Understanding elements, strengths and challenges of explicit instruction for the teaching of computer programming (to post-graduate students)[1]

Pakiso Khomokhoana, University of the Free State, South Africa

ABSTRACT

Computer programming continues to be challenging despite numerous strategies and skills that researchers and instructors have shared over four decades. Using explicit instruction (EI) to help students learn and better understand computer programming presents a promising avenue for tackling this challenge. The aim of this study was twofold. Firstly, to identify elements to consider in an instructional strategy for teaching using the EI principles. Secondly, to identify strengths and challenges presented by the EI interventions in teaching computer programming to postgraduate Computer Science students. Collected data were analyzed through thematic analysis, and the results reveal nine major strengths and five main challenges related to EI. The study followed an integrated methodological approach where narrative data was collected through observations and asking questions. This study informs how improvements can be made in the future teaching of computer programming to enhance the quality of teaching using the principles of EI.

Keywords: computer programming, explicit instruction, explicit instruction steps, teaching strategies, computer science education

INTRODUCTION

Despite 40 years of research into how teaching computer programming can be improved, programming continues to be a challenge to students (Ko et al., 2019; Qian & Lehman, 2017). Over this period, computer programming researchers and instructors have suggested several strategies and specific skills that could be used to address this challenge. The teaching and learning strategies include explicit step-by-step strategies (Ko et al., 2019; LaToza et al., 2020; Xie et al., 2018), peer instruction (Porter et al., 2013), modelling (Middendorf & Pace, 2004; Wood et al., 1976), pair programming (Porter et al., 2013; Williams et al., 2000), and use of worked examples (Griffin, 2015; Sweller et al., 2011). The specific skills include doodling, walkthroughs, pattern recognition (Fitzgerald, Simon & Thomas, 2005), the teaching of strategic skills (O'Dell, 2017), the teaching of cognitive skills (Ko & Uttl, 2003; Von Mayrhauser & Vans, 1996), and the teaching of (meta)cognitive processes/strategies (Khomokhoana & Nel, 2020; Preece et al., 2015).

Of these strategies, explicit instruction (EI) – pioneered by Rosenshine and Stevens (1986) – is one of the set of promising strategies in helping students learn and understand (Guilmois et al., 2020) computer programming better. EI is described as 'a systematic method of teaching with emphasis on proceeding in small steps, checking for understanding, and achieving active and successful participation by all students' (Rosenshine, 1987: 34). The effectiveness of EI has been confirmed many times (Hughes et al., 2017; Rastle et al., 2021). The underlying principle of EI is that the transfer of knowledge is dealt with in a structured, systematic and planned way, and concepts are treated in the order from simple to complex and from easy to difficult. Any teaching strategy employing EI follows sequenced and strongly integrated steps. Literature suggests that in using EI, it is equally important to clarify the learning objectives and intended outcomes, identify key ideas, and determine students' prior knowledge (Guilmois et al., 2020). Greene (2022) summarises steps of EI as follows:

- Step 1 - Identify clear and specific objective(s)
- Step 2 - Break the information into chunks
- Step 3 - Model with clear explanations
- Step 4 - Verbalize the thinking process
- Step 5 - Provide opportunities to practice
- Step 6 - Give feedback.

All activities performed as part of EI aim to create some form of cognitive scaffolds for the student (Archer & Hughes, 2011), and to reduce working memory overload (Hughes et al., 2017). According to Bocquillon et al. (2020: 12), students taught using this strategy can learn 'without conscious effort' and 'without taking up the memory working space'. Tshukudu and Jensen (2020) found that EI interventions deepen understanding of programming concepts, hence concluding that these interventions are effective. Notwithstanding the effectiveness of EI, some studies have also reported on the following inherent limitations: encouraging students to sit passively and engage in rote learning (Hammond, 2019); encouraging students to over-rely on memorisation; encouraging students to lose interest (boredom); and limiting students creativity (Brainscape Academy, 2023; Iain, 2023). However, there seems to be little prior work investigating learning strengths and challenges presented by EI classroom interventions, specifically from the viewpoint of both students and the instructor. As such, this study aims to address the following research questions:

- What are the elements to consider in an instructional strategy for teaching using EI?
- What are the strengths and challenges presented by EI in teaching computer programming to postgraduate CS students (as experienced by both students and the instructor)?

In the remainder of this paper, the review of computer programming-related aspects and discussion of a conceptual framework guiding this study are presented in Section 2. The pedagogical intervention activities, research design and methods are presented in Section 3. The results and interpretation are presented in Section 4, followed by the discussion of findings in Section 5. The conclusion is presented in Section 6, followed by the limitations and recommendations for future research in Section 7.

LITERATURE REVIEW

This section is divided into two sub-sections that respectively provide the review of relevant literature related to computer programming aspects and the conceptual framework of the study.

Computer programming related aspects

In relation to the aspects of EI alluded to in the introduction, literature shows that when learning to program, students are highly likely to fail to learn when their memory is overloaded (Sweller et al., 2019). The use of explicit instruction, thus, helps to alleviate the impact of this failure as steps are given in small amounts (e.g., scaffolds) that are not overwhelming to students (Greene, 2022; Rosenshine & Stevens, 1995). Several studies have been conducted to show the influence that self-efficacy and motivation have on the success of programming students (Fang, 2012; Kovari & Katona, 2023; Korkmaz & Altun, 2014). In learning situations, students are likely to set high goals, choose more challenging tasks, and use constructive learning strategies when their self-efficacy is high (Rosenberg-Kima et al., 2022). As programming requires a lot of mental effort (Maalej et al., 2014), high interactivity (Rosenberg-Kima et al., 2022), and high level of abstraction (Gomes & Mendes, 2007), students are encouraged to apply exploratory programming for their effective learning. In this type of programming, students are encouraged to experiment with different possibilities in trying to understand concepts, develop plans to tackle problems, convert these plans into executable instructions, think and reason algorithmically, create/modify/implement code, evaluate the results, solve problems, etc. (Kery & Myers, 2017; Sheil, 1986).

In performing exploratory programming activities, students have to integrate a lot of aspects. For example, on investigating a whole-task instructional approach compared with a part-task instructional approach for students learning to program, Rosenberg-Kima et al. (2022) indicate that in programming, students should not only master the separate coding elements, but they should also master how to integrate such elements in solving a problem. Furthermore, in learning computer programming, mental models (e.g., frameworks that help students to understand how their minds work and why they think the way they do) are useful and enhance programming ability (Danao, 2022; Mayer, 1981). Moreover, in computer programming, students can be allowed to use learning styles that suit them, and this has been found to help students learn effectively (Gomes & Mendes, 2014; Kumar, 2017). Besides the learning styles, all concepts taught in the computer programming classroom should be as real as possible and relevant to the real world. This, in turn, helps students to develop skills necessary for the world of work (Cronjé & Brittz, 2005; Sentance & Csizmadia, 2017). In addition to the aspects discussed in this section, further literature is provided within the conceptual framework section of this study.

Conceptual framework

The framework guiding this study is based on the six steps of EI as presented in Section 1.

