

# Programming education and learner motivation in the age of generative AI: student and educator perspectives

Age of  
generative AI

91

Samuel Boguslawski, Rowan Deer and Mark G. Dawson  
*CODE University of Applied Sciences, Berlin, Germany*

Received 31 October 2023  
Revised 27 February 2024  
27 May 2024  
Accepted 28 May 2024

## Abstract

**Purpose** – Programming education is being rapidly transformed by generative AI tools and educators must determine how best to support students in this context. This study aims to explore the experiences of programming educators and students to inform future education provision.

**Design/methodology/approach** – Twelve students and six members of faculty in a small technology-focused university were interviewed. Thematic analysis of the interview data was combined with data collected from a survey of 44 students at the same university. Self-determination theory was applied as an analytical framework.

**Findings** – Three themes were identified – bespoke learning, affect and support – that significantly impact motivation and learning outcomes in programming education. It was also found that students are already making extensive use of large language models (LLMs). LLMs can significantly improve learner autonomy and sense of competence by improving the options for bespoke learning; fostering emotions that are conducive to engendering and maintaining motivation; and inhibiting the negative affective states that discourage learning. However, current LLMs cannot adequately provide or replace social support, which is still a key factor in learner motivation.

**Research limitations/implications** – Integrating the use of LLMs into curricula can improve learning motivation and outcomes. It can also free educators from certain tasks, leaving them with more time and capacity to focus their attention on developing social learning opportunities to further enhance learner motivation.

**Originality/value** – To the best of the authors' knowledge, this is the first attempt to explore the relationship between motivation and LLM use in programming education.

**Keywords** Programming education, Large language models, LLMs, Generative AI, ChatGPT, Copilot, Introductory programming, Learning science, Self-determination theory, Motivation

**Paper type** Research paper

## 1. Introduction

The latest development of the digital revolution is the advent of generative AI, or large language models (LLMs). Just as the computer revolution of the 1960s transformed countless



© Samuel Boguslawski, Rowan Deer and Mark G. Dawson. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

This research was funded by the German Ministry for Education and Research (BMBF).

Information and Learning  
Sciences  
Vol. 126 No. 1/2, 2025  
pp. 91-109  
Emerald Publishing Limited  
2398-5348  
DOI [10.1108/ILS-10-2023-0163](https://doi.org/10.1108/ILS-10-2023-0163)

sectors of society – including manufacturing, communication and education – ongoing digitization and generative AI are set to radically transform how we live, work and learn. While industry actors must quickly adopt new technologies to stay ahead, educational institutions can be slower to integrate advances into their teaching (Gilch *et al.*, 2019), meaning that students may leave university unable to meet industry expectations (Craig *et al.*, 2018). Recognizing this, the German Government has invested in various efforts to digitize education (BMBF, 2021), including funding the development of an online learning platform for programming. As part of this latter project, researchers at CODE University of Applied Sciences investigated the experiences of programming students and educators to inform the platform design.

What emerged was a general picture of contemporary programming students' motivations, emotions, needs and preferences – all of which significantly impact learning. Initially, the researchers did not aim to explore LLM use, but it quickly became clear that this has already become an integral resource for many learners. While educators are still debating the pros, cons and impacts of LLMs on programming education (Lau and Guo, 2023), they have already impacted programming students' learning strategies and motivation, with significant implications for education design going forward.

## 2. Literature review

As with all learning, motivation is key to success in programming education. In the following three subsections, we review the literature around learner motivation and self-determination theory (SDT), before moving on to discuss how this relates to programming education. Finally, we discuss the early research on LLM use in programming education.

### 2.1 Learner motivation and self-determination theory

Intrinsic and extrinsic motivation both significantly impact learners' engagement and performance (Jones, 2013). Intrinsic motivation refers to the internal enjoyment and satisfaction of engaging in an activity, whereas extrinsic motivation refers to external factors that drive individuals to do so, often involving receiving rewards or avoiding punishments (Schwartz and Wrzesniewski, 2019). Intrinsic motivation is recognized as key to effective learning (Renniger and Hidi, 2019). In the context of programming education, this involves curiosity about coding and the satisfaction derived from completing programming tasks. Extrinsic motivation is provided by external factors such as deadlines, assessments or the promise of career opportunities. It is generally accepted that intrinsic motivation leads to better engagement and performance (Vansteenkiste *et al.*, 2004), whereas extrinsic motivation can inhibit intrinsic motivation in certain circumstances (Deci and Ryan, 2000; Murayama, 2019). However, because intrinsic and extrinsic motivations coexist and can interact in complex ways (Cerasoli *et al.*, 2014), it is important to take both into account. Holistically, motivation can be fostered by linking external motivators to more intrinsic drives – for example, by framing learning in terms of learners' future goals (Vansteenkiste *et al.*, 2004) and underlining the usefulness of computational thinking skills – while tangible records of achievement foster self-perceptions of capability (Eckerdal *et al.*, 2005; Demertzi *et al.*, 2018).

According to SDT, motivation is significantly influenced by the basic human need to experience autonomy, competence and relatedness (Deci and Ryan, 2000). Autonomy is a sense of agency, control and independence; competence involves feeling capable, confident and effective; and relatedness involves feeling connected to others and social structures. A learner might increase their sense of autonomy by choosing when, how and what they learn; believe they have improved in competence as they successfully complete activities; and feel

connected to other learners via social affiliations in class or online (Chen *et al.*, 2010). A decrease in any of these factors is likely to negatively impact learner motivation, engagement and achievement (Chiu, 2023).

Autonomy has been identified as the most salient factor in student motivation, with students who reported higher degrees of perceived autonomy also reporting higher motivation, whereas motivation can decrease when elements of learning are made mandatory (Chen *et al.*, 2010; Xie *et al.*, 2006). However, learners vary in the degree to which they display autonomous learning behaviors, so increasing or decreasing autonomy is also likely to have different motivational outcomes for individual learners (Rienties *et al.*, 2012). It is also important to note that the benefits of increased autonomy may be counteracted by the increased challenge and uncertainty that can arise from the need to appraise multiple options (Patell *et al.*, 2008). Educators should therefore aim to provide structure without excessively controlling learning (Reeve and Jang, 2006; Ryan *et al.*, 2023).

Student self-perceptions of competence and self-efficacy have been shown to be significantly correlated with achievement (Cigdem, 2017; Gurer and Tokumaci, 2020), with positive past experiences and use of a greater variety of learning strategies found to be predictive of task performance (Wilson and Narayan, 2016). Increasing learners' perceived self-efficacy, which significantly affects satisfaction, is highly dependent on the student–educator relationship (Yu, 2022). Effective educators facilitate the relationships and supportive environments required to increase learner confidence and ensure timely progression (Gomes and Mendes, 2014).

Relationships are a vital component of student motivation. Educator–learner dialogue and reactive instructional intervention influence learning and student satisfaction (Sher, 2009; Eom and Ashill, 2016). In programming education, socially orientated practices such as peer-assisted and group-based learning can be beneficial (Altinas *et al.*, 2016; Garcia, 2021). Social learning can also pose challenges, however, particularly with regard to unequal levels of engagement that can lead to negative emotions like frustration (Goswami *et al.*, 2023).

