Contents

1.0 Introduction

This thesis investigates how the use of the Large Language Models (LLM) impact student learning performance and computational literacy. Participants were recruited from an introductory Python programming course to participate in the empirical study.. As a long time instructor of the course, previous observations revealed students use AI for both superficial task completion and to gain a deeper understanding of course material. To study the repercussions of this phenomena, student interactions with a course-provided LLM were captured, categorized then quantified to determine their correlation with academic outcomes. Given the disruptive potential of LLMs in software development and education, the findings will provide crucial insights into how to best leverage LLMs to foster positive learning outcomes and mitigate negative ones for novice programmers.

This study was very much a realization from years of teaching introductory Python programming to undergraduates. CLEAN THIS UP. . . .into which facets of Large Language model use influence student grades. Oftentimes students are outcomes-focused so any insights into LLM use that foster positive outcomes, yet diminish negative ones would likely serve as a motivational driver.

Having taught undergraduates Python programming to undergraduates for seven years, I've observed several trends (ELABORATE). I am constantly improving the course and its delivery to help students achieve those outcomes. When generative AI burst onto the scene it was a disruptor. Students now had a programming expert at their fingertips. It could serve as a 24/7 tutor, teacher, code debugger. It was an entity students could now use to have an dialog on-demand for their questions. I embraced it not only allowing use in my courses but encouraging it.

With all of the upside, the downside of course, it can be used to circumvent learning though not-so constructive activities such as generating answers to assignments, or having AI explain their own code as part of a code reflection.

My suspicion was that this type of superficial learning was going on in my courses. My observation was based on students with quality assignments, yet poor exam scores. There are a variety of reasons why student achievement on the numerous

practice activities in the course might not correlate with an exam score. And I've always had this problem to some degree, my sense was AI was exacerbating the problem.

While I lacked the evidence to support it, on rare occasions I would observe it. For example, students would come to office hours stuck on a problem. I'd ask them to walk through what they had so far, but they could not explain it. They informed me AI helped them with the problem but their understanding of the generated code was so lacking they could not adapt it to fit the problem.

This got me thinking. Perhaps I should observe students completing a programming task with AI and ask them to think-aloud while they do it. I invited a handful of students to participate and observed the results. Of the behaviors observed, two were the most interesting.

Some students engaged in scaffolding with the AI, using it to support and justify their own thinking as they solved the problem in code. These students were critical of the code that was generated, seeking to understand it and often asking the AI to explain or justify its decisions.

Another group of students used the AI for task completion; delegating to it and trusting the generations were correct. There was clear evidence of overreliance, with students believing the problem was solved correctly, when it was not. The students that were able to identify the execution output from the AI-generated was not correct, would often start over or make unreasonable requests for the AI to fix itself.

2.0 Literature Review

The following literature review presents the necessary background to better understand Large Language Models (LLMs) and their impacts on how novice programmers learn to code. We begin by exploring key research around LLMs and how they have impacted disciplines since their inception. Specifically, we provide evidence of LLMs as a disruptive innovation in the field of computer programming. LLMs have transformed not only how people code, but the ways in which programmers think about programming itself. This has implications for not only how code will be written in the future, but what skills are necessary for becoming a programmer and the methods by which those skills are taught to novices.

In addition, this literature review will explore the research of LLM use by teachers, students and administration within higher education. The research embodying the strengths, drawbacks, and unique challenges of LLMs within this context is discussed. This is necessary to differentiate between the overarching issues of LLMs in education and those specific to the topic of computer programming education.

Research will show LLMs are transforming how skilled professionals write code.

LLMs are being used predominantly by teachers and students in computer programming education as well, which has forced academics to rethink not only how to teach computer programming to novices, but what should be taught in the first place. The literature will identify this through the positive and negative impacts LLMs are having programming education.

The academic literature on computer science education discusses the well-known challenges of novice users learning to program. These challenges are the impacts of cognitive load on learning, getting learners to focus on computational literacy over syntax and technology, the role of self-efficacy and motivation on success, and the importance of metacognition for building problem-solving skills. This literature review will revisit each of these challenges through the lens of how LLMs are impacting them both positively and negatively.

2.1 Rise of the Large Language Model (LLM)

Natural Language Processing (NLP) has undergone significant changes in the past few years. With the introduction of transformer-based neural network architecture, Natural Language Processing took a giant step forward in performance and accuracy (Gillioz et al., 2020).

The Generative Pre-Trained Transformer (GPT) reasonably predicts the next word in a sequence using the input and previously generated output. While predicting the next token in the sequence is the extent of their ability (Shanahan, 2024), transformer-based language models trained on large data have demonstrated highly effective reasoning capabilities.

Open AI's foundational paper, "Language Models are Few-Shot Learners", demonstrated these transformer-based language models can produce human-level performance when they are trained on large copra (Brown et al., 2020). This was a pivotal discovery because at the time since prior to this paper transformer-based models were trained to be relatively task specific (Zhao et al., 2023).

The paper from Brown et al., 2020 led to significant advancements in research with respect to understanding the capabilities of LLMs. Wei et al. (2022) discovered reasoning capabilities of LLM's can be improved through a technique called chain-of-thought prompting. By including few-shot examples that break down complex reasoning into steps, the LLM can use those shots provided as an example of how to explain a complex process.

(Halevy et al., 2009) seminal paper, "The unreasonable effectiveness of data", explains as the training data set size increases, so does the model accuracy. In addition, the specific selected model algorithm becomes less relevant as training data set size increases. (Wei, Bosma, et al., 2022) documented a similar effect with large language models. Larger-sized models exhibited emergent abilities not found in their smaller counterparts. Examples of emergent abilities seen in the larger models include complex arithmetic and reading comprehension.

2.2 LLMs as a Disruptive Innovation for software development

(Christensen et al., 2018) divide technological innovations into two distinct types. Sustaining innovations improve existing products and services, while disruptive innovations provide a unique set of features to an initial set of customers. From the perspective of companies that offer generative AI such as Google, Anthropic, Open AI, and Microsoft, Horn considers generative AI to be a sustaining innovation (Horn, 2024). In their comprehensive literature review of AI as a disruptive innovation, (Păvăloaia & Necula, 2023) consider the application of Generative AI to be a disruptive innovation across different sectors such as healthcare, agriculture business and education.