Step 1 - Identify a clear and specific objective

In planning a lesson, instructors will typically regard it as key to ensure that its purpose is clearly spelled out. They also ensure that they have looked into prior knowledge that students have acquired in the previous modules (Greene, 2022; Rosenshine & Stevens, 1995). This helps instructors ensure that the new knowledge students will acquire from the lesson, builds on the knowledge that students already possess (Barkley, 2010). Research in Computer Science (CS) cannot overemphasize the importance of prior computer programming knowledge in helping students to perform better in their studies (Iv et al., 2019; Veerasamy et al., 2018). According to Greene (2022), a clear and precise objective facilitates easy planning of the instructor`s EI interventions and unclear objectives may hinder the implementation of the subsequent steps of EI.

Step 2 - Break the information into chunks

Instructors are encouraged to holistically look at the content to convey to students, then break it down into small and meaningful segments that students can easily grasp and understand (Greene, 2023). These segments should also be presented separately (Archer & Hughes, 2011; Doabler et al., 2012), and sequentially to students (Guilmois et al., 2020; Rosenshine & Stevens, 1986). Inherently, students are likely to better recall information if organized in chunks (McKeithen et al., 1981). Furthermore, it is worth noting that when learning new content, human beings should only be stretched to an optimal level because their working memory capacity is limited (Constantinidis & Klingberg, 2016).

Step 3 - Model with clear explanations

Students learn better when they have examples to follow (Atkinson et al., 2000). Therefore, it is essential for instructors to explicitly explain/demonstrate a skill students should acquire in the same way that they will practise it. In demonstrating the concepts, instructors should try to be as natural and straightforward as possible so that it becomes easier for students to understand rather than for them to try to do the guesswork (Greene, 2023). Students could also be asked to model what was modelled to them earlier to confirm their understanding (Nilson, 2013). With close reference to Step 2 outcomes, instructors typically apply various modelling principles such as metaphors and physical demonstrations (LaRiviere, 2012; Middendorf, 2014). These principles help to explicitly model to students how to understand the learning content presented to them successfully. In applying these modelling techniques, the crucial skills are constantly highlighted (Middendorf, 2014).

Step 4 - Verbalize the thinking process

Instructors will typically perform a think-aloud of what is going on in their minds as they model explanations made and skills that need to be fostered with students (Middendorf & Pace, 2004; Pace, 2017). At this stage, instructors can also pose questions, recite affirmations and identify more valuable resources (Ellis, Denton & Bond, 2014). The thought process(es) of an instructor help(s) students to see how the thinking unveils as instructors might even have to refine their thinking at some stage (Greene, 2023). The questioning helps students engage and interact with the learning content and the instructor. During this engagement, instructors can identify students' level of understanding from the strengths and deficiencies portrayed by students. This helps instructors monitor the teaching and learning progress and reflect whether their current approach is working or not. Verbalizing the thinking process helps students learn and think in the same way they are taught. In this way, students can know how to begin with the task and what to do when they get stuck (Greene, 2023).

Step 5 - Provide opportunities to practise

According to Fenton (2015), practice in the learning process plays a vital role in student's acquisition of long-term knowledge and necessary skills. The element of practice may not be implanted in students if an instructor does not provide them with opportunities to practise independently (Lahtinen, Ala-Mutka & Järvinen, 2005). Practice opportunities can be presented as either guided or independent. During guided practice, an instructor might work together with the students through several problems and either pre-correct or correct errors as they occur. During independent practice, students are given a manageable problem to solve independently. This problem should align with the skills already modelled/verbalized. Instructors can re-model/verbalise the necessary skills if it becomes apparent from the given problem that students have still not mastered the skill (Fletcher et al., 2019). Inherently, the practice opportunities discussed above are forms of scaffolding, a concept defined as the 'process by which instructors provide students with cognitive supports early in their learning, and then gradually remove them as students develop greater mastery and sophistication' (Ambrose et al., 2010: 146). Various authors (Davis, 2014; Feyzi-Behnagh et al., 2014) have identified scaffolding as essential in ensuring that student learning occurs.

Step 6 - Give feedback

Providing immediate, continuous, relevant and actionable/descriptive feedback to students is a vital part of the learning process (Campos et al., 2012; Wlodkowski & Ginsberg, 2010). Feedback indicates and do not know relative to the required knowledge of the subject matter (Brookhart, 2008). Similarly, if an instructor does not return students' feedback in time and with constructive comments (Ertmer & Newby, 2013), that may negatively influence students' learning, hence affecting their ability to comprehend the necessary skills. Furthermore, students may practise such a skill with errors (Greene, 2023). There are several forms of feedback that computer programming students can receive. For example, most Integrated Development Environments (IDEs) provide feedback to programmers through everyday tools integrated into them, such as compilers, interpreters and run-time environments (Jerše & Lokar, 2018). Some authors argue that feedback should be provided immediately to students; otherwise it significantly slows down a student's progress (Hattie & Timperley, 2007; Jerše & Lokar, 2018). As such, systems for automated assessment of programming tasks are developed (Lokar & Pretnar, 2015). Some of these systems provide a range of feedback options from no feedback, simple verification feedback, correct response feedback, elaborated feedback and try again feedback (Vasilyeva et al., 2007). In other instances, instructors provide feedback while grading students' programs - focusing on program aspects such as functionality, design, programming style, syntax, semantics, use of best practices, efficiency, programming errors and software metrics (Ihantola, 2011; Koyya et al., 2013).

Conclusion

The aspects that can be considered in helping students better learn computer programming have been presented in Section 2.1. The discussion of the conceptual framework of this study provides answers to the first research question. These answers inform how the constituent components of the second research question can be empirically addressed.

METHODOLOGY

This section is divided into three sub-sections that respectively provide the pedagogical intervention activities of the study, the research design and methods, as well as how data were analysed.

Pedagogical intervention

Various pedagogical activities were carried out from inception throughout the intervention. Initially, all the instructional goals of the module for which this intervention was made were investigated. Due to time limitations, the research focused on the main objective (ability to design, code and implement mobile applications) when discussing three concepts: animation, Google maps, and SQLite Database. For the animation concept, all content was presented to students in a traditional face-to-face class using lecture slides and making the necessary demonstrations. For both the Google maps and SQLite Database concepts, audio-visual lectures were shared with students well before class, and summaries of the discussion of these concepts were presented to students in class and they asked questions where they needed help understanding. In discussing these three concepts, all the explicit steps necessary to achieve the learning goals were elaborated on and demonstrated to students.

To prepare the learning environment, all the necessary resources that students would need were made available to them. For instance, the installation of Android Studio and the essential software packages on computers in the venue that students used. This installation was also tested before students could start using it to ensure they would not experience problems in using the software. A module guide detailing all the meeting times and venues was also made available to students upfront, and all content in the module guide was discussed with the students.

An instructional strategy helping to achieve the selected instructional goal was also devised. This involved creating various learning components such as preparing further lecture notes; designing instructional activities that students would complete; documenting strategies to use to ensure that students would get engaged in the learning content presented to them (i.e., allowing students to practise a task and provide them with timely and descriptive feedback); as well as how students would be assessed on the skills acquired. In the main, students were kept motivated by praising and valuing their outstanding solutions when giving feedback.

Two assessments that tested whether students could master the instructional goal specified earlier were designed. In preparing the assessments, careful consideration was made to ensure that they specifically assessed the design, coding and implementation skills linked to various features of the three concepts specified earlier. The questions included in the assessments were straightforward and worded with correct punctuation and grammar. The lectures were presented in a computer lab where students had access to computers installed with the software.

