative Artificial Intelligence) supported application-based learning on high school students' academic achievement and course perceptions in the programming languages course. The study was conducted over six weeks with 77 10th-grade students in a public high school. A quasi-experimental design was used, involving two experimental groups and one control group. While one experimental group engaged in application-based activities under teacher guidance in class, the other completed the same activities at out-of-class using ChatGPT prompts. The control group followed the standard curriculum. Quantitative data were collected using an achievement test and course evaluation scale. One-way ANOVA results indicated no statistically significant difference in academic achievement among the groups. But the mean scores of the students in the experimental groups were higher than the control group. Moreover, students in both experimental groups reported significantly more positive course perceptions compared to the control group, particularly in the dimensions of course, instructor, and method-technique. Furthermore, while a weak positive correlation was found between course perception and academic achievement, it was not statistically significant. The findings highlight that although short-term academic gains may not differ significantly, both in-class application-based activities and GenAI-supported out-of-class activities enhance students' perception of the course. The study underscores the potential of GenAI tools as pedagogical aids in promoting active learning, especially when in-class application time is limited. It suggests increasing the number of application-based course hours in the curriculum and emphasizes that, in cases where this is not possible, GenAI-supported out-of-class activities can be considered as an alternative.

Keywords: application-based learning, generative artificial intelligence (GenAI), programming education, course perception, academic achievement

Ders Zamanı Yetmediğinde: GenAI ile Uygulama Tabanlı Öğrenmeye Alternatif Bir Yol

Öz

Bu çalışmada GenAI (Üretken Yapay Zekâ) destekli öğrenmenin lise öğrencilerinin programlama dilleri dersindeki akademik başarılarına ve ders algılarına etkisi incelenmektedir. Çalışma, bir devlet lisesindeki 77 onuncu sınıf öğrencisi ile altı hafta boyunca yürütülmüştür. İki deney grubu ve bir kontrol grubunun yer aldığı yarı deneysel bir tasarım kullanılmıştır. Bir deney grubu, sınıfta öğretmen rehberliğinde uygulamalı programlama etkinliklerine katılırken, diğeri aynı etkinlikleri ChatGPT tarafından oluşturulan istemleri kullanarak sınıf dışında tamamlamıştır. Kontrol grubu standart müfredatı takip etmiştir. Nicel veriler, bir başarı testi ve ders değerlendirme ölçeği kullanılarak toplanmıştır. Tek yönlü ANOVA sonuçları, gruplar arasında akademik başarıda istatistiksel olarak anlamlı bir fark olmadığını göstermiş olsa da deney grubundaki öğrencilerin puanları daha yüksektir. Ayrıca, her iki deney grubundaki öğrenciler ders, öğretmen ve yöntem-teknik boyutlarında, kontrol grubuna kıyasla daha olumlu ders algısına sahiptir. Ders algısı ile akademik başarı arasında anlamlı olmayan zayıf bir pozitif ilişki bulunmuştur. Bulgular, kısa vadeli akademik kazanımların önemli ölçüde farklılık göstermeyebileceğini vurgulasa da hem sınıf içi etkinlilerin hem de GenAI destekli sınıf dışı etkinliklerin öğrencilerin ders algısını geliştirdiğini ortaya koymaktadır. Çalışma, özellikle sınıfta içi uygulama süresi sınırlı olduğunda, GenAI araçlarının aktif öğrenmeyi teşvik etmedeki pedagojik potansiyeline işaret etmektedir. Bu bağlamda bu çalışma uygulama tabanlı sınıf içi etkinliklerin artırılmasını ve mümkün olmadığı durumlarda GenAI destekli etkinliklerin alternatif olarak değerlendirilebileceğini vurgulamaktadır.

Anahtar Kelimeler: uygulama tabanlı öğrenme, üretken yapay zekâ (GenAI), programlama eğitimi, ders algısı, akademik başarı

Sorumlu Yazar / Corresponding Author: Necati Taşkın necatitaskin@odu.edu.tr

Introduction

As digital technologies permeate every aspect of life, it has become important for students to not only use these tools but also interact with them from a critical and creative perspective. In this context, digital literacy stands out as a multidimensional competence area that encompasses students' skills in accessing, using, producing, and sharing information through digital tools (Reichert et al., 2020). One of the fundamental components of digital literacy is software literacy. Vee (2017) demonstrates that software literacy (coding literacy) has become a universal competence area, just like reading and writing, and that this is necessary for individuals to be active in the digital world. Khoo et al. (2017) considers software literacy a basic digital citizenship requirement. Software literacy is regarded as a structure that can transform the individual's cognitive processes and modes of production. This understanding reveals an approach that encompasses not only the knowledge of programming languages but also the critical evaluation of their possibilities and limitations. Rush Hovde & Renguette (2017) state that programming languages improve students' technological literacy as well as their learning to learn, evaluation, and critical thinking competencies. In this context, the "Information Technologies and Software" course taught at a high school level aims to develop students' software literacy skills (MEB, 2023).