Emotions and emotional regulation are closely linked with learner motivation, with clear impacts on academic performance (Yu *et al.*, 2022) and commitment (Hagenauer *et al.*, 2018). Learner affect can be divided into five key categories: engagement, curiosity, boredom, confusion and frustration (Bosch and D'Mello, 2017). Positive emotions are linked with high-quality learning and increased learner effort and persistence (Lin *et al.*, 2023), whereas negative emotions are linked with less favorable outcomes (Linnenbrink-Garcia *et al.*, 2016). Learner curiosity and “joyous exploration” can be encouraged via autonomy of choice (Schutte and Malouff, 2019) and increasing perceived self-efficacy can increase satisfaction (Yu, 2022).

## 2.2 Programming education and motivation

Programming education is frequently loosely defined in the literature. While programming in Computer Science and Software Engineering may include a large variety of subtopics (e.g. coding, algorithms and data structures), it is often referred to generically as “computer education” or “computer programming” (Goswami *et al.*, 2023; Loksa *et al.*, 2022) – although sometimes it concerns a specific programming language or level (Gurer *et al.*, 2019). Regardless of the exact learning context, learning to program presents its own distinct challenges and is often perceived as difficult (Becker *et al.*, 2023). Students must learn computational thinking – a logical problem-solving approach involving abstraction, debugging and iteration (Demertzi *et al.*, 2018). Many learners find such skills challenging and demotivation is a common problem (Medeiros *et al.*, 2019). Research has identified a

number of key factors that can support programming learner motivation, including good student–teacher relationships; clear explanations of key concepts; students working at their own pace; exercises that are meaningful to learners; games, challenges and competition to stimulate interest; building learner confidence; and ensuring learners feel comfortable seeking help (Gomes and Mendes, 2014). Although it has been found that gamification elements can motivate some learners, this is not the case for all learners (Pilkington, 2018; Facey-Shaw *et al.*, 2020).

Work that explicitly relates programming to well-established theories of motivation such as SDT is limited. Some studies have attempted to investigate how learner traits such as introversion or extraversion might influence learning motivation, though many of these rely on self-report measurements that could be subject to demand characteristics and results have been inconclusive (Dirzyte *et al.*, 2021). There is more convincing evidence that learners' attitudes to programming are significantly influenced by their academic achievement, sense of self-efficacy and perceived learning (Gurer *et al.*, 2019). Researchers have also drawn on existing theories about learning and motivation to help inform teaching practice, investigating the role of self-regulated learning strategies, co-regulation and metacognition in influencing motivation in programming learning (Loksa *et al.*, 2022; Goswami *et al.*, 2023).

Generative AI has opened up new ways for teachers to facilitate motivation, with LLMs being used, for example, to provide personalized learning resource recommendations and bespoke support for learners (Huang *et al.*, 2023; Humble *et al.*, 2023). There is considerable scope to investigate how LLM use for programming learning might intersect with what is known about learning motivation. This paper offers an initial foray into this intersection.

### *2.3 Large language models in programming education*

There are two distinct attitudes to generative AI tools that have emerged in the programming education community: to resist their use (by teaching fundamentals, creating AI-proof assessments and running paper-based exams) or to embrace them by integrating them into teaching (by giving personalized help to students, focusing on code reading and critique and teaching students to collaborate with AI) (Lau and Guo, 2023). Given the already widespread use of generative AI among students, completely resisting its use is likely to be a failed endeavor, and some educators are embracing and welcoming it (e.g. Porter and Zingaro, 2023). Because there is still a gap between what is taught on computer science courses and the expectations of industry (Craig *et al.*, 2018), incorporating generative AI into programming teaching may help better prepare students for employment.

Early research on the integration of generative AI into programming education has investigated using it to create programming exercises, generate explanations for complex concepts and provide bespoke content for individual learners (Denny *et al.*, 2023). Barke *et al.* (2022) found that programmers interact with generative AI in two main ways: in “exploration” mode (to learn new concepts) and “acceleration” mode (where they are already capable, but AI helps them achieve goals faster). One study with beginners found that using generative AI can provide code suggestions as a starting point, saving the need to search extensively online (Vaithilingham *et al.*, 2022), while Kazemitabaar *et al.* (2023) noted that the use of AI code generators helped learners feel less stressed and more engaged while significantly increasing performance on authoring tasks – though those with some prior knowledge might benefit more. Peng *et al.* (2023) found that less experienced programmers benefited more, but there was nonetheless a productivity gain for developers across all experience levels, highlighting a need for further research in this area. Indeed, one serious potential caveat to all the documented positive impacts of any AI tools is that of novelty

effect, “when the simple newness of a technology causes a rise in motivation or in achievement” as recognized by [Fryer et al. \(2017, p. 463\)](#). More longitudinal studies of the impacts of AI tools on motivation and learning are therefore necessary.

The use of these new tools also presents challenges ([Becker et al., 2023](#)). Learners may be tempted to cheat during formal assessment, using LLMs to generate code that they should be able to write themselves; uncritically accept LLM-generated content, even if it is inaccurate or biased ([Lau and Guo, 2023](#)); and have trouble fixing errors in AI-generated code ([Vaithilingham et al., 2022](#)). There are also concerns that overreliance on generative AI early in the learning process may hinder the development of the meta-cognitive skills essential to good programming ([Denny et al., 2023](#)) and reduce satisfaction derived from solving a problem without support ([Kazemitabaar et al., 2023](#)). While the significant benefit of AI as an assistive tool is clear, the risk that overreliance on it might lead to impairment of learning is recognized by both students and educators, underlining the need for an informed, balanced approach to its educational use ([Silva et al., 2024](#); [Husain, 2024](#)). Indeed, the LLM revolution has potential parallels with technology’s impact on math education: “the recent history of Math education and [its] struggle with technology suggests that fluency, which includes basic skills, is important regardless of the tools available” ([Jacques, 2023, p. 46](#)). Ultimately, learners still need to learn how to understand code and problem-solve to become effective programmers.

### 3. Methodology

In early 2023, a series of semistructured interviews were conducted with six Software Engineering educators (E1–E6) and 12 students (S1–S12) enrolled on a degree course (Software Engineering, Product Management or Interaction Design) at CODE University of Applied Sciences. The initial aim of the research was to explore experiences of teaching and learning to inform the design and content of an online learning platform for programming. Educators and students with recent experience teaching or learning programming were recruited as a purposeful sample via institutional social networking channels and announcements made in class. The investigation was deliberately focused on learning a programming language rather than on Software Engineering as a whole. This provided valuable insights from students learning programming on other study programs, not just those studying Software Engineering. However, the interviewees will likely have learned or taught programming in tandem with various other Software Engineering and Computer Science concepts. Student interviewees were between 22 and 36 years old, five identified as female and seven as male, and all had undertaken programming units as part of their studies, though they varied in experience level.