Research design

Within the scope of an EI-based research design, an approach based on Plowright's Frameworks for an Integrated Methodology (FraIM) (Plowright, 2011) was followed. FraIM advocates that there is no philosophical position that needs to be taken before commencing the study. Such a position can, however, be taken as the study evolves or even with the interpretation of results. Thereby, the focus was on collecting narrative data using observations and asking questions. The study population consisted of postgraduate Honours CS students (referred to as postgraduate students in this paper) from a South African university. The study sample consisted of 14 students registered for an Honours Advanced Programming module. The reason for this sample size is that most of the time, it is normally a few students who enrol for the postgraduate studies in Computer Science at the selected university. The selection of this sample was both 'purposeful' and 'convenient' (Patton, 2015). The sample was purposeful because the instructor wanted to improve the teaching and learning strategies for the current and upcoming students for the module. The sample was also convenient since the instructor had easy access to the participants because a few minutes of the scheduled class sessions for students could be used to complete a one-minute paper (Angelo & Cross, 1993). Furthermore, observations throughout all activities carried out as part of the teaching and learning in the module were recorded. For the individual interviews at the end of the semester, students were invited to participate during their leisure time as it was not completed during the regular class sessions. According to Plowright (2011), a data collection strategy using one-minute papers and individual interviews can be regarded as a means of 'asking questions', and the strategy for making observations can be considered as a means of 'observations'.

Students were asked to complete a one-minute paper three times in the semester anonymously. These covered only the three concepts discussed, namely: Animation, Google maps, and SQLite Database. Nine students participated in the one-minute paper for both the Animation and Google Maps concepts, while 10 students participated in the SQLite Database concept. For each of these concepts on the one-minute papers, participants had to write down three key things they learned in the previous lecture, and share what they understood by the concept discussed. They were further asked to share what was most confusing with the concept discussed, and how helpful the relevant activity they did covering the concept was. The questions were adapted from various examples (Angelo & Cross, 1993; Suskie, 2018) of one-minute papers.

For the individual interviews, all the students were invited. However, only eight of them agreed to participate. In these interviews, students were asked to share their experiences with the EI interventions that were implemented in the classes for the selected module. Probing questions were asked where necessary. All the proceedings of the interviews were recorded after permission for audio recording was obtained from the corresponding participant. The interview sessions were each scheduled for 60 minutes.

Data analysis

To transcribe and analyze the audio recordings from the interviews, Creswell and Creswell's (2017) approach was followed. After transcribing the recorded interviews, the data was cleansed by searching for faults and repairing them accordingly (Chu et al., 2016). Since the questions were open-ended, the transcripts contained numerous illogical and repeated statements. As such, 'fuzzy validation' (instead of strict validation, which requires complete removal of invalid/undesired responses) was used (Parcell & Rafferty, 2017). In fuzzy validation, researchers are allowed to correct some data if there is a reasonably 'close match' to a known correct answer. Thereafter, familiarization with the data (Marshall & Rossman, 2016), by listening and re-listening to the audio records numerous times and by intensively and repeatedly reading the transcripts was done. This helped in devising a coding plan in which the analysis would be guided by the data related to the research questions. The eight validated transcripts were imported into the Nvivo tool at this stage. After that, codes were developed for each strength and challenge identified in the data. For coding, Klenke (2016) recommends the use of 'units of analysis' (e.g., words, sentences, or paragraphs). Accordingly, the data was coded by highlighting and/or underlining text (from which strengths and challenges of EI could be extracted) within the domain of the stated units of analysis. Then the created codes were populated by associating the corresponding texts with them. During this refinement process, the codes' names were continuously revised until relevant themes began to emerge. For each emerging theme, its Nvivo-generated frequency of occurrence was considered.

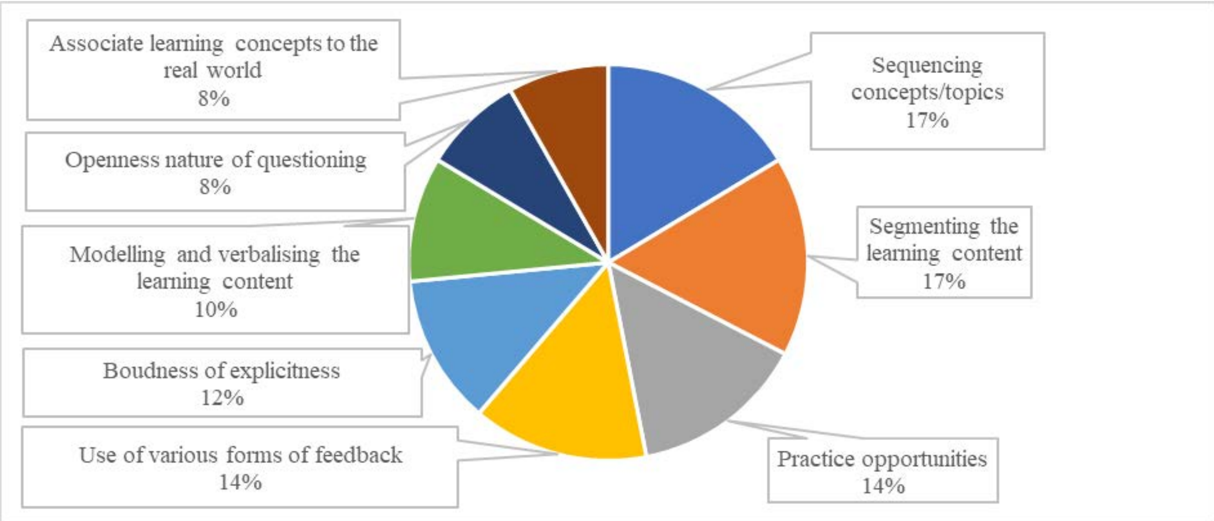
Results and interpretation

The discussion in this section addresses the rest of this study's research questions. The elements to consider in an instructional strategy for EI have already been presented in Section 2. The strengths and challenges identified are presented in the sub-sections that follow.

Strengths

This study's data revealed nine significant strengths related to the EI interventions. These are presented in Figure 1 below showing the percentage distribution of their occurrence from the data.

Figure 1:
A pie chart of strengths related to the EI interventions



Sequencing concepts/topics – This refers to presenting concepts/topics in a manner that makes logical sense, and concepts/topics offered at a later stage build on the ones discussed previously (Doabler et al., 2012; Rosenshine & Stevens, 1986). All the participants, with a total of 17 occurrences, indicated that they witnessed this sequencing (Guilmois et al., 2020) in the presentation of the learning content. P1 specifically remarked:

> There was never a point in the course where we were taught something, and we were told that we could just forget about it, and we'll learn more about it later. Everything was introduced, and it depended on something that we had already done in the past, except when we reached the point where we now needed to implement independent types of Google Maps functionalities.

The reason why P1 noted not necessarily seeing the sequencing in dealing with the Google Maps concept might have been that the concept typically requires students to integrate most of the skills learned with the other concepts (Rosenberg-Kima et al., 2022).

Segmenting the learning content — This refers to breaking down the learning content into small chunks for easy student consumption (Archer & Hughes, 2011; Greene, 2022). Students find this helpful for their better understanding of the learning content. All participants, with a total of 14 occurrences, provided evidence for this assertion. Pieces of evidence include:

> Yes, I feel like the learning content was broken down in a very nice way and given the fact that we only dealt with one concept at a time. (P1)

P2 remarked:

> I believe the [learning content] was given to us in pieces. So, we understood each component, of basically how to put Apps on your phone.