Within the scope of this course, ninth-grade students take the "*Introduction to Programming and Algorithms*" module, followed by modules such as "*Programming Languages*", "*Robotic Coding*", or "*Mobile Application Development*" in tenth grade. This study is based on the "*Programming Languages*" module. Providing only theoretical knowledge is not enough to develop programming skills. It is essential to provide students with application-based activities to support their learning and skill development (MEB, 2023). However, the weekly in-class course hours are limited to two. This in-class hour limitation does not provide students with sufficient practice opportunities in application-based courses such as programming languages. Programming languages education should not be limited to teaching only the basic rules of coding. It should also be structured as a learning process that allows students to understand for what purposes and how they can use this language effectively (Rush Hovde & Renguette, 2017). The fact that students receive education through application-based learning, where they interact with software, is a determining factor in their software literacy skill (Reichert et al., 2020). In this context, it is an important necessity to develop different application-based approaches and to examine the quality of these approaches. To overcome the limited in-class time for practice, one effective way is to make use of new technologies that support learning.

In programming education, students often learn best through application-based experiences. Activities such as designing simple projects like quizzes, calculators, or games using platforms like Python or Scratch help students engagingly understanding core concepts. These kinds of tasks allow them to practice writing code, spotting and fixing errors, and applying basic structures like loops and conditionals (Medeiros et al., 2019). In many classrooms, students collaborate in pairs or small groups to solve problems together, which supports both learning and motivation. Research also shows that such application-based approaches can be effectively adapted for younger learners by integrating coding into creative activities like storytelling, music, and art (Macrides et al., 2022). Among these, Generative Artificial Intelligence (GenAI) tools have become increasingly popular in programming education. When supported by GenAI tools like ChatGPT, students can take these activities further, asking questions about code, getting help when stuck, and receiving instant feedback. These interactions help make programming feel more personal and accessible, and they encourage active participation and engagement (Åkerfeldt et al., 2024).

GenAI tools can support students' learning according to their individual needs, provide real-time feedback, and make learning processes more interactive (Hsu & Ching, 2023). The personalized learning opportunities offered by these tools make learning more effective by providing guidance for students' cognitive processes, especially in application-based courses (Kadaruddin, 2023). Well-structured prompts enable students to understand complex concepts, recognize their mistakes, and access accurate information (Bozkurt & Sharma, 2023). In this context, GenAI tools can partially compensate for the lack of application-based activities by taking on a guiding role in courses such as programming languages (Yılmaz & Yılmaz, 2023). GenAI tools like ChatGPT can act as virtual tutors, providing students with information, explaining, giving examples, and providing step-by-step guidance (Baidoo-Anu & Ansah, 2023). This feature can provide support for home-based tasks carried out outside the classroom and help students structure their learning process. Moreover, it becomes possible for students to diagnose their learning gaps and receive effective feedback (Dai et al., 2023). Thus, an environment that supports the individual development of students can be created even in learning environments devoid of teacher guidance.

Although GenAI has significant potential to improve learning and teaching processes (Bahroun et al., 2023), there are substantial shortcomings in integrating these tools into educational environments (Cooper, 2023). Lodge et al. (2023) state that there are still many questions to be resolved regarding the effects of GenAI, and the need for evidence regarding the benefits of these tools continues to grow. It is seen that studies on GenAI in education mostly focus on technical structures, and the pedagogical dimension is not taken into sufficient consideration (Dogan et al. 2023). GenAI should be embraced as here to stay (Lim et al., 2023) and integrated into all levels of education (Baidoo-Anu & Ansah, 2023). Although recent research has explored the technical capabilities and potential risks of GenAI tools in education, many of these studies primarily focus on system performance, ethical concerns, or usage patterns rather than pedagogical impact. For example, questions around bias, data privacy, and the reliability of AI-generated content are frequently addressed, yet there is limited attention to how these tools can be meaningfully integrated into teaching and learning processes (Cooper, 2023; Dogan et al., 2023; Lodge et al., 2023). In contrast, this study highlights the pedagogical value of GenAI-supported application-based learning in programming education and aims to contribute to a growing but still underdeveloped body of work focused on instructional design and student engagement. In this context, this study will not only reveal the impact of application-based learning but also provide important clues on how GenAI-supported learning can be used more functionally in educational environments.

Purpose of Research

In this study, two different methods were considered: Application-based in-class activities and GenAI-supported out-of-class activities. In this context, it is aimed to comparatively examine the effects of two different methods developed for the need for application-based learning within the scope of the "*Programming Languages*" module on students' academic achievement and course perceptions. The study sought to answer the following research questions:

1. Do different methods (application-based in-class activities and GenAI-supported out-of-class activities) significantly affect students' academic achievement scores?
2. Do different teaching methods significantly differ students' course perception scores?
3. Is there a significant relationship between students' course perception scores and academic achievement scores?

Method

This research was conducted in a high school affiliated with the Ministry of National Education in Ordu province in the 2024-2025 academic year. The study was conducted in three different classes taking the "*Programming Languages*" module of the "*Information Technologies and Software*" course in the 10th grade. The research process lasted a total of 6 weeks.

