Technology-Based Strategies Predicated on Self-Regulated Learning in a Flipped Computer Programming Classroom

Simphiwe Dayimani and Keshnee Padayachee
University of South Africa, Pretoria, South Africa

56283415@mylife.unisa.ac.za
Padayk@unisa.ac.za

Abstract: The lack of self-regulation is believed to be at the centre of learning difficulties experienced by novice programming students. The Flipped Classroom Model (FCM) is a constructivist pedagogical strategy that could be used to engage students in a programming classroom. While the traditional approach to learning remains a passive environment, the FCM enhances active learning, problem solving while facilitating an unlimited access to the learning content. However, the success of the FCM depends extensively on a student's capability to self-regulate their learning process. As the FCM is a combination of face-to-face and online environments, Self-Regulated Learning (SRL) becomes more germane within the online dimension of the model. This study aims to identify the strategies for SRL that can improve the achievement of learners within a Flipped Programming classroom – specifically to reflect on the design features that should be considered for the adoption of technological tools to support SRL in a Flipped Programming classroom. This involved identifying relevant technological features to facilitate SRL strategies and the self-regulation phases leveraging a systematic review approach. The findings of this study could potentially maximise the student's self-regulation capacity to learn programming within an FCM.

Keywords: Self-Regulated Learning, Self-Regulated Learning Strategies, Flipped Classroom Model, Technology Tools, Programming, Flipped Programming Classroom.

1. Introduction

Learning to program is a complex endeavour for entry level students. These challenges arise from inadequate resources, inefficient teaching pedagogies and the lack of interest from students. Lack of self-regulation is believed to be at the centre of learning difficulties experienced by students in a programming classroom (Derus and Ali, 2012; Falkner, Vivian and Falkner, 2014). Self-regulated learning (SRL) is described as an individual's learning abilities to initiate plans, analyse the task at hand, set goals, assess their potential to accomplish these goals, develop learning strategies upon task implementation and reflect on performance (Zimmerman, 1990). Another possible strategy that could be used to engage students in a programming classroom is the Flipped Classroom Model (FCM) approach. The FCM is described as a form of blended learning that utilises technology to invert the traditional teaching and learning classroom roles and responsibilities (Strayer, 2012). The model may help students to engage with learning content and allow them more time for active and problem-based learning leveraging technology (Maher *et al,* 2015; Fetaji *et al,* 2016). As the FCM is a combination of face-to-face and online environments, SRL becomes more germane within the online dimension of the model (Çakıroğlu and Öztürk, 2017). In the FCM environment, students are expected to utilise their SRL skills to construct their knowledge through active participation in their learning process (Sletten, 2017). This study aims to discover the strategies for SRL that can improve the achievement of learners in a Flipped Programming classroom, specifically to reflect on the design features that should be considered for the adoption of technological tools to support SRL in a Flipped Programming classroom.

The SRL process is described as a significant activity required for students to be successful in learning how to program (Lai and Hwang, 2016). There is a positive correlation between students' application of SRL strategies and their performance (Alhazbi, 2014). The lack of self-regulation translates to deeper contextual problems such as poor coding skills and problem decomposition. Students are expected to regulate their cognition and metacognition strategies, motivation, behaviour and context through the utilisation of strategies that require proper planning, monitoring, control and reflection (Pintrich, 2000). Given the persistent challenges encountered by students to successfully regulate these areas, technology is proving to be a vital tool to assist, develop and improve students' self-regulation processes (Kumar *et al,* 2005). Clearly, the need for SRL tools is amplified within the FCM, negating the advantages of the modality for novice programmers.

Therefore, this study aims to review available SRL technological tools that may be harnessed within an FCM through a systematic analysis guided by the following research questions:

RQ1: What technology-based tools could be integrated into programming language courses to promote SRL?

RQ2: What design features should be considered for the adoption of a technological tool to support SRL in a Flipped Programming classroom?

The study reviews the specific areas of SLR strategies that can be supported through technology tools when learning programming. Furthermore, the study aims to provide potential design features that instructors need to consider when using technological tools within FCM to enhance SRL and coding capacity. The paper is structured as follows – Section 2 presents the background. Section 3 presents the theoretical framing regarding SRL. Section 4 describes the study methodology. Section 5 and Section 6 present the data analysis and discussion respectively. Limitations and delimitations are presented in Section 7. Finally, the paper concludes in Section 8.