Moreover, P4 said:

> I think it [learning content] was given to me in manageable sizes because each week I'd have time from maybe Wednesday night until another Wednesday night, to [work on] an entire functionality ... I think the content was reasonably sized.

This also shows that students enjoyed the fact that the module was taught once a week, giving them an excellent chance to effectively engage with the learning content without feeling any pressure (Bocquillon et al., 2020).

Practice opportunities — These refer to any type of activity that students are given with the objective to practise, and not necessarily for grading. These activities could be completed before, during, and after the lecture. Greene (2022) emphasizes the importance of practice opportunities, and this is supported by the data collected which demonstrate the critical role of practice opportunities in the student learning process. Seven participants, with a total of 11 occurrences, provided evidence for this assertion. Typical examples include:

> For every concept, there was a practice opportunity (P1)

and

> The practicals and presentations, I found them very useful because they force you to engage. (P2)

On the question of practice opportunities, P8 wanted to confirm whether assignments also formed part of these opportunities by asking:

> Are the assignments part of those opportunities?

and upon receiving positive confirmation, he remarked:

> Yeah, it was beneficial, like for instance, the Calculator App will teach you how to create a method, link buttons, and display in the TextView.

The Calculator App was discussed in class, and since various ways of achieving the calculator functionalities exist, students were asked to re-develop the App using one of the other ways. The

Calculator App discussed in class was already an example helping students to further learn in re-developing it, reflecting the approach advocated by Atkinson et al. (2000).

Use of various forms of feedback – This involves using more than one form to provide beneficial feedback to students. In this intervention, two forms of feedback, written and verbal, were used. In the former, students received written comments, while in the latter, the instructor gave oral comments during the lecture in front of all the students. Seven participants, with a total of 11 occurrences, indicated that they benefited from using various forms of feedback (Fleming & Levie, 1993), but specifically from the verbal one. In addition to having identified various forms of feedback, P1 expressed how useful these were to him:

> The feedback was very useful, more so, given the fact that we received different types of feedback.

Participants also appreciated the principled and thoughtful way in which they received feedback. P4 remarked:

> It was the first time this year that an instructor gave feedback in class. Usually, we just receive an email to say, 'hey guys, your results are out' and that's it … it was a great strategy.

This was an overwhelming revelation because most of the instructors seem not to treat student feedback with the consideration it deserves. Peer learning (Porter et al., 2013) and networking, which are promising methods of effective learning, also surfaced in how feedback was given to students. P4 remarked:

> So even now 'I know [Student X] and [Student Y] because of you mentioning them in class', saying that [Student X] presented this so well … I was impressed about that … I started asking him questions.

This feedback-triggered a relationship that could last long-term among the students.

Use of various assessment techniques – This involves using more than one technique to comprehensively assess students. In this study, three assessment techniques namely, pure practical, recorded and verbal presentation, were used. In all the assessments, students were asked to read about a certain concept, and implement how it works. For the first technique, students submitted a practical solution for grading. For the second technique, instead of handing in a practical solution, students were asked to prepare a recorded presentation of their understanding of the concept and how they implemented it. Students submitted a comprehensive presentation for grading. For the third technique, students verbally presented their work in front of their classmates and were each graded throughout their presentation. Seven participants, with a total of 10 occurrences, indicated that they benefitted from the various assessment techniques used. However, all participants, with a total of 12 occurrences, quoted verbal presentation as the technique in which they learned the most. Examples include:

> When I was presenting something, I could tell that there was a level of understanding for how this thing works because you are forced to go also learn how this class works, what is it dependent on, to how it takes these data inputs. (P1)

P4 appeared like he did not prefer verbal presentations; however, he seemed to be enjoying its benefits as he remarked:

> So that's why I prefer being prompted to present in class because most of the things that I've presented, I know very well now … you also promote question-asking in class. So, when you present, there's an incentive for the students to ask you questions.

During the verbal presentations, students were encouraged to ask the presenter questions. The instructor informed them that anyone who asked would receive an extra point on their final mark for the presented assignment. From the observations made, feedback was better because of the promised reward.

The notion of peer learning (Porter et al., 2013) also surfaced in the verbal presentations, as P1 remarked:

> It gave us a chance also to get feedback from other people [fellow students].

This pivotal revelation indicates how verbal presentations stand out from the other two assessment techniques. P1 further emphasised this point:

> Everyone came out of that class understanding how the Google Maps class works.

This was the lecture in which students had to present their work in front of everyone in class. Highlighting what goes on in the preparation of the verbal presentation, P3 remarked:

> It challenges you to go beyond what was required

while P8 said:

> Presenting in front of your peers, you're forced to learn to work and also to answer questions from your peers as well. So basically, that puts my peers and I under pressure because now they have to understand the work to ask a question.

These excerpts show that students need more preparation for verbal presentations than for the other types of assessment. This preparation, consequently, helps students engage more with the learning content and gain more understanding.

Boundness of explicitness — This means that explicitness is only helpful up to a certain extent. Six participants, with a total of nine occurrences, provided evidence for this finding. P1 indicated that he would prefer the application of EI for passing and not necessarily for genuinely understanding the learning content:

> I would have loved that for marks, but it wouldn't have worked well for me gaining a skill and knowledge and learning how to use the different environments and everything.

On being emphatic about the boundness of EI, he further remarked:

> But obviously, we can't stick to explicit instructions forever, that we have established.

From the given examples, it can be deduced that scaffolding (Ambrose et al., 2010; Vygotsky, 1978) surfaces in the EI intervention and that students seem to learn best when they are guided up to a particular stage and are left to exercise their elementary skills to learn other advanced content by themselves.

Modelling and verbalising the learning content — Modelling is when instructors explicitly explain/demonstrate a skill that students should acquire in the same way that they will practise it. Verbalising is when instructors perform a think-aloud of what is going on in their minds as they model explanations made and skills that need to be fostered with students (Greene, 2022; Pace, 2017). These techniques were found to encourage student engagement and stimulated student learning from their implementation. Five participants, with a total of eight occurrences, indicated that they benefitted from these techniques. P2 stated:

> If the emulator doesn't work, you could explicitly see that he [the instructor] first tries to disconnect. Second, he tries to do this, then if that doesn't work, he does that.

Although he benefitted, P1 raised a vital aspect: towards the end of the semester, the responsibility was more on the students to integrate the techniques, especially modelling. He remarked:

> You [the instructor] did it in such a way that you had an error and intentionally so, so that we could also pick up the reason why this thing was happening this way … towards the end of the course, the modelling was difficult, because it felt as if the modelling responsibility was handed over to us.

Openness nature of questioning — This involves asking students, in assessments, to make their own choice on which functionalities they want to implement. This motivates and engages students in more profound and richer learning (Anderson, 2016). Four participants, with a total of six occurrences, provided evidence of this revelation. P4 expressly indicated that this type of questioning did not only help him be creative but kept him engaged with the learning content as well. He said:

> I don't have so many modules where I'm given that kind of leeway, where I can be creative, and then decide what I would like to do … so it kept me engaged.