The student and educator interview schedules (see Appendix in Supplementary Files) were collaboratively designed by the three authors and were composed of broad, open questions regarding student engagement with programming learning content. The interviews lasted between 30 and 45 min and were conducted in person or using video conferencing software. The interview recordings were transcribed (using software and human correction), and the data set was reviewed using thematic analysis ([Braun and Clarke, 2006](#)). Each researcher undertook an initial familiarization process with the entire data set, adding notes to their own versions of the documents to independently highlight points of interest and develop a list of semantic and latent code suggestions. The team then met to compare and discuss notes and agree a code set. Each researcher then reviewed the entire data set again, this time using shared transcripts to apply agreed codes and note areas of commonality and divergence across the data set. Following this, the team met to confirm the reliability and relevance of the codes and identify potential superordinate themes.

Finally, the researchers collaboratively reviewed the data set to identify data extracts illustrative of the themes.

Surprised by the frequency of comments about generative AI use (which had not been the initial focus of the research), the team decided to conduct a wider survey of the student population to gain further information about LLM use and learning motivation. A questionnaire was designed to explicitly ask about LLMs and motivational factors in the context of learning programming (see Appendix in Supplementary files). It was deployed via Google Forms in late 2023, via the same communication channels used to recruit interview participants. This resulted in 44 completed questionnaires. Of the respondents, 29 identified as male and 15 female; 23 were aged 18–24, 18 were 25–34 and three were 35 or older; 35 studied Software Engineering, 4 Interaction Design, 4 Product Management and 4 Software Engineering and Product Management combined.

The results presented below combine findings from these two data sets (interviews and survey) to explore motivation in programming learning and the educational implications of LLM use.

#### 4. Results and discussion

The thematic analysis identified three superordinate themes that characterized the experiences of interviewees and survey respondents: bespoke learning, affect and support. These map well to the three basic psychological needs of SDT – autonomy, competence and relatedness – providing a useful framework via which to investigate learner motivation in contemporary programming education:

*Autonomy and bespoke learning:* There was a clear indication in our data sets that individual preferences and needs at various stages of the learning process impacted motivation and learning effectiveness, meaning that some level of bespoke learning is necessary to facilitate effective programming education.

*Competence and affect:* Our data sets showed that positive and negative emotions impacted learners' sense of self-efficacy and, therefore, the effectiveness of learning.

*Relatedness and support:* Our data sets underlined the ongoing importance of social aspects to effective learning, particularly with regard to support from peers and educators.

The extent to which interviewed students were already using LLMs was unexpected. Because this seemed to be impacting motivation and learning effectiveness, the questionnaire was distributed to see whether this trend was representative of a larger group. And indeed, 34 (77%) respondents reported that they were using LLMs “very frequently” or “frequently,” with 24 (56%) stating that LLMs had increased their motivation.

The three thematic categories are discussed below, alongside a consideration of the current impacts of LLM use on programming education. The implications of these findings for educators implementing LLM use is given in Section 5.

##### 4.1 *Autonomy and bespoke learning*

SDT underlines the importance of enabling students to make self-determined choices to enhance intrinsic motivation (Patall *et al.*, 2008; Reeve *et al.*, 2002), task performance (Murayama *et al.*, 2015) and curiosity (Schutte and Malouff, 2019). That said, too much choice can be overwhelming, and it is therefore still beneficial to provide a structure within which students can choose (Ryan *et al.*, 2023).

4.1.1 *Different programming languages and learning preferences.* While Python, HTML, CSS, JavaScript, Flask, Swift and SQL are actively taught at CODE University of Applied Sciences to varying degrees, students also mentioned learning PHP, Go, Java, C++ and Ruby, thereby exercising autonomy beyond the languages taught on the set curricula.

---

Survey respondents named 11 different languages as their primary programming language. This variety can be a problem for educators, who are likely expert in only a few languages, but have to teach groups using various other languages:

We can't give feedback in depth to any programming language, because we only know a few of them well enough to give feedback, at least more than superficial feedback. [. . .] When I present something, I present it in one language and they might not be too familiar. So I'm kind of teaching two things at once. (E1)

This creates an obstacle to students being able to focus on what they are really interested in, which can even lead them to drop the module; because of the impossibility of catering to every language simultaneously, educators “have to make some kind of restrictions [. . .] and every restriction we put in loses some students” (E1). It is clear that there is a gap between student will for autonomy to go beyond the curriculum and educator expertise – not because of lack of educator expertise, but rather because of the wealth of possible languages and focus points in programming.

Learners also have individual preferences regarding the kinds of resources used to support their learning. Some found videos more engaging and easier to “skip through” – “In a video I can go to double speed, stop when I want, repeat that part I didn't understand 10 times” (S3) – while others preferred text for a similar reason: “I prefer text to videos, they're just much more scannable. I can jump exactly to the part that I want to; I can skim them fairly quickly” (E1). Interviewees also mentioned preferences for particular content producers they trusted or liked. Given the tremendous amount of resources available today, it is impossible for educators to curate high-quality resources on every topic and language and in various media types to cater to these diverse learner interests and preferences.

*4.1.2 Progression from beginner to intermediate or advanced.* The interviews and surveys showed that students' needs change at different stages of the learning process. There is generally a need or desire for more structured learning as a beginner, with a progression to more unguided, exploratory learning as students gain experience. While 14 survey respondents said they prefer a completely unstructured, exploratory approach, the rest (30) said they preferred structure as a beginner or preferred some mix of structured and exploratory learning. One interviewee reported that, as a beginner, they required “the structure of the topics [. . .] I needed to learn” before “trying to figure out some projects [. . .] that actually target some of these topics” (S4). While it is beneficial to provide a course structure, especially for beginners, SDT literature makes a distinction between structure (which still allows for some freedom of choice) and restrictive control (Ryan *et al.*, 2023), with the former having more positive outcomes in terms of learning and motivation (Reeve and Jang, 2006). Indeed, facilitating a sense of choice can be highly motivating for students; e.g.:

The main thing is to have an interesting project. [. . .] I find it really discouraging if I'm following a tutorial to do exactly what they're doing. And as long as I have my own project to look forward to, this is very encouraging for me. [. . .] If it's a website, you know, if it's some plug-in, whatever it is, if it's something personal that I would like to see working. (S3)

Learners at different levels will also have differing needs in terms of the complexity and detail of explanations. Use of LLMs can be used to tailor explanations to the learner's level and thus make the content not only more “accessible,” but also more “interesting,” as this learner recognizes:

It is very, very accessible to ask, to have AI explain a concept to you, and I found that this is very successful because you can also tweak the way that it explains it, which sometimes makes it more entertaining, more understandable. And just a little more interesting in general. (S11)

#### 4.2 Competence and affect

In line with other research (Zatarain Cabada *et al.*, 2020), our data sets indicated that emotional states are strongly linked with learning achievement and student self-efficacy, with negative emotions (insecurity and frustration) creating obstacles to learning and positive emotions being associated with increased confidence and motivation. Mentions of affect in the interviews fell into three categories: insecurity or lack of confidence; boredom or frustration; and enjoyment and satisfaction. The drive for competence was clear from our survey results, with 35 (80%) respondents reporting that feeling capable was an important motivating factor for them – jointly the highest-rated motivation factor (along with “career goals”) – and 28 (64%) stating that feeling “in control” was important for them.