The rise of AI-assistant programming tools in industry is evidence of this disruption. There are a growing set of tools available in the cloud: Github Copilot, Amazon CodeWhisperer, Gemini Code assist, Claude Code, Open AI Codex v2, Tabnine, and Codeium, in addition to self-hosted options like FauxPilot, Tabby, and CodeLLama. While each provides a unique set of features for differentiation, they all have primary functions like code completion, code generation, code explanations, and discussion. The primary value-add touted by these tools is developers will be able to write code in less time, improving productivity. Talk about "vibe coding" and Lovable, Bolt, Replit and Cursor.

IMPORTANT: The gap in the literature is tying the AI use to the grade! Point to studies showing how students use AI and that it may be helpful or harmful. Also find studies on superficial learning and impacts on grades (just learning in general).

Other learning strategies strategic like gaining a deeper understanding.

3.0 Methods

3.1 Introduction

The objective of this research is to study the impact of generative AI on an individual's learning. Specifically, this research studied the impacts of the Large-Language Model (LLM) used by students enrolled in an introductory Python programming course. Their learning was studied through the lenses of academic performance on a summative assessment at the mid-term of the course in addition to scores on a computational literacy instrument. Trace data from the use of LLMs was systematically categorized into activities that demonstrate superficial task completion versus those where the participant seeks to learn. These AI interactions were then quantified for each participant so that the final unit of analysis could investigate impact on exam scores. To further understand the complexities of AI use, a control and treatment group established as part of a randomized controlled study, to discover what influence, if any at all, context-awareness had on exam scores. This chapter explains the methodology that was used to address the research questions of this study.

3.1.1 Research Questions

For this study, the following research questions were established. These questions were addressed using a mixed-methods approach consisting of content analysis (Krippendorff, 2018) of participant interactions with LLMs and statistical analysis (Creswell & Creswell, 2017: 60).

RQ1: How does large-language model use influence student learning performance?

This question was addressed using a quantitative correlational design (Creswell & Creswell, 2017: ch8) where the content analysis from participant's LLM usage was the independent variable and their exam was the dependent variable. Hierarchical linear regression analysis (Raudenbush & Bryk, 2002) was used to explain the contribution of each of the independent variables in addition to their collective impact on exam scores.

RQ2: When the prompt is adjusted to include assignment instructions (in-context learning) what is the impact on student learning performance?

This question was addressed using a between-subjects randomized controlled trial (Creswell & Creswell, 2017: ch8). Linear regression analysis was used to evaluate the causal effect of in-context learning on student exam performance as a dummy variable and exam score as the dependent variable,. There were follow-up mediation and moderation analyses (Edwards & Lambert, 2007) to explore underlying mechanisms of LLM usage.

RQ3: What is the relationship between large language model use and computational literacy?

This question was addressed using quantitative correlational design with repeated measures (Creswell & Creswell, 2017: ch8). Participants' computational literacy was measured at separate intervals (beginning of the course and a midterm) using identical instruments. Similar to RQ1 linear regression analysis was used where the content analysis from participant's LLM usage was the independent variable and their exam was the dependent variable. ANCOVA was used to account for individual differences in prior knowledge. To provide a comprehensive understanding of LLM usage computational literacy relationships, a supplementary change score analysis examined whether usage patterns predicted improvement magnitude.

3.2 Study Design

A diagram of the study design
Figure1: An overview of the study design.

3.2.1 Overview

The study took place over a six-week period in the Spring 2025 semester of an introductory Python programming course, IST256 (https://ist256.com/spring2025/syllabus/). Taught within the School of Information Studies at Syracuse University, the course teaches programming fundamentals from the informatics perspective and is intended for non-computer science majors. There were 173 students enrolled in the course. The study only focused on the first 6 weeks because these units cover basic computational literacy constructs as applied to the Python programming language. These include instruction sequencing, variables, branching, iteration, and composition (functions). The course schedule can be found here: (https://ist256.com/spring2025/syllabus/#course-schedule). The study was exempt from IRB under section §46.104 section 1, which covers research conducted in an established educational setting. A university IRB Exemption application has been filed and obtained for category 1 for research in established or commonly accepted educational settings. The IRB# is 24-346. While all elements of the study design were part of the course, students could elect to opt-out of including their data in the study.

To better understand the complexities of AI use in learning, a variety of research methods were adopted. Measuring the impact of AI use on exam scores used a correlational design CITE. Computational literacy was measured using a pretest-posttest design to establish a baseline prior to the intervention. To measure the impacts of in-context learning a between-subjects design was employed with randomly-assigned control and treatment groups. Analysis of chat activity among participants was crucial for

The study begins with a diagnostic instrument C1 to get the baseline of computational literacy of each participant. This happened within the first week of class before any instruction. At this time students were introduced to the course-provided AI, (https://ai.ist256.com), which was a Large Language Model (GPT4o-mini) configured with a system prompt. Students were encouraged to use the AI as a virtual tutor asking it for help with Python questions and course-related assignments. When asking a specific question about an assignment, students were instructed to switch the LLM context by selecting the assignment in question from a drop-down menu.

Figure 2: Context-Selection from the IST256 AI Tutor

For the control group T1 this action did nothing - it does not add any additional context. For the treatment group T2 the action copied the assignment or lab instructions into the conversational context.

Figure 3: The treatment group (T2) is aware of the selected content.

Because the chatbot was self-hosted, all student interactions and AI responses were captured. D1 chatbot trace data is a dataset of those collected interactions throughout the six weeks of use. After the six week period, there were three

observations. E1 was the midterm exam in the course covering the content from the first six weeks, C2 was a re-issue of the same C1 diagnostic as a means to measure improvement of computational literacy. Observation Q1 was a questionnaire that asked for simple demographic data about each participant. Q1 was issued at the end of the course along with course evaluations.

3.2.2 Participant Funnel

Participants were students recruited from a Spring 2025 section of IST256. There were 173 students enrolled in the course. The research was exempt from IRB under section §46.104 section 1, which covers research conducted in an established educational setting. A university IRB Exemption application was filed and approved for category 1 for research in established or commonly accepted educational settings under the Syracuse University IRB number #24-346.

Of the 173 participants only 126 consented to the study. Five of the students who consented did not complete the observations necessary for the dependent variables: C1, C2 or E1, leaving 121 participants.

3.2.2.1 Population 1: Chatbot participants There are two populations under analysis. The first population is the set of participants who used the AI chatbot and therefore have the trace data D1 necessary to study their AI interactions. There were 48 individuals in the control group and 39 in the treatment group for a total participant population of 87.

Figure 4: The participant funnel for chatbot use. 87 participants.