The research was designed within the quasi-experimental design, and a control group experimental design was used (Büyüköztürk et al., 2024). In this design, a control group and two different experimental groups were included, and the effects of different methods on students' academic achievement and course perceptions were examined. The groups were matched based on the scores of the students in the "*Introduction to Programming and Algorithms*" module they took in the first term. No statistically significant difference was found between the groups in terms of grade point averages (F(2,74) =0.31, p=.734). One of the matched groups was randomly assigned to the control group, and the other two to the experimental group. Quantitative data were collected to answer the research questions within the scope of this study. The achievement test developed by the researcher was used to measure academic achievement, and the Course Evaluation Scale (CES) developed by Koç and Bulut (2022) was used to determine students' perceptions of the course.

Table 1. Research Design

Group	Matching	Treatment	Post-test
Experiment 1	M	X_1	O_1, O_2
Experiment 2	M	X_2	O_1, O_2
Control	M	X_3	O_1, O_2

X_1= application-based in-class activities, X_2= GenAI supported out-of-class activities, X_3= curriculum based standard teaching O_1= Academic achievement test, O_2=Course evaluation scale

Study Group

The study group of the research consists of a total of 77 students studying in three different classes at the 10th grade level, taking the "*Information Technologies and Software*" course's "*Programming Languages*" module in the second term of the 2024-2025 academic years. The study was carried out with the voluntary participation of the students. Before the experimental procedure, it was clearly stated to the students that the study would be conducted for scientific purposes and that the data obtained would be analyzed at the group level without including personal information. This study used convenience sampling, as the participants were selected from existing classes that were already taking the course as part of their regular school schedule. These classes were included in the study because they were readily accessible and directly related to the research, rather than being chosen at random. The distribution of students into groups is presented in Table 2.

Table 2. Demographic Information of Participants

	Male		Female		Total	
Group	n	%	n	%	n	%
Experiment 1	4	15.4	22	8.6	26	33.7
Experiment 2	7	29.2	17	70.8	24	31.2
Control	5	18.5	22	81.5	27	35.1

Research Procedure

This study was conducted within the scope of the "*Information Technologies and Software*" course (Figure 1). The courses were conducted by the same teacher in line with the curriculum plan (MEB, 2023) to ensure instructional consistency.

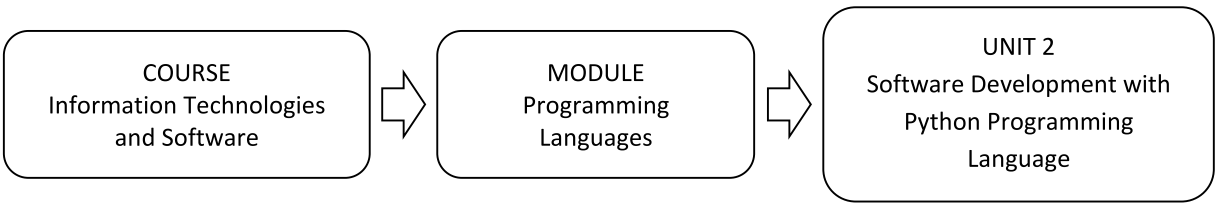

Figure 1. Link between Course, Module and Unit

In all three groups, two-hour lessons were conducted with the same content and method in line with the curriculum. The course aims to develop students' algorithmic thinking, problem-solving, and basic programming skills. In this context, students develop applications using basic programming structures such as variables, decision structures, loops, and learn to produce solutions to the problems they encounter during the coding process. The Python programming language is used within the scope of the course. One in-class lesson hour was added to the first experimental group. In this lesson, students performed application-based activities in the classroom under the guidance of the teacher (Table 3).

Table 3. Application-Based Activities

Week	Activities	Level	Content of Activities
1	Age Calculator	Easy	input(), int(), if, print()
2	Number Sequence Processing	Easy	list, for, if, %
3	Mini Grade Evaluation System	Medium	if-elif-else, input(), int(), str()
4	Calculator App	Medium	while, def, input(), if-else
5	Number Guessing Game	Difficult	random.randint(), while, if
6	Hangman Game	Difficult	string, list, for/while, in, if, break

The second experimental group was given home-based tasks for the application-based activities that the first experimental group did in class. In these tasks, students were guided through the learning process with ChatGPT prompts. Students completed their tasks individually using prompts. This method provided the opportunity to test the potential of GenAI tools (ChatGPT) to take on a role supporting the learning process. Examples of prompts presented to students are given in Table 4.

Table 4. Number Guessing Game Prompts

Stage	Prompts	Content
Stage 1. Game Description		
Prompt 1.1	I want to generate a random number between 1 and 100. How do I do this in Python?	Random module, input () function, int () conversion.
Prompt 2.1	I want the user to enter a number. What should I do to make sure it is received as a number?	
Stage 2: Comparison		
Prompt 2.1	I want to compare the user's guess with the secret number. How do I check if it is equal, greater than or less than?	Use of if, elif, else blocks. Logical comparison operators (==, <, >).
Prompt 2.2	If the guess is large, I want it to say "*Enter a smaller number.*" How do I do that?	
Stage 3: Game Cycle		
Prompt 3.1	Let the game repeat until the user guesses correctly. Which loop and what condition should I use?	Using the while loop and loop control variables.
Prompt 3.2	When you guess correctly, exit the loop and give the message "*Congratulations!*" How do I do it?	
Stage 4: Number of Trials		
Prompt 4.1	I want to count how many times the user guessed. How do I do that?	Counter variable (counter += 1), use of variable in message (f-string, format())
Prompt 4.2	I want to show the user a message showing how many times she guessed the correct answer.	
Stage 5: Error Management		
Prompt 5.1	If the user enters letters instead of numbers, an error occurs. How can I catch this error?	Error catching (try-except).
Prompt 5.2	I want to warn the user when an incorrect entry is made. How can I do this with "*try-except*"?	Input control using, ValueError.
Stage 6: Control and Feedback		
Prompt 6.1	The full code I wrote is below. Can you find my mistakes and give me suggestions?	Code review habit, ability to receive feedback and correct.