According to the self-efficacy theory (Bandura, 1997), in such instances, students increase their self-motivation as well. This implies that they engage and learn more when they work on learning activities that they choose for themselves. This liberty of choice encourages students to learn using their preferred learning style(s) (Gomes & Mendes, 2014; Kumar, 2017).

Associate learning concepts to the real world — This means discussing the learning content in a manner that allows students see how what they are learning can be applied to the outside world. This can be achieved by using as many real-world examples as possible in teaching, and using artefacts that students can touch and play around with. This, in turn, fosters student engagement and learning content retention thereof (Ambrose et al., 2010). Four participants, with a total of five occurrences, supported this finding. P2 remarked:

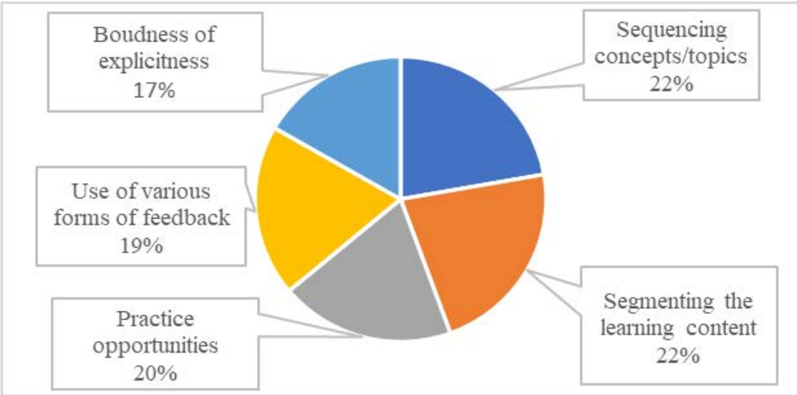
> It would also be nice to, maybe, have practical Apps that we can build, let's say, a Chat App.

As seen in the excerpt, P2 would be interested in learning content that relates to one of the contemporary issues that might be of interest or a challenge to him at the moment. This also suggests that instructors should try to strike a balance by using examples/artefacts that may not be far-fetched from the audience. For example, cartoons could be used for school children, while movies could be used for teenagers and upwards. When discussing the animation concept, the instructor designed and implemented a mobile application and told students that the concept is also used in making movies, and the instructor could see the excitement and curiosity that arose in them. This was emphasizing the concept of being relevant to the real world in a computer programming classroom as recommended by some authors (e.g., Cronjé & Brittz, 2005; Sentance & Csizmadia, 2017).

Challenges

Analysis of collected data revealed five main challenges related to the EI interventions. These are presented in Figure 2 below showing the percentage distribution of their occurrence from the data.

Figure 2:
A pie chart of challenges related to the EI interventions



Difficulty understanding theoretical concepts — To comprehend the practical implementation of any concept and how it works, it is also crucial to understand its theoretical foundations (Wrenn & Wrenn, 2009). The intervention module is based on both practical and theoretical aspects. As such, collected data revealed that theoretical concepts are not easy to understand before seeing their implementation and results. Six participants, with a total of seven occurrences, provided evidence of this revelation. P5 indicated that:

> It was challenging to try and conceptualise how and where I will actually use animations in my future Android Studio endeavours.

However, after the instructor mentioned that animation is normally used in making movies, the participant became more curious as reference to movies sounded a contemporary issue and was encouraged to explore more on the concept. P5 was actually applying the 'exploratory programming' concept where programmers are encouraged to experiment with different possibilities in trying to further understand concepts and in solving problems (Kery & Myers, 2017; Sheil, 1986). This implies that more prospects of learning occurred with the student.

Limited assessment techniques — As the use of various assessment techniques is an strength (see Section 4.1), it becomes a problem when only one assessment technique is used, a case in point being pure practicals that are handed in for grading. Four participants, with a total of seven occurrences, indicated this challenge. P1 was specifically concerned about whether the instructor can read a student's thought process (Archer & Hughes, 2011) through grading a practical assignment. He remarked:

> I only offer you the solution that I have; **you don't know the thought process that I was having when I was doing this** … and I might have also done something that **I also didn't understand**.

This is a genuine concern primarily because students usually work in collaboration, and copying each other's work is inevitable. One common pedagogical strategy in computer programming called pair programming (Porter et al., 2013; Williams et al., 2000) encourages this type of collaboration. Therefore, this suggests that using more than one assessment technique could be recommended as a viable solution.

Explicit steps may be tedious — Although the instructor has to strike a balance between using explicit and implicit steps in delivering the learning content to students, he/she also has to ensure that some students are not bored by the explicit steps (Rosenshine & Stevens, 1986) if they are already mastering concept(s) under discussion. Five participants, with a total of five occurrences, raised this challenge. In this respect, P7 remarked:

> When is it the time for you [the instructor] to start introducing implicit instruction?... In this case, I feel like the timing was perfect because we had learned all the necessary skills.

It, usually, is not difficult for the instructor to determine when explicit and implicit instruction should be used. This is normally done when students have developed the mental models (Danao, 2022; Mayer, 1981), and at this stage the instructor can observe student behaviours such as being bored by the explicit instructions and react accordingly going forward.

Integration of various concepts — This entails taking most of the skills already learned and putting them together to solve a given problem. Collected data revealed that this integration is not easy for students. Four participants, with a total of five occurrences, provided evidence to this revelation. P1 remarked:

> What complicated things for me was, if I saw a feature being [implemented] in a particular way, step-by-step and I saw another feature being [implemented] in a particular way, step-by-step, me coming to integrate all those features into one thing, which was something that is not presented in the step-by-step. It usually created problems regarding how I adapt that and that together.

This implies that the student did not have a problem understanding individual concepts when they were discussed, but the problem came at the time of integration (Rosenberg-Kima et al., 2022). However, P1 would use his intuition to integrate all concepts together if he had understood the individual concepts (Rosenberg-Kima et al., 2022). He said:

> It was imperative for us to get the explicit step-by-step instructions for implementing just the basic things from the get-go. That served a significant role in terms of us learning how to have intuition with using different tools that are handed to us later on to learn the concept.

Limited learning — The data revealed that EI can potentially limit students' learning (Brainscape Academy, 2023; Hammond, 2019; Iain, 2023). Four participants, with a total of four occurrences, provided evidence to this finding. P1 regards EI as 'spoon feeding', which may create problems when it is taken away:

> Being explicit only means that 'I will learn only what you are telling me', the step-by-step details. But if I have to shift to an environment where I've never seen a concept being implemented, it would create problems for me.

DISCUSSION OF FINDINGS

This study has revealed several important findings in relation to applying the principles of EI in teaching. Overall, it is key for instructors to apply these principles in their teaching, both at undergraduate and post-graduate levels. At postgraduate level, the EI principles are specifically relevant for modules that are as dynamic as Mobile Development as the way of doing things changes all the time. For example, Google can release four versions of the Android Operating System in one year (Google Developers, 2023). This means that students would have to learn using two or more versions in a semester. A typical case is where most of the time the latest versions or Application Programming Interfaces (APIs) are not compatible with the old functions (a concept called deprecation). Furthermore, for any type of teaching intervention in computer programming, students should learn the concepts as individual elements, after which they should also be taught how to put those elements together in solving various programming problems (Rosenberg-Kima et al., 2022).