3.2.2.1 Population 2: Chatbot participants with Survey Responses A survey Q1 was issued to students with the goal of identifying possible covariates. Besides demographic questions around the year of study, major, and gender students were also asked about their programming experience prior to the IST256 course. Ten chatbot users did not complete this survey thus reducing the participant size down to 77 whenever survey responses were needed in the analysis. Among the 77 participants, 41 were in the control group and 36 were in the treatment group.

Figure 5: The participant funnel when accounting for survey responses.

3.2.3 Computational Literacy Instrument (C1/C2)

No consensus models exist for developing computational thinking (Shute et al., 2017), therefore no commonly accepted instrument for measuring computational thinking exists. Brennan & Resnick, (2012) assert three key dimensions of the computational thinking framework: computational concepts, computational practices, and computational perspectives. Computational concepts are programming structures like loops and conditions, debugging and remixing are examples of computational practices, and activities such as expressing and framing problems

are examples of computational perspectives. The psychometrically validated CT-Test instrument developed by Román-González et al., (2017) measures computational concepts only. The computational concepts evaluated through the instrument such as loops, branching, arrays, functions, instruction sequencing aligned closely with the computational concepts taught in the first 6 weeks of content in the IST256 course.

The CT-Test consists of twenty-eight multiple-choice questions, and students are given 45 minutes to complete the test. The test questions were delivered online one question at a time in a random order using Syracuse University's Blackboard learning management system. Like the author's original instrument, students were permitted to skip questions and revisit them later. The second test C2 had the same questions as C1 only the order of the questions was re-randomized for each student. Students received a score out of 28 on the test but no indication of what questions they got correct / incorrect.

The following figure is an example question from the CT-Test. The entire instrument can be found in appendix A.

Figure 6: A sample question fro the CT-Test

3.2.4 Chatbot Design

The AI chatbot website used in the experiment (https://ai.ist256.com) was programmed in Python using the Streamlit framework (https://streamlit.io/). It was released as open source under the Apache 2.0 license and published to Github (https://github.com/cent-ischool/ist256-chatapp). The application was deployed to a Kubernetes cluster in the Syracuse University data center. Users were required to authenticate with their Syracuse University credentials to use the application. Their chat interaction trace data was stored in a Postgresql database in the same data center. A python script was written to extract the data from the database.

3.2.4.1 LLM Selection Acceptance criteria for choosing an LLM for the AI chatbot came down to accuracy and response time. I wanted the accuracy and response time to be on-par with the hosted platforms at that time: ChatGPT ((https://chatgpt.com/) and Claude (https://claude.ai/), so that participants would not churn based on those criteria.

Open-source models such Dolphin3, Llama3 and Qwen2.5 were considered initially. Unfortunately their response times fell off significantly at scale, likely due to these models being hosted on time-shared Syracuse-University GPU hardware. Their accuracy was found to be suitable for the content of the course, however.

Ultimately, the large language model selected Open AI's GPT4o-mini. The model, which strikes a balance between price, response time and performance, had an adequate understanding of introductory Python, and since it was hosted

in the Azure cloud it handled concurrent text generations. GPT4o was also evaluated, but since it performed similarly, at 16 times the price of 4o-mini, the latter seemed an obvious choice.

GPT4o-mini was a very adequate model. With correct prompting, it was able to generate correct solutions to all programming assignments and labs in the course. It also generated correct responses to 43 of 45 questions on the midterm exam (E1), placing it in the 98th percentile.

3.2.4.2 Random Assignment Once authenticated to the AI chatbot website, participants were assigned to the control or treatment groups based on the MD5 hash of their student ID being odd or even. This method assured each student was always assigned to the same group at each login session. The algorithm is provided here:

```
hash = md5(student_id)
number = int(hash)
in_treatment = mod(number, 2) # odd or even
```

3.2.4.3 Control Group (T1) Every request to the AI Tutor included a system prompt. The system prompt was used to place some guardrails on the AI and control the generated output. The system prompt's objectives were set to: (1) generate Python code in the style which is taught in the course, (2) explain all code generated written, (3) adopt a persona of a helpful and friendly tutor for a college level introductory Python course, and (4) not answer questions that are not course related.

Several system prompt iterations were evaluated before the following was determined to be most suitable for meeting the objectives while minimizing the number of input tokens.This was the base model configuration for the control group (T1).

Figure 7: The system prompt for the control group T1.

3.2.4.4 Treatment Group (T2) The treatment group consisted of the control group LLM and configurations (T1) plus a user prompt injected into the conversation based on contextual selection. For example, if the participant selected **03-HW-Conditionals** as the context (assignment), the content of the homework assignment notebook **lessons/03-Conditionals/HW-Conditionals.ipynb**, was loaded into the chat conversation. This was the same assignment to be completed by the student and contains the instructions, suggested approach, sections where code must be written and any sample code. The following prompt template is used to add the information to the conversation:

A black screen with white text AI-generated content may be incorrect.

Figure 8: Treatment T2 context prompt template.

Subsequently, the AI chatbot responds that it is ready to assist with the assignment. The AI response demonstrated context awareness.

A black background with white text AI-generated content may be incorrect.

Figure 9: AI response to context selection from T2.

When a participant asked the AI to complete a specific section of the assignment, the AI responded with working code and with an explanation of it. This is a key difference between T1 and T1. The control group T1 could generate the working code as well, but the participant must elicit the prompt to generate the results. This could be achieved by copying the assignment instructions into the prompt, which was all T2 was doing essentially.

The purpose of the control / treatment group was to help understand the impact of context-awareness on learning. For example, did the context-awareness improve user engagement with AI, or simply make question-answering more convenient?

3.2.5 Midterm Exam (E1)

The purpose of the midterm exam was to measure students' understanding of the fundamental concepts in the course. The exam covered:

1. Conceptual ideas, e.g. Definite and indefinite loops

2. Application of the concepts, e.g. when does one use a definite vs indefinite loop?

3. Concept in Python language, e.g. how does one use while or for to write loops? Is the loop you see a definite or indefinite loop?

4. Application in Python language, e.g. here is code with a while loop. What is the output upon execution?

The exam contained 45 multiple choice questions and students were given 45 minutes to complete it. The exam was closed book with the exception of a single page of notes as a study guide. Exam and academic integrity were preserved by issuing the exam in class and on paper. There were 54 versions of the same exam, with each version having both questions and answers randomized. Versions A-D were issued in class, version E was reserved for students at the testing center.

The Cronbach's Alpha (Cronbach, 1951) of the exam versions was between 0.83 and 0.88, indicating high reliability the instrument reflects students' subject knowledge.