In the control group, courses were taught in accordance with the curriculum plan and designated weekly course hours (2 hours of lessons per week). This group followed the standard instructional approach defined by the national curriculum, which primarily emphasizes theoretical explanations, textbook-based learning, and teacher-led instruction. No additional practice or application-based activities were provided beyond the textbook examples and in-class discussions. Students repeated code written by the teacher and had limited opportunity to code independently. In this study, the effects of (1) application-based in-class activities, (2) GenAI-supported out-of-class activities, and (3) curriculum-based standard teaching process on students' academic achievement and course perceptions were comparatively examined.

Data Collection Tools

In the study, a multiple-choice test was developed to measure students' academic performance in the "*Software Development with Python Programming Language*" unit. The test consisted of 20 items, each item including one correct answer and three distractors (four options in total). Test items were aligned with the learning outcomes specified in the national curriculum and were limited to the content covered during the 6-week instructional period. The items were initially developed by the teacher and reviewed by three university experts in the fields of Information Technologies, Computer Engineering, and Educational Measurement and Evaluation. Based on expert feedback, necessary revisions were made to ensure content and construct validity. The test was piloted with a separate sample of 90 students who received the same training the previous year. As a result of item analysis, the test demonstrated acceptable psychometric properties. The KR-20 reliability coefficient was calculated as 0.77, indicating acceptable internal consistency (Başol, 2019). The mean item difficulty was 0.72, showing that most items were moderately easy, and the standard error was 1.86. The test means score was 18.11 with a standard deviation of 3.93. These findings support the reliability and validity of the test for use in the current study. The final version aimed to evaluate students' programming knowledge, algorithmic thinking, problem-solving skills, and their ability to understand and apply coding constructs such as variables, loops, and conditional statements. Sample items for the test are given in Table 5.

Table 5. Sample Test Items

Item	Bloom's Taxonomy of Cognitive Domains	Content	Explanation
What value will be printed to the screen when the following Python code is run? x = 7 y = x + 3 print (y * 2) A) 10 B) 14 C) 20 D) 24	Understanding	Defining variables and performing operations	The student is expected to understand the operations in the given code block and find the result by understanding the values of the variables.

When the following Python code block is run, what will be displayed on the screen if the user enters 13?
age = int (input ("Enter your age: "))
if age >= 18:
    print ("You are an adult.")
else:
    print ("You are not an adult.")
A) You are an adult.
B) You are not an adult.
C) Gives an error.
D) It doesn't write anything on the screen.
	Applying	Use of decision Structure (if-else)	The student is expected to make input-output connections using a decision structure and analyze the operation of this structure on a real scenario.

The Course Evaluation Scale (CES) developed by Koç and Bulut (2022) was used to determine the students' perception levels towards the course. The scale has a 5-point Likert-type scale. The scale consists of 24 items and 4 sub-factors. Course (10 items), Instructor (7 items), Method-Technique (4 items), Exam (3 items). The minimum score that can be obtained from the scale is 24, and the maximum score is 120. High scores obtained from the scale indicate that students perceive the quality of the course positively. The overall Cronbach's Alpha coefficient of the scale is 0.94, and these coefficients for the sub-factors are reported as 0.93 (course), 0.91 (instructor), 0.86 (method-technique), 0.93 (assessment and evaluation). The factor structure of the scale explains 69.3% of the total variance, and the factor loadings vary between 0.47 and 0.98. The values obtained from the fit indices show that the model has a good level of fit ($X^2/df = 1.98$, RMSEA = 0.063, NFI = 0.90, GFI = 0.86, CFI = 0.94, IFI = 0.98, TLI = 0.94). The fact that the item-total correlations are in the range of 0.49–0.90 indicates that the scale is strong in terms of internal consistency and discrimination (Koç & Bulut, 2022).

Data Analysis

Statistical calculations were made on the data collected within the scope of the research using quantitative data analysis methods in line with the research questions. In the analysis of the data, firstly, the assumptions of normality and variance homogeneity were tested, and parametric tests were used (Büyüköztürk, 2024).

One-way ANOVA was applied to determine the effects of different teaching methods on both students' academic achievements and perception scores of the course. The Scheffe test was used when variance homogeneity was provided, and the Games-Howell multiple comparison test was used when variance homogeneity was not provided to determine which groups had significant differences between the groups. This analysis was carried out in the context of perceptions regarding the sub-factors of the scale (course, instructor, method-technique, assessment and evaluation). Pearson Correlation Coefficient was used to examine the relationship between students' course perception scores and academic achievement scores.