**4.2.1 Insecurity/lack of confidence.** Feelings of insecurity or a lack of confidence in the interviews included fear, nervousness or overwhelm; vulnerability; a sense of imposter syndrome; feeling intimidated, insecure or stupid; and the wish for a friendly and safe environment rather than a “toxic” or mean environment (particularly with regards to online forums). A sense of imposter syndrome came up multiple times, where the student would negatively compare themselves to others: “you see all these other people are hacking away in their terminals, super scary. Like, I can’t do that” (S1); “Oh my God, what am I doing here? You know, am I really so dumb? You know, these people [. . .], they know everything” (S3).

Negative feelings led interviewees to avoid certain subjects (“I took more of the business courses than [. . .] informatics courses in the end, because I was fearing that they were too hard for me” (S4)), to not contribute in class or drop out altogether (S3), or to not feel comfortable asking questions when they did not understand things:

I felt nobody was asking questions, or whoever was, was people that already knew and asked pretty advanced questions, and then I was a little intimidated to ask. But then if I would ask another student, they’re like, “Oh, I have no idea either.” And I was like, “Okay so why aren’t you asking?” and I feel like [. . .] everybody was nervous to ask. (S9)

According to one educator, such feelings create an obstacle not just for beginners, but even for expert programmers or educators: “they [the students] are so scared. And all of us are the same”; “insecurity is actually a big known issue even for professional developers” (E3).

Although 10 (23%) of our survey respondents stated that “feeling too embarrassed to ask questions for fear of looking stupid” was not at all a barrier to learning for them, the remainder indicated it was. For six (14%), this was a “significant barrier.” This is a problem with which LLMs can help, as a couple of interviewees identified. One said that “ChatGPT is extremely gentle, you know? It’s so polite” (S3), contrasting it to online forums that can be “a little bit toxic” (S3) while another liked the way that one does not need to worry about looking silly in front of others:

With ChatGPT you can just ask all the nonsense that you were always too shy to ask [. . .], there’s no judgment. The other day, I was kind of like “How does multithreading work in Swift?” and I couldn’t get the difference [in] how it works in Swift [compared] to others and I was just like, “explain it like I’m a high school student.” And then it just tells you. (S6)

With generative AI, students can “safely” ask questions without making themselves vulnerable or revealing they do not know something already. They can fill the gaps in their knowledge and therefore progress on their learning journey without fear of negative judgment. A higher sense of learner self-efficacy (that can be engendered by LLM use, as shown above) is positively correlated with better task performance – both in general (Wilson and Narayan, 2016) and specifically in the context of computer programming (Cigdem, 2017; Gurer and Tokumaci, 2020). However, the literature on the correlation between negative emotions and effort or persistence is inconclusive (Lin *et al.*, 2023). Although the link

---

between negative emotion and academic achievement is complicated, some of our learners expressed a clear reluctance to demonstrate ignorance in public for fear of embarrassment, which represents a clear barrier to effective learning.

Our students were already using LLMs to provide safe, nonjudgmental support, likely increasing their sense of competence and positive forms of affect. That said, information from LLMs is not always reliable and students may not exercise due diligence with regard to accuracy checking – “The downside is, [ChatGPT] might give the wrong answer. You’re not checking if it’s right or wrong” (S3) – a factor we emphasize in the recommendations section below.

*4.2.2 Boredom or frustration.* Negative feelings toward the nature of the task of programming – boredom and frustration – were mentioned by seven interviewees. Only 13 survey respondents said boredom or frustration were not at all a barrier, with the rest (31) recognizing these emotions as some kind of barrier. As [Bosch and D’Mello \(2017\)](#) found, such feelings are negatively correlated with learning in programming education. Among our participants, feelings of boredom or frustration were sometimes caused by a mismatch between the student’s level and what was being taught or attempted (a mismatch that was also named as a slight or significant barrier by 35 survey respondents):

Sometimes it feels hard to match [a course] to my actual learning approach and experience. So because I am maybe faster or slower depending on the course, [...] I always had issues after the first or second lecture to make it fit or find also the fit to what I’m doing [...]. So very quickly I’m a bit demotivated or also a bit bored because it doesn’t really fit well. (S2)

Feelings of frustration were also associated with finding and fixing bugs: “You get errors and you don’t know how to debug them. This is extremely frustrating” (S3). AI tools can assist programmers to fix bugs much more quickly than seeking answers in forums, leading to decreased frustration and faster progress:

I once posted my question on Stack Overflow, and of course, got voted down because I wasn’t rigorous in how I asked and I wasn’t posting all the code [...]. And that’s where I just feel like these AI tools, text generators, are just amazingly helpful. (S6)

Indeed, 37 (86%) of our survey respondents said they use AI tools to debug code. However, one learner hinted at a potential disadvantage of this improved efficiency:

I try to have this mindset that, when I’m stuck with a problem, the longer I need to solve it, the longer it will last in my brain. I think that’s how memory works in general. So if you are stuck with a thing, don’t get frustrated. (S8)

For some learners, LLMs could provide an opportunity to accelerate learning progress while reducing boredom and cognitive load. This would enable them to swiftly reach the sort of complex activities that are more likely to stimulate interest and engagement ([Mayer, 2009](#)). Interestingly, other learners, as in the quotation above, may see value in getting “stuck” for a while – in taking time to work out a solution for themselves. Of our survey respondents, 37 (86%) said they still want to learn to write code themselves, rather than relying on LLMs to do it for them. This approach may also foster more effective learning since it ensures students are capable of engaging in elaborative interrogation (where learners must provide an explanation for why something is correct), which has been linked with significantly improved learning outcomes ([Dunlosky et al., 2013](#)).

*4.2.3 Enjoyment and satisfaction.* Positive feelings of enjoyment and satisfaction were mentioned ten times during the interviews, whereas only two survey respondents rated “creative satisfaction” as not at all a motivating factor. Several interviewees reported feelings of enjoyment and satisfaction associated with having successfully created

something: “I very much enjoy understanding how the things I use work. And it’s even more exciting being able to [...] build it as well” (S11);

What I found attractive about web development is that you make something, you put it online, and potentially 2 billion people have access to it. So it’s really crazy to think about it and that was kind of a very nice incentive, you know? [...] I created something; something that didn’t exist now exists and people can use it. So [...] for me, it’s like an expressive tool. [...] I like music and I mostly write my own stuff, so it’s the same thing. There was nothing, [...] I wrote some music, now there’s something there. So for me, it’s the same kind of joy I get from programming. (S3)

Of course, to experience this kind of joy or satisfaction, one has to be able to create a functioning program that does something useful or interesting. One educator recognized that reaching such a level can be particularly difficult during online learning, via reference to Bloom’s taxonomy – which divides learning into the stages of Remember, Understand, Apply, Analyze, Evaluate and Create and has been recognized as a useful frame for programming learning (Sobral, 2021):