Exam Version (E1)	N	Cronbach's Alpha
A	40	0.87
B	40	0.86
C	40	0.83

Exam Version (E1)	N	Cronbach's Alpha
D	39	0.87
E	9	0.88
	168	

Table 1: Cronbach's Alpha of E1 midterm exam

3.2.6 Questionnaire (Q1)

A debriefing questionnaire (Q1) was issued after the experiment. The purpose of this questionnaire was to:

1. Collect basic demographics from the participant pool. Such as gender, major, and class rank.

2. Identify participants with prior computer programming experience.

3. Identify participants who used AI other than the AI provided.

4. Collect student perceptions of programming and AI use.

The questionnaire was delivered using the Syracuse University Qualtrics survey system and can be found in Appendix B. Only demographic data and responses to prior programming experience were used in the data analysis.

3.2.7 Chatbot Trace Data (D1)

As explained earlier the AI chatbot recorded participant interactions and AI responses into a Postgresql database. The data included when they interacted with the AI, the context they selected, if they were in the control or treatment group. Here's a data dictionary of the fields in the trace data:

Field	**Data Type**	**Purpose**
Id	Sequential integer	A unique number for each item in the trace data, the highe
Session Id	Universally Unique ID (UUID)	A globally unique identifier to record the chat session. Filt
Participant ID	Text	The participant number. Unique for each participant, so th
Timestamp	ISO8601 timestamp	The UTC timestamp user's prompt or AI response as text
Model	Text	The AI Model used. This is always gpt-4o-mini
Rag	Boolean	True is the treatment group T1 whereas False is the contro
Context	Text	The user selected context at the time of the user prompt o
Role	Text	Values "user" or "assistant" Indicator of whether the row w
Content	Text	In the case of Role == "user", the Content is the prompt.

Table 2: Fields in the D1 Chatbot Trace Data

The trace data could be coded in a variety of ways, such as based on the type of question asked, or the specifically of the prompt. Ultimately this trace data was

used to classify users' interactions with the AI by session Id, then quantify those session-based interactions.

There were 9,518 entries in the D1 chatbot trace data. The granularity of the trace data was one row per user prompt or AI response. To analyze the data based on session - a series of requests and responses representative of a conversation - the data was grouped by session id, and sorted by timestamp. Each session was then assembled into a conversation thread consisting of the user prompt and AI responses. This resulted in 1024 sessions.

3.3 Data Analysis

3.3.1 Overview

The following section outlines the methods used to operationalize the D1 chatbot trace data. Because all three research questions necessitated use of the trace data for answering the research questions, this data played a pivotal role in the study. Categorical content analysis (Neuendorf, 2017) was used to classify student AI interactions into two categories of learning-supportive and learning-avoidance activities. The classification unit boundary was defined as each participants' chat session from login to logout. The trace data would be coded at the session level (n=1024) and the codes would be counted by participant for analysis at the participant level (n=87).

Before categorical content analysis could be performed, a codebook had to be built. I built the codebook deductively based on observations of other researchers in literature in addition to my seven years of experience teaching the course. I then coded a random sample of 50 sessions from the 1024 sessions in total using the codebook I constructed. Next I converted the codebook into an AI prompt so I could use an LLM to code the other sessions similar to work done by Xiao et al., (2023). Pilny et al., (2024) discovered that not all LLMs perform equally well at content analysis, and for this reason four different models were evaluated against my human-coded sample of 50 sessions. Coding consistency is also important and some LLMs have shown their outputs to be variable over time (Hackl et al., 2023). To mitigate this issue each candidate LLM would execute the same content analysis of the 50 sample sessions three times. Krippendorff's alpha reliability measure (Krippendorff, 2011) was used to validate the inter-coder reliability among the LLM when compared to my coding and with respect to itself.

Once the best model was determined, all 1024 sessions were classified and then counted in buckets of "Learning" or "Task Completion" at the participant to match the unit of analysis for addressing the research questions.

3.3.2 Tools

Data analysis was performed using the Python programming language, using Jupyter notebooks. Data visualizations were created with the help of the

matplotlib, seaborn, sankeyflow and forestplot libraries. Statistical analysis was accomplished with the help of the pandas, scipy, statsmodels, pingouin, and simpledorff libraries. [QUESTION: Do I cite these]

Openrouter (https://openrouter.ai) was used to provide unified API level access to different Large-Language Models across platforms like OpenAI, xAI, Google and Anthropic.

To protect participants, the source data and any processed data sets were stored with the code in a private Git repository. This repository was backed up to a personal storage device. Final code and data for this research will be accessible on a private Github repository at https://github.com/mafudge/dps-thesis until it can be determined the data may be shared publicly. The final data will be scrubbed of all personally-identifiable information which can link the data back to an individual.

3.3.3 Operationalizing D1 Chatbot Trace Data

To perform the categorical content analysis of the 1024 D1 chatbot trace data sessions, first a codebook was built deductively. An initial set of codes were established based on observations of other researchers in literature in combination with my seven years of experience teaching IST256. A sub-sample of 50 sessions were selected at random from the D1 dataset to identify initial classifications. I classified each session according to the codebook in Table 3. In addition, I summarized the classified activity as an example which could be incorporated into the AI prompt used by the LLM.

Participant Behavior	**Definition**	**Example**	**Classified Activity**	• *Supported Citations**
Seeking Answers (Superficial Learning)	The participant makes a direct ask of the AI to complete their assignment or lab. There is no conversation.	"Can you code the solution for assignment ABC?" "How do I complete section XYZ in the lab?" [(Hassan et al., 2025)] (https://www.zotero.org/google-docs/?ocgJMx)	Task Completion	[(Finnie-Ansley et al., 2022)] (https://www.zotero.org/google-docs/?EeccHE) [(Hassan et al., 2025)] (https://www.zotero.org/google-docs/?M8nObm)

Participant Behavior	**Definition**	**Example**	**Classified Activity**	• *Supported Citations**
Overreliance (Cognitive offloading for task completion)	The participant asks the AI to complete a task and does not further engage. Thus there is no clear intention to understand their work for them. Thus a deep connection to the material is not formed.	"Can you convert this algorithm into code?" "Can you provide an approach to start this assignment?" "Fix this error for me." [(Prather, Reeves, Leinonen, et al., 2024)] (https://www.zotero.org/google-docs/?fWvkuI)	Task Completion	[(Prather, Reeves, Leinonen, et al., 2024)] (https://www.zotero.org/google-docs/?b94RCp) [(Margulieux et al., 2024)] (https://www.zotero.org/google-docs/?afJ6YN)
Scaffolding (AI as Expert)	AI provides support / expertise to the learner to a learner to help them achieve a task they couldn't complete on their own.	"What does this error mean?" "Can you help me locate the error in my code?" "Can you help me troubleshoot my code?" .[(Prather, Reeves, Denny, et al., 2024)] (https://www.zotero.org/google-docs/?CjRe5C)	Learning	[(Prather et al., 2019)] (https://www.zotero.org/google-docs/?wDe1Ck) [(Prather, Reeves, Denny, et al., 2024)] (https://www.zotero.org/google-docs/?svZdyf) [(Hassan et al., 2025)] (https://www.zotero.org/google-docs/?vorDax)