Findings

Findings Regarding Students' Academic Achievements

Descriptive statistics for achievement test score of students taking courses with three different methods: Application-based in-class activities (Experiment 1), GenAI supported out-of-class activities (Experiment 2), and curriculum based standard teaching (Control) are given in Table 6.

Table 6. Descriptive Statistics for Academic Achievement Test Scores

Group	N	M	SD	Skewness	Kurtosis
Experiment 1	26	80.19	12.23	-0,33	-0,31
Experiment 2	27	79.63	8.39	-0,96	0,92
Control	24	75.42	12.41	-1,08	1,13

While the academic achievement mean score of the students in the Experiment 1 and Experiment 2 groups was found to be quite close to each other (80.19 and 79.63), the mean score of the students in the control group was determined to be lower (75.42). The results of the ANOVA test conducted to determine whether this difference was statistically significant are presented in Table 7.

Table 7. ANOVA Results of Academic Achievement Test Scores

Variance Source	Sum of Squares (SS)	Degrees of Freedom (df)	Mean of Squares (MS)	F	p
Between Groups	337.079	2	168.539	1.203	0.306
Within Groups	10366.168	74	140.083		
Total	10703.247	76			

The distribution of the variables was examined based on skewness and kurtosis, and the results (see Table 6) indicated that the data were within acceptable limits (±1,5) for normality (Tabachnick & Fidell, 2019). According to the results of the Levene test performed before the analysis, it was determined that the variances were homogeneous (Levene Statistic = 0.631; p = 0.535). As a result of the one-way ANOVA analysis, no statistically significant difference was found between the groups in terms of academic achievement scores of students (F(2,74) = 1.203, p = 0.306).

Findings Regarding Students' Course Perception

The Course Evaluation Scale (CES) scores of students who received education according to different teaching methods are presented in Table 8.

Table 8. Course Evaluation Scale Mean Scores

	Course		Instructor		Method-Technique		Assessment and Evaluation		Total	
	M	SD	M	SD	M	SD	M	SD	M	SD
Experiment 1	3.91	0.43	4.80	0.21	4.68	0.18	4.47	0.62	4.34	0.25
Experiment 2	4.21	0.64	4.89	0.21	4.98	0.09	4.74	0.37	4.56	0.33
Control	3.43	0.62	4.54	0.43	4.13	0.33	4.21	0.55	3.91	0.37

When both the general scores and the sub-factor scores related to perception of the students are examined, it is seen that the students in the experiment 2 group have the highest scores. While the scores of the experiment 1 group are close to the experiment 2 group, it was determined that the students in the control group have lower perception scores than both experimental groups in all factors. It was observed that the students' scores were close to normal distribution (Table 9) and the group variances were homogeneous (Levene Statistic = 2.218, p = 0.116). The course perception scores of students in different groups are given in Table 10.

Table 9. Skewness and Kurtosis Values of Students' CES Scores

	Skewness		Kurtosis	
	Statistic	Std. Error	Statistic	Std. Error
Experiment 1	-0.68	0.47	-0.34	0.92
Experiment 2	0.01	0.46	-0.86	0.89
Control	-1.10	0.45	0.81	0.87

Table 10. ANOVA Results regarding General Course Perception Scores

Variance Source	Sum of Squares (SS)	Degrees of Freedom (df)	Mean of Squares (MS)	F	p	Post Hoc
Between Groups	5.446	2	2.723	26.977	0.000	C-E1*
Within Groups	7.470	74	0.101			C-E2*
Total	12.916	76				

Note: Significant at p < .01 level, C= Control group, E1= Experiment 1, E2= Experiment 2

As a result of ANOVA analysis, it was found that there was a statistically significant difference between the groups in terms of general course perception scores (F (2,74) = 26.977, p < 0.001). According to the Post-Hoc analysis results, the general course perception scores of the students in both experiment 1 (application-based in-class activities) and experiment 2 (GenAI supported activities) groups were significantly higher than those of the students in the control group (curriculum-based standard teaching). The ANOVA results regarding the students' scores on the scale sub-factors are presented in Table 11.

Table 11. ANOVA Results regarding Scale Sub-factor Scores

Variance Source	Sum of Squares (SS)	Degrees of Freedom (df)	Mean of Squares (MS)	F	p	Post Hoc
Course						
Between Groups	7.919	2	3.959	12.102	0.000	C-E1**
Within Groups	24.210	74	0.327			C-E2*
Total	32.129	76				
Teacher						
Between Groups	1.621	2	0.810	9.183	0.000	C-E1**
Within Groups	6.530	74	0.088			C-E2*
Total	8.151	76				
Method-Technique						
Between Groups	9.508	2	4.754	98.804	0.000	C-E1*
Within Groups	3.560	74	0.048			C-E2*
Total	13.068	76				E1-E2*
Exam						
Between Groups	3.606	2	1.803	6.565	0.002	K-D2*
Within Groups	20.323	74	0.275			
Total	23.929	76				

*Note. * Significant at $p < .01$ level, ** Significant at $p < .05$ level, C= Control group, E1= Experiment 1, E2= Experiment 2*

As a result of ANOVA analysis, significant differences were found between the groups in the students' perception scores regarding the course and instructor sub-factors. According to the Post-Hoc analysis results, the course and instructor sub-factor scores of the students in the control group were significantly lower than those of the students in the experiment 2 group at the $p < .05$ level, and lower than those of the students in the experiment 1 group at the $p < .01$ level. In the method-technique sub-factor, a significant difference was found between the students in control group and both experimental groups in favor of the experimental groups ($p < .01$). Additionally, a significant difference was found between the students in the experiment 1 group and the students in the experiment 2 groups in this sub-factor, in favor of the experiment 2 group ($p < .01$). In the assessment and evaluation sub-factor, a statistically significant difference was found only between the students in group control and group experiment 2, in favor of experiment 2 ($p < .01$).