When we’re talking about [...] the basic levels of understanding and remembering [within Bloom’s Taxonomy], I find [the] type of quick quizzes [built into online courses] quite helpful. If we talk then about coding, that’s kind of helpful for the very basics, but that’s not the goal. The goal is really then to be able to create the application and that, of course, is quite limited there. And that’s also where I often find with online courses I’ve done, [...] you very rarely have the kind of creative thing that you build yourself and then still have expert feedback on it. [...] The ability to then creatively transfer it and be comfortable enough with something to say, “not only have I done this course and got the certificate, but now I have a different problem set to work on and I want to apply those skills that I’ve learned [...]”—I find that that second step is often where I hit my own limits, despite 20 years of working with code. (E5)

Online courses, that is to say, may impart enough learning to pass the assessment, but learners may still lack the level of knowledge to be able to creatively apply what has been learned in a different context. A couple of interviewees also experienced this problem, both with live learning – “I felt that even though I understood the things, there was a problem with the applications” (S7) – and with online learning:

I think the hardest part for me was this moment when you switch from an online course, where you’re learning something very tailored, to the project, where [...] there’s nothing and you have to decide, you know, where to start and how to translate now what you’ve learned online to your actual project. [...] That was a huge step. (S1)

As E5 states above, online courses in particular may leave learners at the lower levels of Bloom’s Taxonomy – which, with regards to programming, would be the ability to read and understand code, but not to write it themselves – and thus in a state of confusion or frustration rather than enjoyment and satisfaction. However, with the advent of LLM-assisted programming, progress toward the Evaluate and Create steps of the taxonomy can be accelerated, as one student realizes:

I don’t find that [being able to write code myself is] a priority for me, because now [...] having Chat[GPT], and me having my ability to recognize what’s written, I don’t need to be able to actually replicate, like to create it from zero. I can ask for it and then recognize what is done. (S5)

While more research needs to be done on the topic, Kazemitabaar *et al.* (2023, p. 1) found that using LLM programming tools “significantly increased code-authoring performance” (the “Create” of Bloom’s Taxonomy), “while not decreasing performance on manual code-modification tasks” (“Remembering” and “Understanding”). By freeing learners from some

of the more mundane aspects of learning to code, they are enabled to concentrate on more interesting creative activity and experience satisfaction. Of our survey respondents, 24 (54.5%) said they took on more creative or complex projects with the aid of LLMs than they would have otherwise.

Some interviewees described how feeling that one understands how things work combined with a recognition of the usefulness of programming as a skill in today's world can lead to an increased sense of independence and agency: "I think it's almost like a superpower to know it these days, because we're surrounded by technology everywhere and [. . .]. I very much enjoy understanding how the things I use work" (S11):

I could just do so many different things with [Python]. And then had so many project ideas that I could use this language on that I just kept using it for different projects and the projects also ranged from a back-end system to really just trying to do some data analytics with it. (S4)

I want to be able to, for example, build websites by myself and to understand how it works. Or if I work together with some other software engineers, I would like to understand what they do [. . .]. I would like to know what is going on. And it's not [. . .] to control; [. . .] it's just because I'm actually curious. (S5)

Meanwhile, all of our survey respondents cited the usefulness of programming as at least a slight motivation, and 35 (80%) cited wanting to feel capable as an important or very important motivator.

### 4.3 Relatedness and support

While our participants varied in the degree to which they value peer support, only seven (16%) survey respondents said that peer support was not a motivating factor for them at all and 35 (80%) said that "explaining concepts to each other" was "useful" or "very useful." Thirteen (30%) reported that access to LLMs had reduced the amount they interact with teachers and peers, although far more (63%) stated that LLMs had not reduced or had even increased (7%) such interactions. It was clear from the data that social support and peer accountability are still important aspects of learner motivation.

*4.3.1 Social support.* Whilst learning to program can be a solitary endeavor, many interviewees noted the value of peer learning, particularly as they progressed to a more intermediate level:

I think I always learned alone until I am at a certain level where I'm also open to having this more collaborative approach [. . .] where I feel like I could help someone else in some parts of it and then I could profit from someone else helping me. Because I also know that explaining stuff to other people helps me understand it better; [. . .] really understanding it oftentimes came from trying to explain it to other people. (S4)

Now that I'm a little bit more experienced, I really enjoy doing team projects [. . .]. For example, the other day we were doing peer coding—two people, one screen—and it was one of the best learning experiences I had in a while, but it's now because I have concepts to offer, he has concepts to offer, but when we both have no idea [as beginners], I feel like at least for me back then I never really asked peers. (S6)

Meanwhile, 35 (80%) of our survey respondents said that explaining concepts to peers was helpful or very helpful. Educators also recognize the benefits of in-person support when it comes to feelings of insecurity or lack of confidence:

This kind of negative loop you go into [when struggling with a problem], is something that you run into even more if you're alone at home. If you have friends that pick you up at that point, or teachers or whatever, if there's a community around you, you're way less likely to give up. (E2)

The value of communities of practice and social interaction in improving learning performance and engagement is well known (Rienties and Toetenel, 2016). Because the need to connect with others is a core psychological drive, activities that promote social connection are likely to increase motivation, especially those that also help increase a sense of self-efficacy – as hinted by S4 above, who felt explaining to others improved their own understanding.

This presents a likely limitation of LLMs; human support can provide a sense of community and mutual connection that AI-generated feedback or answers – no matter how efficient or accurate – simply cannot replace:

I think it can be nice to have some kind of peer-to-peer still, some kind of possibility to [...] get connected to people. [...] Chat[GPT] now gives you all the answers, [...] but maybe, some people's code still need[s] to [be] explain[Ed.] to you. (S5)

**4.3.2 Peer accountability.** Peers and educators are often a source of motivation for learners. Providing effective extrinsic motivation is one area with which AI is likely less able to help, particularly in the form of external accountability. Forms of extrinsic motivation that came up in our interviews included the promise of certificates or qualifications, work requirements and career goals, progress tracking and external accountability (deadlines, exams and group learning). Both students and educators recognized the value of being in a structured learning environment with certain expectations that have to be met:

Deadlines [are] important. [...] These hand-ins every semester forced me to finish a project that has these five checkmarks like, make a database call, refactor something [...], and have to have it somewhere by a [certain] day. [...] It's not this online bootcamp tutorial that you will never finish because there is no obligation. (S6)

One educator reported that students “request [...] more structure, more pressure, and more—‘Give me a homework assignment for next week [...], so that I can feel that I'm progressing and doing my class’ – more of a school-like environment” (E5). In our survey, 43% of respondents reported that deadlines were “essential” for keeping them on track, while 50% said that they were “helpful.” Only three (7%) said they could do just as well without external accountability altogether.

The value of social learning should not be underestimated. As one educator noted, “people are motivated by hanging out with other people” and it is important to create “groups where people can meet or [...] structures where people can motivate each other, hold each other accountable” (E6).