P articipant Behavior	**D efinition**	**Example**	**Classified Activity**	• *Supported Citations**
Questioning (AI as Tutor)	Asking the AI to explain a concept or provide code examples of a concept.	Can you provide an example of a nested loop? Why would I use a nested IF verses elif? Can you help me understand di ctionaries? When would I use a list versus a dictionary?	Learning	[(Fi nnie-Ansley et al., 2022)] (https://ww w.zotero.or g/google-do cs/?nTmux6) [(Cambaz & Zhang, 2024)] (https://ww w.zotero.or g/google-do cs/?NzDiBi)
Unable to Determine	The transcript does not provide enough information to classify the par ticipant's behavior. Off topic question not about the subject matter.	"When is the exam?" "Can you tell me the weather? " What did I ask you last week?	I nconclusive	

Table 3: Codebook of participant behaviors

For example, in one of *Participant 119*'s sessions this was this only question asked the AI chatbot:

> PARTICIPANT_119: "Write a function, score_sentiment() which given 3 inputs: positive words, negative words and some text, will return an integer score of sentiment as output. Assume all inputs are lower case and do not have any punctuations."
> AI_ASSISTANT: (writes code and explains what it does as the bot

was designed)

The participant pasted in part of the assignment instructions asking the AI generate code. I classified this session as Task Completion, under the behavior Seeking Answers.

For another example, here is one of *Participant 107's* sessions. I classified this interaction as Learning under the behavior Questioning, as it is obvious the participant is seeking to understand Python concepts.

> PARTICIPANT_107: Why false?
> a=0.1
> b=0.2
> bool(a+b==0.3)
> AI_ASSISTANT: (provides an explanation with examples)
>
> PARTICIPANT_107: explain 7%2
> AI_ASSISTANT: (provides an explanation with examples)
>
> PARTICIPANT_107: explain == vs =
> AI_ASSISTANT: (provides an explanation with examples)

An AI prompt was formulated from the codebook and the various examples of interactions I observed within the observations. The AI prompt is included in Appendix C.

3.3.4 Model Selection Reliability Testing

Using a Large-Language Model for content analysis raised two concerns. First was inter-coder reliability. I wanted the LLM classifier to match my 50 classifications as closely as possible. Secondly was internal reliability, LLMs are prone to hallucinations so I had to ensure whatever model I selected was as consistent as possible across multiple classifications of the same data. The best model would then be the one which was the most similar to my classification and the most internally consistent.

Four different models were chosen as candidates: x-ai/grok-4-fast, openai/gpt-5-mini, google/gemini-2.5-flash, and anthropic/claude-sonnet-4. Each model candidate came from a different AI provider and were considered appropriate for text-processing type tasks at the time of the study.

The same AI prompt from Appendix C was used across all four models, however the prompt in appendix C was not the initial prompt. It was refined to ensure accurate output representations across the models. To accomplish this task, my interpretation of the session was compared against the outputs of the 4 models, and the prompt was refined until there was consensus. This was repeated for each type of classification: Task Completion / Seeking Answer, Task Completion / Overreliance, Learning / Scaffolding, and Learning / Questioning. Four examples in total.

With an established AI prompt, I could now check for inter-coder reliability. Each of the four models was used to classify the same 50 sessions in the subsample I classified initially. My classifications were compared against the LLM's classifications using the Krippendorff's Alpha to obtain a measure of agreement for inter-coder reliability.

The inter-coder reliability procedure was repeated three times to establish a measure of consistency among model runs. Once again the Krippendorff's alpha measure was used to measure the model's agreement, but this time with itself across three runs. The expectation was an internal consistency of 1.0 which means the LLM performed the task identically each time.

Model	**Agreement with Human**	**Internal Consistency (3 runs)**
x-ai/grok-4-fast	0.873131	0.885031
openai/gpt-5-mini	0.788552	0.914743
google/gemini-2.5-flash	0.833052	1.000000
anthropic/claude-sonnet-4	0.873131	1.000000

Table 4: The Krippendorff's Alpha of model vs human and internal consistency of model against itself.

As table 4 indicates, x-ai/grok-4-fast and anthropic/claude-sonnet-4 classifications were in closest agreement to my classifications, having the highest inter-coder reliability. Only google/gemini-2.5-flash, and anthropic/claude-sonnet-4 classified the data with perfect consistency across the three runs. For these reasons anthropic/claude-sonnet-4 was chosen as the model to complete the classification task performing highest in both categories.

Incidentally, the Krippendorff's alpha across all models + human for the three runs was 0.76108, 0.75115 and 0.77207.

3.3.5 Categorical Content Analysis

With the identification of the best LLM for the task, I now used anthropic/claude-sonnet-4 to classify all 1024 sessions in the D1 chatbot trace dataset. In total there were 623 labeled "Learning", 336 categorized as "Task Completion" and 35 categorized as "Inconclusive." The final unit of analysis is based on participants, so counts of classifications were grouped across the 87 participants. For example, participant #82 had 33 sessions categorized as "Learning", 7 sessions categorized as "Task Completion" and 2 categorized as "Inconclusive." Participant #124 has 5 sessions categorized as "Learning". 25 categorized as "Task Completion" and one categorized as "Inconclusive." With the data setup this way I could now use

the categorized counts as independent variables, and study their impacts on E1, C1 and C2.

Figure 10: Three classifications of chat sessions from the D1 dataset

3.4 Hypothesis Formulation and Methodology

With the dataset established the analysis could commence. This section breaks down the original research questions into testable hypotheses and then explains the methodology used to answer the research questions.

3.4.1 RQ1 Hypothesis and Methodology

Research Question 1 states:

RQ1: How does large-language model use influence student learning performance?

This question investigated the relationship between the quantity of participant sessions classified as “learning” and “task completion” from the D1 dataset and their midterm exam scores E1.