These findings reveal that different teaching methods have a significant effect on students' course perceptions. It was determined that the perceptions of the students in the experimental groups, instructor and method-technique sub-factors were more positive compared to the students in the control group.

Findings Regarding the Relationship between Students' Course Perception Scores and Academic Achievements Scores

Table 12. The Relationship between CES Scores and Academic Achievement Test Scores of Students

Variables	N	M	SD	r	p
Academic Achievement Test Scores	77	78.51	11.87	0.174	0.129
Course Evaluation Scale Scores	77	4.29	0.41		

As seen in Table 12, there is a weak positive relationship between the students' course evaluation scale (CES) scores and their academic achievement test scores ($r = 0.174$, $p = 0.129$). However, this relationship was not found to be statistically significant. This result shows that there is no direct relationship between the students' course perceptions and their academic achievement.

Discussion

Recent studies suggest that the impact of technology-supported and application-based learning may not be fully reflected in short-term measurement (Baidoo-Anu & Ansah, 2023; Hsu & Ching, 2023). In line with this, our findings indicate that although no significant difference was observed in academic performance, the trend favors application-based and GenAI-supported approaches. Reichert et al. (2020) also emphasized that individuals' familiarity with digital software tools can have decisive effects on students' performance. Therefore, the absence of a significant difference in achievement can be considered a result of short-term measurement. In the long term, it can be expected that academic achievement differences will become more pronounced as students become more accustomed to technology-supported learning styles and self-regulated processes.

In this study, students in both experimental groups reported significantly higher scores in general and sub-dimensions of course perception compared to the control group. These findings show that application-based and GenAI-supported learning processes strongly impact student course perceptions. This finding also supports the emphasis in the literature that software literacy and digital literacy are among the basic components of contemporary learning environments (Aydınlar et al., 2024; Reichert et al., 2020). Khoo et al. (2017) emphasize that students need to develop critical and conceptual perspectives beyond being merely digital tool users. The finding that GenAI-supported learning processes and application-based in-class activities increase students' course perception scores is consistent with the principle of active student participation in learning and taking responsibility for learning, as highlighted by constructivist learning theory (Duffy & Jonassen, 1992). The integration of information and communication technologies into the learning process makes it possible to implement constructivist practices more functionally (Kılıç Çakmak et al., 2017). Pavlik (2025) states that GenAI offers significant opportunities for designing participatory learning processes within the framework of constructivist theory. In this context, it becomes a pedagogical learning partner, turning students into active participants in learning environments.

The finding that GenAI-supported learning environments positively affect students' course perception is also parallel to the literature (Bahroun et al., 2023; Lim et al., 2023). Similarly, students tend to perceive GenAI as a personalized, time-efficient, and easily accessible learning assistant, reflecting a positive attitude towards its integration into education (Monib et al., 2025; Obenza et al., 2024). GenAI tools provide students with personalized learning experiences that meet their individual needs. It supports learning processes by providing continuous feedback through formative assessment practice. In the study conducted by Yilmaz and Yilmaz (2023), it was revealed that GenAI-supported education increased students' cognitive thinking skills, programming self-efficacy, and motivation.

In the curriculum-based control group, programming instruction was delivered through theoretical explanations and textbook-based materials. Control group, the instruction followed a teacher-centered "*demonstrate-and-follow*" model. While students repeated the codes written by the instructor, they had limited opportunities to independently write or modify code themselves due to time constraints. As a result, the process relied heavily on passive observation rather than active engagement in programming. Considering that curriculum-based standard teaching cannot meet the needs of students due to time and resource limitations, GenAI-based solutions like ChatGPT can provide an important alternative (Lodge et al., 2023). These tools can overcome time and space limitations by providing individualized guidance and increasing students' active participation in the learning process (Hsu & Ching, 2023). In addition, one of the most challenging situations for students during out-of-class activities is not being able to reach the teacher for support when needed. Lack of feedback negatively affects students' homework completion behaviors and time management, indirectly decreasing success (Núñez et al., 2015). In addition, teachers' inability to adequately recognize the difficulties experienced by students, combined with the lack of this support, leads to a more negative perception of the process from the student's perspective (Hong et al., 2011). At this point, providing GenAI support in out-of-class activities stands out as an important opportunity for students to access guidance when they need it. In this study, prompts were given to the student's ready-made. Providing accurate, clear, and relevant prompts is also a critical requirement for the effective use of GenAI tools (Bozkurt & Sharma, 2023). Students' possession of these skills will contribute to the deepening of learning by supporting critical thinking, problem solving, and creativity.