2. Background

Bergin, Reilly and Traynor (2005) established the relationship between self-regulation and programming performance. They deduced that the effectiveness of self-regulation can function as a major determinant of student performance. Falkner, Vivian and Falkner (2014, p. 295) also examined the use of SRL strategies in software development courses and concluded that these strategies "guide the development of scaffolding activities to assist in learning the process of software development". The identified strategies are then aligned with the desired performance level with constant monitoring and are altered to sustain goal attainment (Panadero and Alonso-Tapia, 2014).

Çakıroğlu and Öztürk (2017) noted that students may possess some preferences to apply strategies depending on the learning environment. This was demonstrated in a flipped classroom setting where strategies like environmental restructuring were mainly executed at home to maximise comfort when accessing the learning content. Assistance was sought in the classroom environment when students encountered difficulties. However, the application of the learning content learned at home was a challenge to students as the use of monitoring strategies was ineffective at home (Çakıroğlu and Öztürk, 2017). Manso-Vázquez and Llamas-Nistal (2015) suggest students may struggle with monitoring in the absence of a teacher. The study proposed monitoring tools to visualise and quantify learning activities to develop SRL.

While using the FCM approach provides more learning time for programming learners, the issue of self-regulation is amplified in this context, and this is the basis of this research undertaking.

3. Theoretical Framing

SRL has been a subject of research to understand how students develop from novice to improved levels of learning. Panadero (2017) reports on the most popular SRL models that have a different view of SRL. Pintrich (2000) focuses on the students' specific self-regulation areas and phases (see Table 1). To achieve the desired learning goals, students should start planning, monitoring, controlling and reacting to the areas of regulation (Pintrich, 2000). Each phase has four areas of self-regulation – cognition, motivation, behaviour, and context. Although there is no clear visible structure on the order of execution of the SRL phases, Ye and Pennisi (2022) suggest that the model is naturally cyclical as it is an extension of the SRL model (Zimmerman, 2000).

Table 1: Self-regulation phases with areas of regulation (adapted from (Pintrich, 2000))

Phases	Areas of regulation			
	Cognition	Motivation/affect	Behaviour	Context
Forethought, planning and activation	Goal formation and plans to achieve goals with previously acquired knowledge.	Self-examination of capacity to complete a task and conduct complexity judgment including interest activation.	Time and effort planning and planning for self-observation of behaviour.	Perception of the task and context.
Monitoring	Metacognitive awareness and monitoring of cognition.	Awareness and exercising motivational control.	Self-observation of behaviour.	Monitoring changing task and context conditions
Control	Selection and adaptation of learning strategies for learning and thinking.	Selection and adaptation of learning strategies for managing motivation/affect.	Efforts altered based on how much of it is required, seek help and remain persistent.	Change or renegotiate a task. Change or leave context.
Reaction and activation	Examining performance to determine success or failure.	An affective reaction to one's performance.	Reflection on behaviour leading to the performance outcomes.	Evaluation of the task itself and the surrounding learning space.

Zimmerman and Martinez-Pons (1988) investigated and reported on general self-regulation strategies that students undertake for effective learning**.** These include self-evaluation, organising and transforming, goal setting and planning, seeking information, keeping records and monitoring, environmental restructuring, self-consequence, rehearsing and memorising, seeking social assistance, reviewing records, and others (Zimmerman and Martinez-Pons, 1988). These strategies proved to be highly correlated to student learning improvement. However, technology tools may be vital in assisting and supplementing these strategies (Kitsantas, 2013). These SRL strategies will be used as a framing for the systematic review as they pose a direct impact on learning performance including factors that necessitate self-regulation such as personal, metacognitive and environmental (Zimmerman and Martinez-Pons, 1988).

4. Research Methodology

This research adopts a systematic review technique suitable for investigating technological tools for learning programming as proposed by Kitchen and Charters (2007).

4.1 Research Design

First, the researcher considered various database sources that meet the inclusion criteria and acceptable quality standards. The eligibility criteria and quality check were used to determine whether the sources should be included in the final stages of analysis (Table 2).