## 5. Conclusions and recommendations

### 5.1 Conclusions

In investigating the experiences of students and educators in programming education, we found three salient thematic categories that impact learning: bespoke learning, affect and support. We also found that LLMs are already being used extensively by students with current and potential implications for learning and motivation. Framing our findings using SDT yields clear implications for how educators might work with LLMs to enhance programming education. The following three subsections summarize the interview and survey findings and Section 5.2. gives recommendations for educators based on these findings and the current state and limitations of LLMs.

*5.1.1 Autonomy and bespoke learning.* Interviewees and survey respondents showed a great diversity of learner interests and needs—even within a relatively small sample size. Students were interested in a wide range of programming languages, had diverse preferences in terms of learning modes (e.g. via text or video, alone or in groups, with structured or exploratory learning modes), and enjoyed doing projects according to individual interests. These needs also changed or developed as learners progressed from beginner to intermediate or advanced. The data sets made it clear that effective programming education must cater to different learning preferences and levels to best engage students. LLMs can effectively provide such bespoke learning, as we outline in our recommendations below.

*5.1.2 Competence and affect.* In line with other research (Zatarain Cabada *et al.*, 2020), our interviews and survey showed the key role that positive and negative emotions play in the experience of learning to program, with negative emotions creating obstacles and positive emotions encouraging and motivating students. Participants reported the experience of successfully creating something via programming as engendering feelings of enjoyment and satisfaction, which acts as an intrinsic motivator to learn more. In terms of negative affect, the data sets revealed insecurity or lack of confidence as a significant barrier for learners, preventing students from asking questions or contributing in class for fear of looking stupid. Boredom and frustration also impeded the learning process, particularly when course content was mismatched to a student's abilities, or during particularly tricky or repetitive tasks. Students already reported using LLMs to reduce or avoid both of these negative forms of affect, increasing their sense of self-efficacy and capability.

*5.1.3 Relatedness and support.* Our interviews and survey confirmed the ongoing value of human support in the age of LLM-assisted programming. Participants appreciated having forms of external accountability (deadlines, assessments, etc.); they benefited from educator-provided learning structure and advice; and they found peer learning useful for maintaining motivation and for increasing learning via explaining things to peers. These findings underline a potential limit to current LLM technology in terms of enhancing learner relatedness and motivation.

## 5.2 Recommendations for educators

Our data confirmed that both intrinsic and extrinsic motivation play important roles in programming learning and should be combined to maximize learning outcomes. Our research indicated that the use of LLMs in programming education may work to increase intrinsic motivation. Using LLMs to empower students to write more advanced, complex programs more quickly and to explain difficult concepts can increase autonomy and competence, and may even motivate students to go deeper into the material. Meanwhile, our data sets indicated that effective forms of extrinsic motivation (particularly the external accountability of assignments, exams and the expectations of the workplace) are best provided by humans and likely cannot currently be effectively replaced by generative AI. Similarly, social interaction with trusted experts and peers enhances learner motivation and LLMs have limitations in this regard.

Informed by our interviews, surveys and existing literature on the current state and limitations of LLMs, we make the following recommendations, at least some of which are likely applicable to education beyond programming.

LLMs can help increase learner motivation by:

- increasing autonomy by enabling students to work with whichever programming language and project they choose and providing bespoke support tailored to individual interests, capabilities and learning preferences;

- increasing competence and self-efficacy by explaining difficult concepts or answering questions students might be embarrassed to ask;
- increasing competence and self-efficacy by enabling students to progress to writing complex programs more quickly; and
- decreasing negative affect (boredom and frustration) by minimizing time spent on boring or repetitive tasks and reducing incidences when learners get stuck on difficult concepts or learning challenges.

Educators can work with LLMs to enhance programming education by:

- framing LLMs as a tool that can assist students to progress more quickly in programming and be better prepared for contemporary industry expectations;
- teaching students how to procure good-quality answers from LLMs;
- teaching students how to read, understand and debug AI-generated code;
- teaching students how to check AI-generated answers for accuracy;
- supporting intermediate students in choosing a language and project that excites them, for which LLMs can then provide bespoke support;
- providing opportunities for students to work in pairs or groups to facilitate social learning; and
- providing and facilitating forms of accountability tailored to students' level and interests (assignments, exams and group learning).

Generative AI has already begun to significantly transform how professional programmers work, with 92% of US-based developers already using AI tools (Shani, 2023). Students are already using generative AI to support their learning and, given its effectiveness and availability, they are likely to continue to do so. Programming education should therefore incorporate the judicious use of LLMs in order to best prepare students for the reality of industry and the coming generative-AI-assisted world.

## 6. Limitations and areas for further research

This research was conducted with a relatively small number of participants drawn from a highly specific context, with learners who are likely to be relatively highly motivated to learn programming skills. The small purposeful samples were appropriate given the research focus on the needs and challenges involved in learning programming in a higher education context. However, further research with more diverse groups of learners (that vary by age, educational experience, subject area, etc.) would give a broader sense of how learners in different contexts might benefit from the integration of LLMs into programming education.

Although our interviews and surveys revealed a variety of ways learners are already making use of LLMs in their learning (providing straightforward explanations, generating possible solutions, providing feedback on incorrect code, etc.), an assessment of the effectiveness of these various strategies is beyond the scope of this study. There is a need for further research to investigate the relative value of different types of LLM use for learners.

Finally, although qualitative research provides insights regarding subjects' thoughts, perspectives and feelings, which can provide valuable information to educators seeking to design effective learning programs, our study cannot claim that LLM use has a causative, positive effect on the efficacy of programming education. Carefully designed quantitative and mixed-methodology investigations are needed to construct the robust evidence base that would enable educators to integrate LLMs in a way that will maximize learning potential.