In conclusion, the findings of this study show that in-class application-based activities and GenAI-supported out-of-class activities, compared to curriculum-based instruction, strengthen students' course perceptions scores across all sub-dimensions (course, instructor, method-technique, and assessment). While no statistically significant differences were found in achievement scores among the groups, both experimental groups showed higher average scores than the control group. Although the short-term academic achievement effects may not yet be significant, they may become more visible over time as students adapt to more personalized and participatory learning methods. In this context, the integration of GenAI tools in education should be considered not only as a technological innovation but also as a pedagogical transformation, especially in programming instruction. Establishing a structure that encourages students to take responsibility for their learning and supports constructive and personalized learning experiences will ensure more effective and sustainable learning outcomes in educational environments.

Conclusion and Recommendations

The findings of this study show that application-based in-class activities and ChatGPT-supported activities strengthen students' course perceptions, but their effects on academic achievement are not evident in the short term. The increase in student perception highlights the importance of an environment that encourages individual participation and responsibility for learning within the framework of constructive learning theory. GenAI-supported learning applications stand out as an effective alternative, especially for educational environments with time and space constraints, by offering personalized guidance and instant feedback. In this context, the integration of GenAI tools into educational environments may be considered as a pedagogical way to support students with application-based learning experiences in out-of-class.

Based on the significant differences observed in students' course perception levels, this research recommends that application-based lesson hours be expanded in the curriculum. Students in both in-class activities and GenAI-supported learning conditions reported more positive perceptions of the course, instructor, and instructional methods, highlighting the need to foster active and engaging learning environments. Especially in courses such as "*Information Technologies and Software*", course weekly lesson hours can be increased, and application-based in-class activities opportunities can be provided. In cases where in-class hours are limited, moving application-based activities to the out-of-class environment supported by GenAI-based methods can be considered as an alternative. To encourage students' familiarity with software tools and the development of their software literacy, GenAI-supported activities can be planned from early on. In addition, teachers' pedagogical competencies can be improved so that they can provide effective prompts and sample application scenarios to their students. Additionally, for students to utilize GenAI tools, they must develop the skills to create accurate, clear, and relevant prompts when interacting with these tools. As a result, rather than seeing GenAI tools merely as a support tool, using them as a component that will strengthen students' active participation and meaningful learning processes will pave the way for permanent and sustainable gains in education.

Limitations and Future Research

This study has several limitations. First, the short-term measurement of academic achievement and course perception prevented the evaluation of long-term effects. The absence of a pre-test limits internal validity, as prior group equivalence could not be fully ensured despite matching based on previous course grades. Additionally, individual differences in students' familiarity with digital tools may have influenced learning outcomes. Conducting the study within a single course and grade level limits external validity and generalizability. Finally, relying solely on self-reported perception data suggests the need for more objective measures in future research.

Ethics Statement

There are no ethical issues with the publication of this article.

Conflicts of Interest

There are no conflicts of interest.

ORCID

Necati Taşkın https://orcid.org/0000-0001-8519-6185

References

Åkerfeldt, A., Kjällander, S., & Petersen, P. (2024). A research review of computational thinking and programming in education. Technology, *Pedagogy and Education, 33*(3), 375–390. https://doi.org/10.1080/1475939X.2024.2316087

Aydınlar, A., Mavi, A., Kütükçü, E., Kırımlı, E. E., Alış, D., Akın, A., & Altıntaş, L. (2024). Awareness and level of digital literacy among students receiving health-based education. *BMC Medical Education, 24*(1), 38. https://doi.org/10.1186/s12909-024-05025-w

Bahroun, Z., Anane, C., Ahmed, V., & Zacca, A. (2023). Transforming education: A comprehensive review of generative artificial intelligence in educational settings through bibliometric and content analysis. *Sustainability, 15*(17), 12983. https://doi.org/10.3390/su151712983

Başol, G. (2019). *Eğitimde ölçme ve değerlendirme*. Pegem Akademi Yayıncılık.

Baidoo-Anu, D., & Ansah, L. O. (2023). Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Journal of AI, 7*(1), 52-62. https://doi.org/10.61969/jai.1337500

Bozkurt, A., & Sharma, R. C. (2023). Generative AI and prompt engineering: The art of whispering to let the genie out of the algorithmic world. *Asian Journal of Distance Education, 18*(2), i-vii. https://doi.org/10.5281/zenodo.8174941

Büyüköztürk, Ş., (2024). *Sosyal bilimler için veri analizi el kitabı istatistik, araştırma deseni SPSS uygulamaları ve yorum.* Pegem Akademi Yayıncılık.

Büyüköztürk, Ş., Akgün, Ö. E., Demirel, F., Karadeniz, Ş., & Çakmak, E. K. (2024). *Eğitimde bilimsel araştırma yöntemleri.* Pegem Akademi Yayıncılık.