Table 2: Overview of the Eligibility Criteria

CODE	Included papers	CODE	Excluded papers
IC1	Must have the functionality to support SRL.	EC1	Do not support SRL.
IC2	Must have the potential to be used by students learning programming.	EC2	Not usable by students learning programming.
IC3	One or more SRL strategies as outlined by Zimmerman and Martinez-Pons (1988) must be supported.	EC3	No clear self-regulation strategy is supported.
IC4	Must have the potential to improve student learning.	EC4	No improvement in student performance was reported.
IC5	Published from the year 2010 to 2022.	EC5	Prior to 2010.

4.2 Information and Database sources

The following databases were consulted – IEEE (Institute of Electrical and Electronics Engineers) Computer Society, Science Direct, ACM (Association for Computing Machinery) Digital Library and Scopus. The following search word combination was used: learning ("Programming" OR "coding") AND "self-regulated learning".

4.3 Risk of Bias

Hammersley (2020) suggests that systematic reviews are subject to publication bias, which is defined as the omission of smaller studies or negative results yielding studies. The methodology process was clearly outlined to ensure transparency and the repeatability of the study.

4.4 Ensuring research quality

The study quality may be achievable by the unbiased presentation of findings which form the basis for internal validity which refers to measures to avoid “systematic errors” External validity refers to the applicability of study effects outside the study (Kitchenham and Charters, 2007, p. 20).

The quality criteria to assess the study quality (adapted from (Kitchenham and Charters, 2007)) are as follows:

- How well does the evaluation address its original aims and purpose?
- How clear and coherent is the reporting?
- Are the results clarified with enough details? (Kmet et al, 2004)

4.5 Data Collection

The search was guided by the search phrases, the field of study and publication type to limit the search scope to retrieve the most relevant studies. Each study was analysed to identify the underlying self-regulatory design features and the effect on students learning. A spreadsheet was used to tabulate and dissect each study to determine the data to be collected. Ethical clearance for the study was granted by the University of South Africa (UNISA).

5. Data Analysis

A total of 819 papers were retrieved with duplicates removed from all the consulted data sources. Through the identification of study relevance and screening of each study title and abstract, the initial number was reduced to 58 studies. The remaining studies were further reviewed and analysed to determine correspondence to the eligibility criteria. These studies had to demonstrate evidence of improved student performance when SRL is applied to any programming concept. To that end, 36 studies were removed. The remaining eleven (n=11) studies were included in the review process. Figure 1 illustrates the systematic review procedure followed.

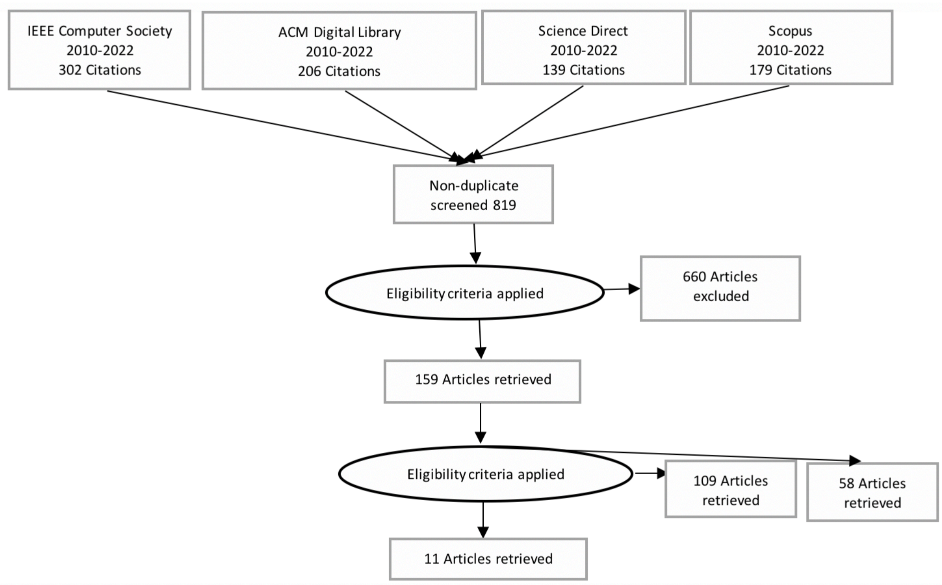

Figure 1: Preferred reporting items for systematic reviews and meta-analyses (PRISMA) diagram (Moher *et al.*, 2010).