- Eom, S.B. and Ashill, N. (2016), "The determinants of students' perceived learning outcomes and satisfaction in university online education: an update", *Decision Sciences Journal of Innovative Education*, Vol. 14 No. 2, pp. 185-215.
- Facey-Shaw, L., Specht, M., van Rosmalen, P. and Bartley-Bryan, J. (2020), "Do badges affect intrinsic motivation in introductory programming students?", *Simulation and Gaming*, Vol. 5 No. 1, pp. 33-54, doi: [10.1177/1046878119884996](https://doi.org/10.1177/1046878119884996).
- Fryer, L.K., Ainley, M., Thompson, A., Gibson, A. and Sherlock, Z. (2017), "Stimulating and sustaining interest in a language course: an experimental comparison of Chatbot and human task partners", *Computers in Human Behavior*, Vol. 75, pp. 461-468, doi: [10.1016/j.chb.2017.05.045](https://doi.org/10.1016/j.chb.2017.05.045).
- Garcia, M. (2021), "Cooperative learning in computer programming: a quasi-experimental evaluation of jigsaw teaching strategy with novice programmers", *Education and Information Technologies*, Vol. 26 No. 4, pp. 4839-4856.
- Gilch, H., Beise, A.S., Krempkow, R., Müller, M., Stratmann, F. and Wannemacher, K. (2019), "Digitalisierung der hochschulen: ergebnisse einer schwerpunktstudie für die expertenkommission forschung und innovation", [Digitalization of universities: Results of a focus study for the Expert Commission on Research and Innovation], IDEAS Working Paper Series from RePEc.
- Gomes, A. and Mendes, A. (2014), "A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations", The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings, Vol. 1.
- Goswami, L., Senges, A., Estier, T. and Cherubini. (2023), "Supporting co-regulation and motivation in learning programming in online classrooms", *Proceedings of the ACM on Human-Computer Interaction*, Vol. 7 No. CSCW2, pp. 1-29.
- Gurer, M.D. and Tokumaci, S. (2020), "Factors affecting engineering students' achievement in computer programming", *International Journal of Computer Science Education in Schools*, Vol. 3 No. 4, pp. 23-34, doi: [10.21585/ijcses.v3i4.74](https://doi.org/10.21585/ijcses.v3i4.74).
- Gurer, M.D., Cetin, I. and Top, E. (2019), "Factors affecting students' attitudes toward computer programming", *Informatics in Education*, Vol. 18 No. 2, pp. 281-296, doi: [10.15388/infedu.2019.13](https://doi.org/10.15388/infedu.2019.13).
- Hagenauer, G., Gläser-Zikuda, M. and Moschner, B. (2018), "University students' emotions, life-satisfaction and study commitment: a self-determination theoretical perspective", *Journal of Further and Higher Education*, Vol. 42 No. 6, pp. 808-826, doi: [10.1080/0309877X.2017.1323189](https://doi.org/10.1080/0309877X.2017.1323189).
- Huang, A.Y.Q., Lu, O.H.T. and Yang, S.J.H. (2023), "Effects of artificial intelligence-enabled personalized recommendations on learners' learning engagement, motivation, and outcomes in a flipped classroom", *Computers and Education*, Vol. 194, doi: [10.1016/j.compedu.2022.104684](https://doi.org/10.1016/j.compedu.2022.104684).
- Humble, N., Boustedt, J., Holmgren, H., Milutinovic, G., Seipel, S. and Ostberg, S. (2023), "Cheaters or AI-enhanced learners: consequences of ChatGPT for programming education", *Electronic Journal of e-Learning*, available at: [www.diva-portal.org/smash/get/diva2:1821072/FULLTEXT01.pdf](http://www.diva-portal.org/smash/get/diva2:1821072/FULLTEXT01.pdf)
- Husain, A.J.A. (2024), "Potentials of ChatGPT in computer programming: insights from programming instructors", *Journal of Information Technology Education: Research*, Vol. 23, p. 2, doi: [10.28945/5240](https://doi.org/10.28945/5240).
- Jacques, L. (2023), "Teaching CS-101 at the dawn of ChatGPT", *ACM Inroads*, Vol. 14 No. 2, pp. 40-46, doi: [10.1145/3595634](https://doi.org/10.1145/3595634).
- Jones, A.R. (2013), "Increasing adult learner motivation for completing self-directed e-learning", *Performance Improvement*, Vol. 52 No. 7, pp. 32-42, doi: [10.1002/pfi.21361](https://doi.org/10.1002/pfi.21361).
- Kazemitaabar, M., Chow, J., Ka To Ma, C., Ericson, B.J., Weintrop, D. and Grossman, T. (2023), "Studying the effect of AI code generators on supporting novice learners in introductory programming", Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, doi: [10.1145/3544548.3580919](https://doi.org/10.1145/3544548.3580919)

- Lau, S. and Guo, P.J. (2023), "From 'ban it till we understand it' to 'resistance is futile': how university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and Github copilot", ACM Conference on International Computing Education Research (ICER), available at: [www.samlau.me/pubs/cs-instructors-adapting-to-chatgpt-copilot-ai-tools\\_ICER-2023.pdf](http://www.samlau.me/pubs/cs-instructors-adapting-to-chatgpt-copilot-ai-tools_ICER-2023.pdf)
- Lin, G.-Y., Liao, L.-W., Su, Z.-Y., Wang, Y.-M. and Wang, Y.-S. (2023), "What drives undergraduates' effort and persistence in learning programming", *Education and Information Technologies*, doi: [10.1007/s10639-023-11670-3](https://doi.org/10.1007/s10639-023-11670-3).
- Linnenbrink-Garcia, L., Pattall, E.A. and Pekrun, R. (2016), "Adaptive motivation and emotion in education: research and principles for instructional design", *Policy Insights from the Behavioral and Brain Sciences*, Vol. 3 No. 2, pp. 228-236.
- Loksa, D., Margulieux, L., Becker, B.A., Craig, M., Denny, P., Pettit, R. and Prather, J. (2022), "Metacognition and self-regulation in programming education: theories and exemplars of use", *ACM Transactions on Computing Education*, Vol. 22 No. 4, p. 39, doi: [10.1145/3487050](https://doi.org/10.1145/3487050).
- Mayer, R.E. (2009), *Multimedia Learning*, 2nd ed., Cambridge University Press, Cambridge and New York, NY.
- Medeiros, R.P., Ramalho, G.L. and Falcão, T.P. (2019), "A systematic literature review on teaching and learning introductory programming in higher education", *IEEE Transactions on Education*, Vol. 62 No. 2, pp. 77-90.
- Murayama, K. (2019), "Neuroscientific and psychological approaches to incentives", in Renniger, K.A. and Hidi, S.E. (Eds), *The Cambridge Handbook of Motivation and Learning*, Cambridge University Press, Cambridge, pp. 141-162, doi: [10.1017/9781316823279.008](https://doi.org/10.1017/9781316823279.008).
- Murayama, K., Matusumoto, M., Izuma, K., Sugiura, A., Ryan, R.M., Deci, E.L. and Matsumoto, K. (2015), "How self-determined choice facilitates performance: a key role of the metromedial prefrontal cortex", *Cerebral Cortex*, Vol. 25 No. 5, pp. 1241-1251, doi: [10.1093/cercor/bht317](https://doi.org/10.1093/cercor/bht317).
- Patall, E.A., Cooper, H. and Robinson, J.C. (2008), "The effects of choice on intrinsic motivation and related outcomes: a meta-analysis of research findings", *Psychological Bulletin*, Vol. 134 No. 2, pp. 270-300, doi: [10.1037/0033-2909.134.2.270](https://doi.org/10.1037/0033-2909.134.2.270).
- Peng, S., Kalliamvakou, E., Cihon, P. and Demirel, M. (2023), "The impact of AI on developer productivity: evidence from GitHub copilot", doi: [10.48550/arxiv.2302.06590](https://doi.org/10.48550/arxiv.2302.06590)
- Pilkington, C. (2018), "A playful approach to fostering motivation in a distance education computer programming course: behaviour change and student perceptions", *International Review of Research in Open and Distributed Learning*, Vol. 19 No. 3, pp. 282-298.
- Porter, L. and Zingaro, D. (2023), *Learn AI-Assisted Python Programming with Github, Copilot and ChatGPT*, Manning Publications, Shelter Island.
- Reeve, J. and Jang, H. (2006), "What teachers say and do to support students' autonomy during a learning activity", *Journal of Educational Psychology*, Vol. 98 No. 1, pp. 209-218, doi: [10.1037/0022-0663.98.1.209](https://doi.org/10.1037/0022-0663.98.1.209).
- Reeve, J., Jang, H., Hardre, P. and Omura, M. (2002), "Providing a rationale in an autonomy-supportive way as a motivational strategy to motivate others during an uninteresting activity", *Motivation and Emotion*, Vol. 26 No. 3, pp. 183-207.
- Renniger, K.A. and Hidi, S.E. (2019), "Interest development and learning", in Renniger, K.A. and Hidi, S.E. (Eds), *The Cambridge Handbook of Motivation and Learning*, Cambridge University Press, pp. 265-290, doi: [10.1017/9781316823279.017](https://doi.org/10.1017/9781316823279.017).
- Rienties, B. and Toetnel, L. (2016), "The impact of learning design on student behaviour, satisfaction and performance: a cross-institutional comparison across 151 modules", *Computers in Human Behavior*, Vol. 60, pp. 333-341, doi: [10.1016/j.chb.2016.02.074](https://doi.org/10.1016/j.chb.2016.02.074).