Cooper, G. (2023). Examining science education in ChatGPT: An exploratory study of generative artificial intelligence. *Journal of Science Education and Technology, 32*(3), 444-452. https://doi.org/10.1007/s10956-023-10039-y

Dai, Y., Liu, A., & Lim, C. P. (2023). Reconceptualizing ChatGPT and generative AI as a student-driven innovation in higher education. *Procedia CIRP, 119*, 84-90. https://doi.org/10.1016/j.procir.2023.05.002

Dogan, M. E., Goru Dogan, T., & Bozkurt, A. (2023). The use of artificial intelligence (AI) in online learning and distance education processes: A systematic review of empirical studies. *Applied Sciences, 13(*5), 3056. https://doi.org/10.3390/app13053056

Duffy, T.M., & Jonassen, D.H. (1992). *Constructivism and the technology of instruction: A conversation* (1st ed.). Routledge. https://doi.org/10.4324/9780203461976

Hong, E., Wan, M., & Peng, Y. (2011). Discrepancies between students' and teachers' perceptions of homework. *Journal of Advanced Academics, 22*(2), 280–308. https://doi.org/10.1177/1932202X1102200205

Hsu, Y. C., & Ching, Y. H. (2023). Generative artificial intelligence in education, part one: The dynamic frontier. *TechTrends, 67*(4), 603-607. https://doi.org/10.1007/s11528-023-00863-9

Kadaruddin, K. (2023). Empowering education through Generative AI: Innovative instructional strategies for tomorrow's learners. *International Journal of Business, Law, and Education, 4*(2), 618-625. https://doi.org/10.56442/ijble.v4i2.215

Khoo, E., Hight, C., Torrens, R. & Cowie, B. (2017). Software Literacy: Education and beyond. In *Software Literacy. Springer Briefs in Education* (pp. 91-99). Springer. https://doi.org/10.1007/978-981-10-7059-4_6

Kılıç Çakmak, E., Taşkın, N., & Kokoç, M., (2017). Öğrenme yaklaşımlarının öğretim teknolojileri alanındaki yeri ve önemi. In H. F. Odabaşı, B. Akkoyunlu, & A. İşman (Ed.), *Eğitim Teknolojileri Okumaları 2017* (pp. 451-465). TOJET- The Turkish Online Journal of Educational Technology. http://www.tojet.net/e-book/eto_2017.pdf

Koç, A., & Bulut, İ. (2022). Course evaluation scale (CES): Validity and reliability study. *Religious Studies, 25*(63), 501-528. https://doi.org/10.15745/da.1159706

Lim, W. M., Gunasekara, A., Pallant, J. L., Pallant, J. I., & Pechenkina, E. (2023). Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. *The International Journal of Management Education, 21*(2), 100790. https://doi.org/10.1016/j.ijme.2023.100790

Lodge, J. M., Thompson, K., & Corrin, L. (2023). Mapping out a research agenda for generative artificial intelligence in tertiary education. *Australasian Journal of Educational Technology, 39*(1), 1-8. https://doi.org/10.14742/ajet.8695

Macrides, E., Miliou, O., & Angeli, C. (2022). Programming in early childhood education: A systematic review. *International Journal of Child-Computer Interaction, 32*, 100396. https://doi.org/10.1016/j.ijcci.2021.100396

MEB, (2023). Ortaöğretim bilişim teknolojileri ve yazılım dersi: Öğretim programı. Milli Eğitim Bakanlığı. https://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=1663

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education, 62*(2), 77–90. https://doi.org/10.1109/TE.2018.2864133

Monib, W. K., Qazi, A., & Mahmud, M. M. (2025). Exploring learners' experiences and perceptions of ChatGPT as a learning tool in higher education. *Education and Information Technologies, 30*(1), 917-939. https://doi.org/10.1007/s10639-024-13065-4

Núñez, J. C., Suárez, N., Rosário, P., Vallejo, G., Cerezo, R., & Valle, A. (2015). Teachers' feedback on homework, homework-related behaviors, and academic achievement. *The Journal of Educational Research, 108*(3), 204–216. https://doi.org/10.1080/00220671.2013.878298

Obenza, B. N., Salvahan, A., Rios, A. N., Solo, A., Alburo, R. A., & Gabila, R. J. (2024). University students' perception and use of ChatGPT: Generative artificial intelligence (AI) in higher education. *International Journal of Human Computing Studies, 5*(12), 5-18. https://ssrn.com/abstract=4724968

Pavlik, J. V. (2025). Considering the pedagogical benefits of generative artificial intelligence in higher education: applying constructivist learning theory. In *Generative AI in Higher Education* (pp. 46-58). Edward Elgar Publishing. https://doi.org/10.4337/9781035326020.00014

Reichert, F., Zhang, D., Law, N. W., Wong, G. K., & de la Torre, J. (2020). Exploring the structure of digital literacy competence assessed using authentic software applications. *Educational Technology Research and Development, 68,* 2991-3013. https://doi.org/10.1007/s11423-020-09825-x

Rush Hovde, M., & Renguette, C. C. (2017). Technological literacy: A framework for teaching technical communication software tools. *Technical Communication Quarterly, 26*(4), 395–411. https://doi.org/10.1080/10572252.2017.1385998

Tabachnick, B. G., & Fidell, L. S. (2019). *Using multivariate statistics* (7th ed.). Pearson Publications.

Vee, A. (2017). *Coding literacy: How computer programming is changing writing*. Cambridge, MA: MIT Press.

Yilmaz, R., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence, 4*, 100147. https://doi.org/10.1016/j.caeai.2023.100147