It was also key to realise the small things that the instructor did with the students as part of the interventions that ultimately triggered the motivation of students and encouraged them to engage and learn. For example, asking students to do the oral presentations and encouraging other classmates to ask questions triggered considerable interactivity (Rosenberg-Kima et al., 2022) within the students, and they were able to learn a lot. Apart from the data findings, even the instructor observed an overwhelming interactivity and excitement during this activity such that the allotted time proved not enough. Furthermore, by virtue of the instructor making reference to other students in giving feedback, some students were able to create relationships that helped them to engage, work together and achieve in the selected module. It was also key to realise that students would prefer to be given feedback (Greene, 2023) in a manner that is principled and thoughtful. Inherently, this practice is not seen with many instructors. As such, it would be recommended that they re-consider the manner in which they give feedback to their students. Moreover, it was key to note the cognitive aspect of the learning process (Ertmer & Newby, 2013) surfacing from the students that, it may not be easy for instructors to decipher the thought process of the students from the work that gets graded. However, if students were to discuss their work with the instructor, more insights into the thought processes of students could be better understood.

There is other several aspects that instructors should observe throughout the process so that the EI interventions will not have an influence that could make the students, not only to develop negative attitude toward the interventions, but the module content as well. From the observational perspective, the instructor noted that students learn more if they are given problems to tackle, and ways of solving such problems are discussed in a class setting where the instructor and students are able to engage openly. Furthermore, students may not be aware of some learning techniques applied in their teaching; hence instructors are encouraged to discuss such techniques with students. Students may also not figure out additional resources by themselves; hence instructors are encouraged not to '*assume the obvious*', instead, should inform students of such resources (Ellis et al., 2014) if they are available. The instructor further observed that student learning difficulties might go unnoticed if the EI steps are not followed in the teaching process. It was additionally overwhelming to see elements such as excitement, engagement, interest and curiosity that students displayed when presenting verbally in class. Moreover, the EI steps look common and straightforward, however, instructors are encouraged not to overlook the simple activities within the main steps. Lastly, notwithstanding the benefits of EI, activities inherent within this intervention are time-consuming from the perspective of both students and the instructor.

CONCLUSION

Computer programming continues to be a challenge to CS students. Employing EI can be crucial in helping students to overcome related challenges. By focusing on the six steps of EI, this study aimed to identify elements to consider in an instructional strategy for teaching using EI principles, as well as strengths and challenges presented by EI interventions in teaching computer programming to postgraduate CS students as experienced by both students and the instructor. The literature review revealed key elements that need to be considered in an instructional strategy that uses the EI principles. These range from identifying clear and specific objectives of the module throughout the other steps of EI to giving feedback to students (Greene, 2023). Thematic analysis of collected data revealed nine major strengths and five main challenges related to the EI interventions. From the identified strengths and challenges, instructors can make improvements that could help students better grasp the subject's learning content. Through this paper, improvements on how to enhance the quality of the future teaching of computer programming have also been identified.

Limitations and future research

Some limitations of this study should be noted. The study was conducted within a specific context (a selected South African university) for students enrolled for a specific module. The study was also focused on identifying the strengths and challenges experienced by a very specific population (senior CS students). Due to the exploratory nature of this research study, no claims can therefore be made to the generalisability of the study findings. Natural extensions of this work could be conducted in a different setting, perhaps with participants from another CS module (undergraduate or post-graduate), and with a different instructor. The objective would be to ascertain whether there would be variations in the findings when a different group of students is used as participants, and a different instructor is the one implementing the interventions. Moreover, another limitation was the small sample as this type of intervention may be challenging to implement with a large class as it involves a lot of activities that, in turn, may attract considerable administration in which other instructors might not want to engage.

REFERENCES

Ambrose, A.S., Bridges, W.M., DiPietro, M., Lovett, C.M., & Norman, K.M. (2010) *How learning works : Seven research-based principles for smart teaching*. San Francisco: Jossey-Bass Publishers. https://doi.org/10.1002/mop.21454

Anderson, M. (2016) *Learning to Choose, Choosing to Learn: The Key to Student Motivation and Achievement*. Alexandria, VA USA: ASCD.

Angelo, T. & Cross, P. (1993) *Classroom Assessment Techniques*. San Francisco: Jossey-Bass Publishers.

Archer, A. & Hughes, C. (2011) *Explicit instruction: Effective and efficient teaching*. New York: Guilford Press.

Atkinson, R.K., Derry, S.J., Renkl, A. & Wortham, D. (2000) Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70(2) pp.181-214. https://doi.org/10.3102/00346543070002181

Bandura, A. (1997) *Self-efficacy: The exercise of control*. New York: W.H. Freeman.

Barkley, E. F. (2010) *Student engagement techniques : A handbook of college faculty*. California: Jossey-Bass Publishers.

Beth Kery, M., & Myers, B.A. (2017) Exploring exploratory programming. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, *2017-Octob*, 25-29. https://doi.org/10.1109/VLHCC.2017.8103446

Bocquillon, M., Gauthier, C., Bissonnette, S. & Derobertmasure, A. (2020) Enseignement explicite et développement decompétences : antinomie ou nécessité? *Formation et Profession* 28(2) pp.3-18. https://doi.org/10.18162/fp.2020.513

Brainscape Academy. (2023) *The pros and cons of explicit grammar instruction when learning a language*. https://www.brainscape.com/academy/pros-cons-explicit-grammar-instruction/ (Accessed 04 April 2023).

Brookhart, S.M. (2008) *How to give effective feedback to your students*. Alexandria: Association for Supervision and Curriculum Development.

Campos, D.S., Mendes, A.J., Marcelino, M.J., Ferreira, D.J. & Alves, L.M. (2012) A multinational case study on using diverse feedback types applied to introductory programming learning. *Proceedings - Frontiers in Education Conference, FIE*. https://doi.org/10.1109/FIE.2012.6462412

Chu, X., Ilyas, I.F., Krishnan, S. & Wang, J. (2016) Data cleaning: Overview and emerging challenges. *Proceedings of the 2016 International Conference on Management of Data*, 2201–2206. https://doi.org/10.1145/2882903.2912574

Constantinidis, C. & Klingberg, T. (2016) The neuroscience of working memory capacity and training. *Nature Reviews Neuroscience* 17(7) pp.438-449. https://doi.org/10.1038/nrn.2016.43

Creswell, J.W. & Creswell, J.D. (2017) *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). Thousand Oaks: Sage Publishers.

Cronjé, J.C. & Brittz, B. (2005) Programming in the real world. *Education as Change* 9(2) pp.131-161. https://doi.org/10.1080/16823200509487120

Danao, M. (2022) *SitePoint*. 10 Mental Models Developers Can Use to Get Unstuck. https://www.sitepoint.com/10-mental-models-developers-can-use-to-get-unstuck/ (Accessed 10 April 2023).

Davis, E.A. (2014) Scaffolding Learning. In R. Gunstone (Ed.) *Encyclopedia of Science Education*. Dordrecht, Netherlands: Springer.