Alternate Hypothesis for H1

The quantity and classification of participant sessions in the D1 dataset are significantly correlated with their midterm exam scores (E1). Specifically:

1. There is a positive correlation between the quantity of sessions > classified as “learning” and midterm exam scores.
2. There is a negative correlation between the quantity of sessions > classified as "task completion" and midterm exam scores.

Null Hypothesis for H1

There is no statistically significant correlation between the quantity of participant sessions from the D1 dataset, whether classified as "learning" or "task completion," and their midterm exam scores (E1).

Methodology

A hierarchical multiple regression analysis (Raudenbush & Bryk, 2002) was employed to test the study hypotheses. This helped explain the contribution of each of the independent variables in addition to their collective impact on exam scores. The beta coefficients from the regression indicated the nature of the relationship such as positive or negative and the degree of influence of the independent variables. This strategy allows examination of each predictor’s independent contribution before assessing their combined effects. This would help to reveal whether both variables provide unique information about the outcome. There are three models:

Model 1 (Task completion only): E1 was regressed on task completion session count alone to establish the bivariate relationship.

Model 2 (Learning only): E1 was regressed on learning session count alone to establish its bivariate relationship.

Model 3 (Full model): E1 was regressed simultaneously on both task completion session count and learning session count to examine their independent effects when controlling for one another.

For each model, I reported regression coefficients (beta), standard errors, t-statistics, p-values, 95% confidence intervals, and overall model fit statistics (r-squared, adjusted r-squared, and F-statistic). Changes in r-squared across models were calculated to assess whether both predictors contributed unique variance beyond what either explained alone.

Methodological analysis with respect to H1:

1. The "learning" beta coefficient being positive with a p-value $< > 0.05$, would indicate statistical significance. This implies there $>$ would be a positive correlation between the quantity of sessions $>$ classified as “learning” and midterm exam scores (E1).
2. The “task completion” beta coefficient being negative with a p-value $> <$ 0.05 would imply a negative correlation between the quantity of $>$ sessions classified as "task completion" and midterm exam scores $>$ (E1).

Both conditions 1 and 2 must be satisfied to reject the null hypothesis for H1.

3.4.2 RQ2 Hypothesis and Methodology

Research Question 1 states:

RQ2: When the prompt is adjusted to include assignment instructions (in-context learning) what is the impact on student learning performance?

This question investigated the impact of in-context learning on midterm exam scores E1. The assignment instructions were added to LLM’s context of participants in the treatment group. This meant the LLM was aware of assignment-related questions.

Alternate Hypothesis for H2

The mean student learning performance score (E1) for the treatment group, which used the context-aware AI, will be statistically significantly higher than the mean score for the control group.

Null Hypothesis for H2

There is no statistically significant difference in the mean student learning performance scores (E1) between the treatment group and the control group.

Methodology

A between-subjects experimental design (Creswell & Creswell, 2017: ch8) was employed with participants assigned at random to either a control group (T1) or

treatment group (T2), which employed using context-aware AI. The dependent variable was midterm exam score (E1), measured on a continuous scale. The independent variable—group assignment—was dummy coded (0 = control, 1 = treatment) for regression analysis.

Linear regression was selected as the primary analytical approach, with the beta coefficient representing the mean difference in exam scores between control and treatment. Statistical significance was assessed using p-values <= .05.

Mediation and moderation analyses (Edwards & Lambert, 2007) were also conducted to understand the underlying mechanisms that session counts of task completion and learning might have on the treatment.

Mediation analysis was used to explore whether the treatment effect operated through changes in student usage behaviors:

Model 1 (Treatment → Learning Session Count) Does control/treatment group assignment predict learning session count?

Model 2 (Treatment → Task Completion Session Count) Does control/treatment group assignment predict task completion session count?

Moderation analysis examined whether the treatment condition altered the strength of relationships between usage patterns and exam performance.

Model 3 (Treatment x Learning Session Count) This model tested whether context-aware AI moderated the relationship between learning session engagement and exam performance.

Model 4 (Treatment x Task Completion Session Count) This model tested whether context-aware AI moderated the relationship between task completion session engagement and exam performance.

3.4.3 RQ3 Hypothesis and Methodology

Research Question 3 states:

RQ3: What is the relationship between large language model use and computational literacy?

This question investigated the relationship between the quantity of participant sessions classified as "learning" and "task completion" from the D1 dataset and their computational literacy scores (C1, C2).

Alternate Hypothesis for H3

The quantity and classification of participant sessions in the D1 dataset are significantly correlated with their computational literacy scores (C1, C2). Specifically:

1. There is a positive correlation between the quantity of sessions > classified as "learning" and computational literacy scores.

2. There is a negative correlation between the quantity of sessions > classified as "task completion" and computational literacy > scores.

Null Hypothesis for H3

There is no statistically significant correlation between the quantity of participant sessions from the D1 dataset, whether classified as "learning" or "task completion," and their computational literacy scores (C1, C2).

Methodology

To measure test-retest reliability a Pearson correlation will be measured against C1 and C2 (Vaz et al., 2013). To account for baseline computational literacy and test-retest effects from repeated administration of the same assessment from C1 to C2, I employed an analysis of covariance (ANCOVA). Specifically, I used linear regression with midterm computational literacy scores (C2) as the dependent variable with the baseline scores (C1), task completion session count, and learning session count as the independent variables (Huitema, 2011). This approach controls for individual differences in prior knowledge as measured on C1 while examining whether usage patterns predict final outcomes beyond what would be expected from baseline ability alone (Maxwell et al., 2017).

I also conducted a supplementary analysis using the change in diagnostic scores (C2 - C1) as the dependent variable to examine whether LLM usage patterns predicted the magnitude of improvement. This was a complementary approach as change scores low reliability when pre- and post-test correlations are high (Cronbach & Furby, 1970). Both analyses together offered a comprehensive understanding of the relationship between LLM usage and computational literacy development.

Methodological analysis with respect to H1:

1. The "learning" beta coefficient being positive with a p-value < > 0.05, would indicate statistical significance. This implies there > would be a positive correlation between the quantity of sessions > classified as “learning” and computational literacy scores (C2).
2. The “task completion” beta coefficient being negative with a p-value > < 0.05 would imply a negative correlation between the quantity of > sessions classified as "task completion" and computational > literacy scores (C2).

Both conditions 1 and 2 must be satisfied to reject the null hypothesis for H3.