- Rienties, B., Giesbers, B., Tempelaar, D., Lygo-Baker, S., Segers, M. and Gijsselaers. (2012), "The role of scaffolding and motivation in CSCL", *Computers and Education*, Vol. 59 No. 3, pp. 893-906, doi: [10.1016/j.compedu.2012.04.010](https://doi.org/10.1016/j.compedu.2012.04.010).
- Ryan, Richard, M., Reeve, J., Kaplan, H., Matos, L. and Hyeon Cheon, S. (2023), "Education as flourishing: self-determination theory in schools as they are and as they might be", in Richard, M.R. (Ed.), *The Oxford Handbook of Self-Determination Theory*, Oxford University Press, Oxford, pp. 591-618.
- Schutte, N.S. and Malouff, J.M. (2019), "Increasing curiosity through autonomy of choice", *Motivation and Emotion*, Vol. 43 No. 4, pp. 563-570, doi: [10.1007/s11031-019-09758-w](https://doi.org/10.1007/s11031-019-09758-w).
- Schwartz, B. and Wrzesniewski, A. (2019), "Reconceptualizing intrinsic motivation: excellence as goal", in Renniger K.A and Hidi, S.E. (Eds), *The Cambridge Handbook of Motivation and Learning*, Cambridge University Press, Cambridge, pp. 373-393, doi: [10.1017/9781316823279.017](https://doi.org/10.1017/9781316823279.017).
- Shani, I. (2023), "Survey reveals AI's impact on the developer experience", The Github Blog, available at: <https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-experience/>
- Sher, A. (2009), "Assessing the relationship of student-instructor and student-student interaction to student learning and satisfaction in [a] web-based online learning environment", *Journal of Interactive Online Learning*, Vol. 8 No. 2, pp. 102-120.
- Silva, C., Ramos, F.N., de Moraes, R.V. and Santos, E. (2024), "ChatGPT: challenges and benefits in software programming for higher education", *Sustainability*, Vol. 16 No. 3, p. 1245, doi: [10.3390/su16031245](https://doi.org/10.3390/su16031245).
- Sobral, S.R. (2021), "Bloom's taxonomy to improve teaching-learning in introduction to programming", *International Journal of Information and Education Technology*, Vol. 11 No. 3, pp. 148-153, doi: [10.18178/ijiet.2021.11.3.1504](https://doi.org/10.18178/ijiet.2021.11.3.1504).
- Vaithilingham, P., Zhan, T. and Glassman, E.L. (2022), "Expectation vs. experience: evaluating the usability of code generation tools powered by large language models", paper presented at ACM CHI Conference on Human Factors in Computing Systems, New Orleans, April 30 – May 5, doi: [10.1145/3491101.3519665](https://doi.org/10.1145/3491101.3519665)
- Vansteenkiste, M., Simons, J., Lens, W., Soenens, B., Matos, L. and Lacante, M. (2004), "Less is sometimes more", *Journal of Educational Psychology*, Vol. 96 No. 4, pp. 755-764, doi: [10.1037/0022-0663.96.4.755](https://doi.org/10.1037/0022-0663.96.4.755).
- Wilson, K. and Narayan, A. (2016), "Relationships among individual task self-efficacy, self-regulated learning strategy use and academic performance in a computer-supported collaborative learning environment", *Educational Psychology*, Vol. 36 No. 2, pp. 236-253, doi: [10.1080/01443410.2014.926312](https://doi.org/10.1080/01443410.2014.926312).
- Xie, K., DeBacker, T.K. and Ferguson, C. (2006), "Extending the traditional classroom through online discussion: the role of student motivation", *Journal of Educational Computing Research*, Vol. 34 No. 1, pp. 67-89.
- Yu, J., Huang, C., He, T., Wang, X. and Zhang, L. (2022), "Investigating students' emotional self-efficacy profiles and their relations to self-regulation, and academic performance in online learning contexts: a person-centered approach", *Education and Information Technologies*, Vol. 27 No. 8, pp. 11715-11740, doi: [10.1007/s10639-022-11099-0](https://doi.org/10.1007/s10639-022-11099-0).
- Zatarain Cabada, R., Barrón Estrada, M.L., Ríos Félix, J.M. and Alor Hernández, G. (2020), "A virtual environment for learning computer coding using gamification and emotion recognition", *Interactive Learning Environments*, Vol. 28 No. 8, pp. 1048-1063, doi: [10.1080/10494820.2018.1558256](https://doi.org/10.1080/10494820.2018.1558256).

### Further reading

- Yu, Q. (2022), "Factors influencing online learning satisfaction", *Frontiers in Psychology*, Vol. 13, doi: [10.3389/fpsyg.2022.852360](https://doi.org/10.3389/fpsyg.2022.852360).

Altintas, T., Gunes, A. and Sayan, H. (2016), "A peer-assisted learning experience in computer programming language learning and developing computer programming skills", *Innovations in Education and Teaching International*, Vol. 53 No. 3, pp. 329-337.

Maarten, V., Simons, J., Lens, W., Soenens, B., Matos, M. and Lacante, M. (2004), "Less is sometimes more", *Journal of Educational Psychology*, Vol. 96 No. 4, pp. 755-764, doi: [10.1037/0022-0663.96.4.755](https://doi.org/10.1037/0022-0663.96.4.755).

### Supplementary material

The supplementary material for this article can be found online.

### Corresponding author

Samuel Boguslawski can be contacted at: [samuel.boguslawski@code.berlin](mailto:samuel.boguslawski@code.berlin)

---

For instructions on how to order reprints of this article, please visit our website:

[www.emeraldgroupublishing.com/licensing/reprints.htm](http://www.emeraldgroupublishing.com/licensing/reprints.htm)

Or contact us for further details: [permissions@emeraldinsight.com](mailto:permissions@emeraldinsight.com)