Doabler, C.T., Cary, M.S., Jungjohann, K., Clarke, B., Fien, H., Baker, S., Smolkowski, K. & Chard, D. (2012) Enhancing Core Mathematics Instruction for Students at Risk for Mathematics Disabilities. *TEACHING Exceptional Children* 44(4) pp.48-57. https://doi.org/10.1177/004005991204400405

Ellis, A.K., Denton, D.W. & Bond, J.B. (2014) An analysis of research on metacognitive teaching strategies. *Procedia - Social and Behavioral Sciences* 116(2014) pp.4015-4024. https://doi.org/10.1016/j.sbspro.2014.01.883

Ertmer, P.A. & Newby, T.J. (2013) Behaviorism, Cognitivism, Constructivism: Comparing Critical Features from an Instructional Design Perspective. *Performance Improvement Quarterly* 6(4) pp.50-72. https://doi.org/10.1002/piq

Fang, X. (2012) Application of the participatory method to the computer fundamentals course. *Advances in Intelligent and Soft Computing 137 AISC* pp.185-189. https://doi.org/10.1007/978-3-642-27866-2_23

Fenton, N. (2015) *Using the "Top 20 Principles."* Psychology Teacher Network. http://www.apa.org/ed/precollege/ptn/2015/09/top-20-principles.aspx (Accessed 13 December 2022)

Feyzi-Behnagh, R., Azevedo, R., Legowski, E., Reitmeyer, K., Tseytlin, E. & Crowley, R.S. (2014) Metacognitive scaffolds improve self-judgments of accuracy in a medical intelligent tutoring system. *Instructional Science* 42(2) pp.159-181. https://doi.org/10.1007/s11251-013-9275-4

Fitzgerald, S., Simon, B. & Thomas, L. (2005) Strategies that students use to trace code: An analysis based in grounded theory. *Proceedings of the First International Workshop on Computing Education Research* pp.69-80. https://doi.org/10.1145/1089786.1089793

Fleming, M. & Levie, W. (1993) *Instructional Message Design: Principles from the Behavioral and Cognitive Sciences*. Englewood Cliffs, NJ, USA: Educational Technology Publications.

Fletcher, J., Lyon, G.R., Fuchs, L. & Barnes, M.A. (2019) *Learning disabilities: from identification to intervention*. New York: The Guilford Press.

Gomes, A. & Mendes, A. (2007) Learning to program - difficulties and solutions. *International Conference on Engineering Education*. http://ineer.org/Events/ICEE2007/papers/411.pdf (Accessed 12 November 2022)

Gomes, Anabela, & Mendes, A. (2014) A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. *Proceedings - Frontiers in Education Conference, FIE*.

https://doi.org/10.1109/FIE.2014.7044086

Google Developers. (2023) *Documentation*. Build.Version_Codes. https://developer.android.com/reference/android/os/Build.Version_Codes (Accessed 10 April 2023).

Greene, K. (2023) *Understood*. How to Teach Using Explicit Instruction. https://www.understood.org/articles/en/how-to-teach-using-explicit-instruction (Accessed 24 May 2022).

Griffin, J. (2015) Worked Examples with Errors for Computer Science Education. *ICER '15: Proceedings of the Eleventh Annual International Conference on International Computing Education Research* pp.261-262. https://doi.org/10.1145/2787622.2787741

Guilmois, C., Popa-Roch, M., Clément, C., Bissonnette, S. & Troadec, B. (2020) Effective numeracy educational interventions for students from disadvantaged social background: a comparison of two teaching methods. *Educational Research and Evaluation* 25(7-8) pp.336–356. https://doi.org/10.1080/13803611.2020.1830119

Hammond, L. (2019) *Explainer: what is explicit instruction and how does it help children learn?* https://theconversation.com/explainer-what-is-explicit-instruction-and-how-does-it-help-children-learn-115144 (Accessed 06 April 2023).

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

Hughes, C.A., Morris, J.R., Therrien, W.J. & Benson, S.K. (2017) Explicit Instruction: Historical and Contemporary Contexts. *Learning Disabilities Research and Practice* 32(3) pp.140-148. https://doi.org/10.1111/ldrp.12142

Iain, M. (2023) *ESL Lesson Handouts*. Some Advantages and Disadvantages of Explicit Grammar Instruction in EFL. https://www.esllessonhandouts.com/some-advantages-and-disadvantages-of-explicit-grammar-instruction-in-efl/ (Accessed 07 April 2023).

Ihantola, P. (2011) *Automated Assessment of Programming Assignments: Visual Feedback, Assignment Mobility, and Assessment of Students' Testing Skills* [Aalto University]. http://lib.tkk.fi/Diss/2011/isbn9789526043982/isbn9789526043982.pdf (Accessed 19 December 2022).

Iv, D.H.S., Jagodzinski, F., Hao, Q., Liu, Y. & Gupta, V. (2019) Quantifying the effects of prior knowledge in entry-level programming courses. *Proceedings of the ACM Conference on Global Computing Education* pp.30-36. https://doi.org/10.1145/3300115.3309503

Jerše, G. & Lokar, M. (2018) Providing better feedback for students solving programming tasks using project tomo. *1st Workshop on Innovative Software Engineering Education* pp.28-31.

Khomokhoana, P.J. & Nel, L. (2020) Decoding the underlying cognitive processes and related support strategies utilised by expert instructors during source code comprehension. *South African Computer Journal* 32(2) pp.232-257. http://dx.doi.org/10.18489/sacj.v32i2.811

Klenke, K. (2016) *Qualitative Research in the Study of Leadership*. Bingley: Emerald Group Publishing Limited.

Ko, A.J. & Uttl, B. (2003) Individual differences in program comprehension strategies in unfamiliar programming systems. *Proceedings of the 11th IEEE International Workshop on Program Comprehension*. https://doi.org/10.1109/WPC.2003.1199201

Ko, Andrew J., Ko, E.A., LaToza, T.D., Kwok, W., Hull, S. et al. (2019) Teaching explicit programming strategies to adolescents. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* pp.469-475. https://doi.org/https://doi.org/10.1145/3287324.3287371

Korkmaz, Ö. & Altun, H. (2014) Adapting Computer Programming Self-Efficacy Scale and Engineering Students' Self-Efficacy Perceptions. *Participatory Educational Research* 1(1) pp.20-31. https://doi.org/10.17275/per.14.02.1.1

Kovari, A. & Katona, J. (2023) Effect of software development course on programming self-efficacy. *Education and Information Technologies 0123456789*. https://doi.org/10.1007/s10639-023-11617-8

Koyya, P., Lee, Y. & Yang, J. (2013) Feedback for Programming Assignments Using Software-Metrics and Reference Code. *ISRN Software Engineering* 2013(2) pp.1-8. https://doi.org/10.1155/2013/805963

Kumar, A.N. (2017) Learning styles of computer science 1 students. *Proceedings - Frontiers in Education Conference, FIE*, *2017-Octob* pp.1-6. https://doi.org/10.1109/FIE.2017.8190464

Lahtinen, E., Ala-Mutka, K. & Järvinen, H. (2005) A study of the difficulties of novice programmers. *Proceedings of the 10th Annual SIGSCE Conference on Innovation and Technology in Computer Science Education* pp.14-18. https://doi.org/10.1145/1151954.1067453

LaRiviere, K.J. (2012) *Disciplinary Teaching: Exploring the "Decoding the Disciplines" Model*. https://www.youtube.com/watch?v=UKxnpNrNY_4 (Accessed 11 May 2023).