3.4.4 Satisfying Assumptions of Linear Regressions

For every linear regression, tests for linear assumptions were applied to ensure the reliability of the results (Cohen et al., 2013; Osborne & Waters, 2002) . Residual plots and regression plots were observed for linearity. To determine if the variance of the residuals were constant across all levels of the independent variables, the Breusch-Pagan test for homoscedasticity was used (Breusch & Pagan, 1979). The

Durbin-Watson statistic was calculated to check for autocorrelation (Durbin & Watson, 1971). To verify the residuals were normally distributed, Shapiro-Wilk test was performed (Shaphiro & Wilk, 1965). And finally, to check for conditions of multicollinearity (Farrar & Glauber, 1967) among independent variables, the variance inflation factor metric was calculated .

Diagnostic testing that revealed heteroscedasticity, or non-constant error variance across the range of fitted values, would be regressed with robust standard errors. This ensured valid statistical inference in the presence of heteroscedasticity (MacKinnon & White, 1985). This approach provides robust hypothesis tests without requiring the homoscedasticity assumption, yielding accurate p-values and confidence intervals even when error variance is non-constant.

4.0 Results

4.1 Introduction

The findings of my research will be discussed in this section. [TODO: Write up a summary of findings for this section.]

4.2 Findings for RQ1

4.2.1 Model Overview for RQ1

This research question investigated whether the quantity and classification of student sessions with large language models predicted midterm exam performance (E1). Specifically, we examined whether learning session count and task completion sessions count were differentially associated with academic outcomes ($N = 87$).

The regression model used to test the hypothesis had learning session count and task completion count as independent variables, with midterm exam E1 as the dependent variable.

E1 ~ Learning Session Count + Task Completion Session Count

Alternate Hypothesis for H1

The quantity and classification of participant sessions in the D1 dataset are significantly correlated with their midterm exam scores (E1). Specifically:

1. There is a positive correlation between the quantity of sessions > classified as “learning” and midterm exam scores.
2. There is a negative correlation between the quantity of sessions > classified as "task completion" and midterm exam scores.

Null Hypothesis for H1

There is no statistically significant correlation between the quantity of participant sessions from the D1 dataset, whether classified as "learning" or "task completion," and their midterm exam scores (E1).

4.2.2 Regression Assumptions Tests of the Model for RQ1

A comprehensive assessment of ordinary least squares (OLS) regression assumptions was conducted prior to interpreting the findings of the model for RQ1. These checks assured the validity and reliability of the model estimates.

To check for linearity, partial regression plots were examined to assess the linear relationship between the dependent variable (E1) and each predictor variable while controlling for other predictors in the model. Visual inspection of these plots indicated approximate linear relationships, supporting the use of linear regression for analysis. Both independent variables had fewer observations as the counts of the participants' sessions increased. This was expected for session-oriented data which is commonly right-skewed, where most participants have few sessions and few participants have several sessions..
Figure ??: Partial regression plots when holding the other constant of E1 ~ Task Completion Session Count + Learning Session Count.

The plot of residuals versus fitted values showed symmetrical, randomly scattered points with no obvious pattern. The Locally Estimated Scatterplot Smoothing (LOESS) curve was relatively flat and close to the horizontal. The end of the LOESS curve does indicate a slight uptick likely due to lack of observations in that region.

Figure ??: Residuals vs Fitted Values for E1 ~ Learning Session Count + Task Completion Session Count

The Durbin-Watson statistic (DW = 2.09) was within the acceptable range, falling close to the ideal value of 2.0. This indicated no substantial autocorrelation among residuals, thereby satisfying assumption of independence.

The Breusch-Pagan test for heteroscedasticity yielded a non-significant result (chi-square = 0.29, p-value = .867). I interpreted this as the assumption of constant error variance across levels of the predictor variables was met. This was further supported by visual inspection of the residuals versus fitted values plot, which showed no clearly discernible pattern or funnel shape.

The Shapiro-Wilk test indicated that residuals were normally distributed (W = 0.99, p-value = .679). Since p>=0.05 we fail to reject the null hypothesis of normality. This was corroborated visually via a histogram of residuals and the Q-Q plot, both of which demonstrated close adherence to the normal distribution.

Figure ??: Evidence of normality among the residuals of E1 ~ Task Completion Session Count + Learning Session Count.

To check for multicollinearity, the variance inflation factor (VIF) test was performed. Values for both predictor variables were well below conventional thresholds of concern (VIF = 1.19 for both task completion session count and learning session count), indicating no problematic multicollinearity.

Given that all diagnostic tests indicated satisfaction of OLS regression assumptions, the model was deemed appropriate for interpretation. The model explained 18.9% of the variance in E1 (r-squared = .189, adjusted r-squared = .169), and was statistically significant overall (F(2, 84) = 9.76, p-value < .001).

4.2.3 Hierarchical Regression Analysis for the RQ1 Model

Prior to testing the full regression model, bivariate relationships between predictor variables and midterm exam scores (E1) were examined through simple linear regression analyses. This preliminary step allows for assessment of individual predictor effects before examining their combined contribution as a means to gain a better understanding of each independent variables' contribution to the model.

4.2.3.1 Model 1: E1 ~ Task Complection Session Count A simple linear regression revealed that task completion session count was a significant negative predictor of E1 ($F(1, 85) = 12.35$, $p < .001$). The model explained 12.7% of variance in exam scores ($R^2 = .127$, adjusted $R^2 = .117$), with a medium effect size (Cohen's $f^2 = 0.15$). For each additional task completion session, E1 decreased by 0.41 units (b = -0.41, SE = 0.12, t = -3.52, $p = .001$, 95% CI [-0.65, -0.18]). This bivariate relationship confirmed that greater engagement with task completion sessions was independently associated with lower exam performance.

4.2.3.1 Model 2: E1 ~ Learning Session Count In contrast, a simple linear regression with learning session count as the sole predictor yielded a non-significant model ($F(1, 85) = 0.62$, $p = .435$). Learning sessions explained less than 1% of variance in E1 ($R^2 = .007$, adjusted R^2 = -.004). The regression coefficient, while positive in direction, was not statistically significant ($b = 0.06$, SE = 0.08, $t = 0.79$, $p = .435$, 95% CI [-0.10, 0.23]). Considered in isolation, learning session count did not predict exam performance.

4.3 Findings for RQ2

Djhsfgkjhsdagfk

4.4 Findings for RQ3

dfhsgkljsad

4.5 Overall Summary of Findings

Hsdgakfjhg

5.0 Summary

This page is intentionally blank.