LaToza, T.D., Arab, M., Loksa, D. & Ko, A.J. (2020) Explicit programming strategies. *Empirical Software Engineering* 25(4) pp.2416-2449. https://doi.org/10.1007/s10664-020-09810-1

Lokar, M. & Pretnar, M. (2015) A low overhead automated service for teaching programming. *Proceedings of the 15th Koli Calling Conference on Computing Education Research*. https://doi.org/https://doi.org/10.1145/2828959.2828964

Maalej, W., Tiarks, R., Roehm, T. & Koschke, R. (2014) On the comprehension of program comprehension. *ACM Transactions on Software Engineering Methodology* 23(4) pp.1-37. https://doi.org/10.1145/2622669

Marshall, C. & Rossman, G.B. (2016) *Designing qualitative research* (6th ed.). Thousand Oaks : Sage Publications, Inc.

Mayer, R.E. (1981) The Psychology of How Novices Learn Computer Programming. *ACM Computing Surveys (CSUR)* 13(1) pp.121-141. https://doi.org/10.1145/356835.356841

McKeithen, K.B., Reitman, J.S., Rueter, H.H. & Hirtle, S.C. (1981) Knowledge organization and skill differences in computer programmers. *Cognitive Psychology* 13(3) pp.307-325. https://doi.org/10.1016/0010-0285(81)90012-8

Middendorf, J. (2014) *Decoding the Disciplines*. https://citl.indiana.edu/files/pdf/Middendorf Joan_SOTL bibliography.pdf (Accessed 29 August 2022).

Middendorf, J.K. & Pace, D. (2004) Decoding the disciplines: A model for helping students learn disciplinary ways of thinking. *New Directions for Teaching and Learning* 2004(98) pp.1-12. https://doi.org/10.1002/tl.142

Nilson, L.B. (2013) *Creating self-regulated learners: Strategies to strengthen student's self-awareness and learning skills*. Sterling: Stylus.

O'Dell, D.H. (2017) The debugging mindset: Understanding the psychology of learning strategies leads to effective problem-solving skills. *ACM Queue* 15(1) pp.1-20. https://doi.org/10.1145/3055301.3068754

Pace, D. (2017) *The decoding the disciplines paradigm: Seven steps to increased student learning*. Indiana University Press.

Parcell, E.S. & Rafferty, K.A. (2017) Interviews, recording and transcribing. In M. Allen (Ed.) *The SAGE Encyclopedia of Communication Research Methods*. Sage Publications, Inc. https://doi.org/10.4135/9781483381411.n275

Patton, M.Q. (2015) *Qualitative research & evaluation methods: Integrating theory and practice* (4th ed.). Bloomington: Sage Publications.

Plowright, D. (2011) *Using mixed methods: Frameworks for an integrated methodology*. Sage Publications.

Porter, L., Garcia, S., Glick, J., Matusiewicz, A. & Taylor, C. (2013) Peer Instruction in Computer Science at Small Liberal Arts Colleges. *ITiCSE '13: Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education* pp.129-134.

Porter, Leo, Guzdial, M., McDowell, C. & Simon, B. (2013) Success in introductory programming. *Communications of the ACM* 56(8) pp.34-36. https://doi.org/10.1145/2492007.2492020

Preece, J., Rogers, Y. & Sharp, H. (2015) *Interaction design: Beyond human-computer interaction* (4th ed.). New Delhi: John Wiley & Sons, Inc.

Qian, Y. & Lehman, J. (2017) Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Trans. Comput. Educ.* 18(1) 1:1--1:24. https://doi.org/10.1145/3077618

Rastle, K., Lally, C., Davis, M.H. & Taylor, J.S.H. (2021) The Dramatic Impact of Explicit Instruction on Learning to Read in a New Writing System. *Psychological Science* 32(4) pp.471-484. https://doi.org/10.1177/0956797620968790

Rosenberg-Kima, R.B., Merrill, M.D., Baylor, A.L. & Johnson, T.E. (2022) Explicit instruction in the context of whole-tasks: the effectiveness of the task-centered instructional strategy in computer science education. *Educational Technology Research and Development* 70(5) pp.1627-1655. https://doi.org/10.1007/s11423-022-10143-7

Rosenshine, B. (1987) Explicit teaching and teacher training. *Journal of Teacher Education* 38(3) pp.34-36.

Rosenshine, B. & Stevens, R. (1986) Teaching functions. In M.C. Wittrock (Ed.) *Handbook of research on teaching* (3rd ed.) New York: Macmillan Publishers, pp.376-391.

Rosenshine, B. & Stevens, R. (1995) Functions for teaching well-structured tasks. *Journal of Educational Research* 88 pp.262-268.

Sentance, S. & Csizmadia, A. (2017) Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies* 22 pp.469-495. https://doi.org/10.1007/s10639-016-9482-0

Sheil, B. (1986) *Power tools for programmers*. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Suskie, L. (2018) *Assessing Student Learning: A Common Sense Guide* (3rd ed.). San Francisco, California : Jossey-Bass Publishers.

Sweller, J., Ayres, P. & Kalyuga, S. (2011) *Cognitive Load Theory*. Heidelberg: Springer.

Sweller, John, van Merriënboer, J.J.G. & Paas, F. (2019) Cognitive Architecture and Instructional Design: 20 Years Later. *Educational Psychology Review* 31(2) pp.261-292. https://doi.org/10.1007/s10648-019-09465-5

Tshukudu, E. & Jensen, S.A.M. (2020) The role of explicit instruction on students learning their second programming language. *UKICER '20: United Kingdom & Ireland Computing Education Research Conference* pp.10-16. https://doi.org/10.1145/3416465.3416475

Vasilyeva, E., Puuronen, S., Pechenizkiy, M. & Räsänen, P. (2007) Feedback adaptation in web-based learning systems. *International Journal of Continuing Engineering Education and Life-Long Learning* 17(4–5) pp.337-357. https://doi.org/10.1504/IJCEELL.2007.015046

Veerasamy, A.K., D'Souza, D., Lindén, R. & Laakso, M.J. (2018) The impact of prior programming knowledge on lecture attendance and final exam. *Journal of Educational Computing Research*, 56(2) 226-253. https://doi.org/10.1177/0735633117707695

Von Mayrhauser, A. & Vans, A.M. (1996) Identification of dynamic comprehension processes during large scale maintenance. *IEEE Transactions on Software Engineering* 22(6) pp.424-437.

Vygotsky, L. (1978) *Mind in society*. Cambridge: Harvard University Press.

Williams, L., Kessler, R.R., Cunningham, W. & Jeffries, R. (2000) Strengthening the case for pair programming. *IEEE Software* 17(4) pp.19-25. https://doi.org/10.1109/52.854064

Wlodkowski, R.J. & Ginsberg, M.B. (2010) *Teaching intensive and accelerated courses: Instruction that motivates learning*. San Francisco, CA: Jossey-Bass Publishers.

Wood, D., Bruner, J. S. & Ross, G. (1976) The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry* 17(2) pp.89-100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Wrenn, J. & Wrenn, B. (2009) Enhancing Learning by Integrating Theory and Practice. *International Journal of Teaching and Learning in Higher Education* 21(2) pp.258-265. http://www.isetl.org/ijtlhe/

Xie, B., Nelson, G.L. & Ko, A.J. (2018) An explicit strategy to scaffold novice program tracing. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* pp.344-349. https://doi.org/10.1145/3159450.3159527