NOTE: Important not to draw too much from the findings This study took place over 1 semester and the method was figured out along with the analysis (leading the witness??) due to an accelerated timeline. . Further study is required as the accelerated timeline between understanding the chat interactions and performing the analysis. Plans to continue this study in future semesters to strengthen these findings further.

Also consider this thesis very must an exploration. Trying to understand what among AI use influences grades.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., & Henighan, T. (2020). *Language Models are Few-Shot Learners*.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*.

Breusch, T. S., & Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica: Journal of the Econometric Society*, 1287–1294.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., & Henighan, T. (2020). *Language Models are Few-Shot Learners*.

Cambaz, D., & Zhang, X. (2024). Use of AI-driven Code Generation Models in Teaching and Learning Programming: A Systematic Literature Review. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 172–178. https://doi.org/10.1145/3626252.3630958

Christensen, C. M., McDonald, R., Altman, E. J., & Palmer, J. E. (2018). Disruptive Innovation: An Intellectual History and Directions for Future Research. *Journal of Management Studies*, *55*(7), 1043–1078. https://doi.org/10.1111/joms.12349

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2013). *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge.

Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*(3), 297–334.

Cronbach, L. J., & Furby, L. (1970). How we should measure" change": Or should we? *Psychological Bulletin*, *74*(1), 68.

Durbin, J., & Watson, G. (1971). Testing for serial correlation in least squares regression. III. *Biometrika*, *58*(1), 1–19.

Edwards, J. R., & Lambert, L. S. (2007). Methods for integrating moderation and mediation: A general analytical framework using moderated path analysis. *Psychological Methods*, *12*(1), 1.

Farrar, D. E., & Glauber, R. R. (1967). Multicollinearity in regression analysis: The problem revisited. *The Review of Economic and Statistics*, 92–107.

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). The robots are coming: Exploring the implications of openai codex on introductory programming. *Proceedings of the 24th Australasian Computing Education Conference*, 10–19.

Gillioz, A., Casas, J., Mugellini, E., & Khaled, O. A. (2020). *Overview of the Transformer-based Models for NLP Tasks*. 179–183. https://doi.org/10.15439/2020F20

Hackl, V., Müller, A. E., Granitzer, M., & Sailer, M. (2023). Is GPT-4 a reliable rater? Evaluating consistency in GPT-4's text ratings. *Frontiers in Education*, *8*, 1272229.

Halevy, A., Norvig, P., & Pereira, F. (2009). The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, *24*(2), 8–12. https://doi.org/10.1109/MIS.2009.36

Hassan, M., Chen, Y., Denny, P., & Zilles, C. (2025). On Teaching Novices Computational Thinking by Utilizing Large Language Models Within Assessments. *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, 471–477. https://doi.org/10.1145/3641554.3701906

Horn, M. B. (2024, June 3). *What does Disruptive Innovation Theory have to say about AI? - Christensen Institute*. https://www.christenseninstitute.org/blog/what-does-disruptive-innovation-say-about-ai/

Huitema, B. (2011). *The analysis of covariance and alternatives: Statistical methods for experiments, quasi-experiments, and single-case studies*. John Wiley & Sons.

Krippendorff, K. (2011). *Computing Krippendorff's alpha-reliability*.

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage publications.

MacKinnon, J. G., & White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, *29*(3), 305–325.

Margulieux, L. E., Prather, J., Reeves, B. N., Becker, B. A., Cetin Uzun, G., Loksa, D., Leinonen, J., & Denny, P. (2024). Self-Regulation, Self-Efficacy, and Fear of Failure Interactions with How Novices Use LLMs to Solve Programming Problems. *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, 276–282. https://doi.org/10.1145/3649217.3653621

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2017). *Designing experiments and analyzing data: A model comparison perspective*. Routledge.

Neuendorf, K. A. (2017). *The content analysis guidebook*. sage.

Osborne, J. W., & Waters, E. (2002). Four assumptions of multiple regression that researchers should always test. *Practical Assessment, Research, and Evaluation*, *8*(1).

Păvăloaia, V.-D., & Necula, S.-C. (2023). Artificial Intelligence as a Disruptive Technology—A Systematic Literature Review. *Electronics*, *12*(5), 1102. https://doi.org/10.3390/electronics12051102

Pilny, A., McAninch, K., Slone, A., & Moore, K. (2024). From manual to machine: Assessing the efficacy of large language models in content analysis. *Communication Research Reports*, *41*(2), 61–70.

Prather, J., Pettit, R., Becker, B. A., Denny, P., Loksa, D., Peters, A., Albrecht, Z., & Masci, K. (2019). First Things First: Providing Metacognitive Scaffolding for Interpreting Problem Prompts. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 531–537. https://doi.org/10.1145/3287324.3287374

Prather, J., Reeves, B., Leinonen, J., MacNeil, S., Randrianasolo, A. S., Becker, B., Kimmel, B., Wright, J., & Briggs, B. (2024). *The Widening Gap: The Benefits and Harms of Generative AI for Novice Programmers* (No. arXiv:2405.17739). arXiv. http://arxiv.org/abs/2405.17739

Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2024). "It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers. *ACM Transactions on Computer-Human Interaction*, *31*(1), 1–31. https://doi.org/10.1145/3617367

Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods* (Vol. 1). sage.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Shanahan, M. (2024). Talking about Large Language Models. *Communications of the ACM*, *67*(2), 68–79. https://doi.org/10.1145/3624724

Shaphiro, S., & Wilk, M. (1965). An analysis of variance test for normality. *Biometrika*, *52*(3), 591–611.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Vaz, S., Falkmer, T., Passmore, A. E., Parsons, R., & Andreou, P. (2013). The case for using the repeatability coefficient when calculating test–retest reliability. *PloS One*, *8*(9), e73990.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2022). *Finetuned Language Models Are Zero-Shot Learners* (No. arXiv:2109.01652). arXiv. http://arxiv.org/abs/2109.01652

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.*

Xiao, Z., Yuan, X., Liao, Q. V., Abdelghani, R., & Oudeyer, P.-Y. (2023). *Supporting Qualitative Analysis with Large Language Models: Combining Codebook with GPT-3 for Deductive Coding.* 75–78. https://doi.org/10.1145/3581754.3584136

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., … Wen, J.-R. (2023). *A Survey of Large Language Models* (No. arXiv:2303.18223). arXiv. http://arxiv.org/abs/2303.18223
stensen, C. M., McDonald, R., Altman, E. J., & Palmer, J. E. (2018). Disruptive Innovation: An Intellectual History and Directions for Future Research. *Journal of Management Studies*, *55*(7), 1043–1078. https://doi.org/10.1111/joms.12349