Several studies from the literature were reviewed to identify technology-based SRL strategies. For example, SRL strategies improve student performance in learning programming (Huang *et al,* 2014; Lazar, Sadikov and Bratko, 2017; Keung *et al,* 2018). Some tools provide students with differentiated methods of problem-solving (Lazar, Sadikov and Bratko, 2017), permit code modification (Keung *et al,* 2018), track their code attempts (Daradoumis *et al,* 2021) and assess and provide students with feedback in varied forms for self-reflection (Breymann, Mader and Kok, 2020). Several tools were developed to promote SRL in learning to code, which are summarised in Table 4.

Table 4: Tool objectives towards sustaining SRL

Tool	Reference	Objective of Tool
PoliCAT	(Dascalu *et al,* 2017)	Assessment and feedback shortcomings are shared with other students on social media where other students can then comment and share their suggestions concerning learning goals including performance results.
SNAP	(Marwan *et al,* 2021)	Provides progress monitor through a checklist. Provides progress feedback in a dedicated interface.
Doubtfire++	(Law *et al,* 2017)	Students set learning goals. Manage learning. Track activity and learning progress. Document the learning process.

Tool	Reference	Objective of Tool
TA-HELP.ME	(Breymann, Mader and Kok, 2020)	Students seek help and select suitable questions for the required solution.
WPGA	(Hsiao, Huang and Murphy, 2020)	Provides for manual and automated task grading. Feedback and record keeping.
TA	(Yamada *et al,* 2019)	Students learn based on their preferred physical location. Students wear the device, and their movement is detected. Instructions are given based on a suitable location. Student movement determines the required learning content.
LET^{J}S	(Huang *et al,* 2014)	Students set their own learning goals. Self-paced activities using learning indicators for completed work and time tracking. Self-evaluation.
E-Journal	(Alhazbi, 2014)	Monitoring attitude and learning. Goal setting. Behaviour monitoring. Self-evaluation.
SRL-LMS	(Selvi and Panneerselvam, 2012)	Students assess their learning with feedback. Performance self-monitoring. Strategies for evaluation.
OLP	(Çakıroğlu and Öztürk, 2017)	Students reply to and ask questions to the instructor and peers. Provides online assessment. Record keeping in video and PDF format. Tracks the number of times a video watched.
DSLAB	(Daradoumis *et al,* 2021)	Provides feedback. Retrievable code records keeping.

By zooming into each tool's features and uses, we identify supported strategies and SRL phases and the possible relationship with the FCM performance. Table 5 demonstrates the design features with the corresponding SRL strategy deployment in the FCM and the context of the self-regulation.

Table 5: Design Features to support SRL Strategies

Strategies	Design Feature	Context of Self-Regulation
Self-evaluation	Enabled automatic self-evaluation. Utilise automated evaluation functionality on built-in or instructor designed assessments.	Monitoring changing task and context conditions.
Organising and transforming	Simplification of complex concepts (e.g., simple diagrams corresponding to complex code). Cater for modifiable learning space suitable for students with uncomplicated alternatives.	Change or renegotiate a task.
Goal setting and planning	Allow for planning and setting of the learning target. Attainable learning outcomes must be identified in advance including clear preparation on how to achieve them.	Perception and task context.
Seeking information	Simplified information search and easy navigation. Avail extra varying content to complement the available material.	Change or renegotiate a task.
Keeping records and monitoring	Keeping track of learned content and with respective outcomes. Provide means to retain content for use at a later stage (e.g., downloads and progress reports).	Monitoring changing task and context conditions.
Environmental restructuring	Suitable learning space or environment to maximise learning ability. Permit and arrange for the preferred learning environment.	Perception and task context.
Self-consequences	Decision-making on suitable consequences based on outcomes, where students will reward/punish themselves for passing/failing. This could be achieved thorough a performance self-monitoring tool.	Evaluation of the task itself and the surrounding learning space.
Rehearsing and memorizing	Provide the opportunity to retain and revise the already learned content. Enable unlimited review of previously completed work.	Change or renegotiate a task.

Strategies	Design Feature	Context of Self-Regulation
Seeking social assistance	Allow individuals to request and obtain assistance from others. Provide means for communication with others through specific channels such as messaging, email and social media.	Change or renegotiate a task.
Reviewing records	Learning content is available for review in multiple locations. Avail multiple complementary sources for thorough preparation.	Perception and task context.

6. Discussion

The identified tools were studied concerning the SRL strategies that may have a positive impact on learning programming. In response to *RQ1* (*What technology-based tools could be integrated into programming language courses to promote SRL?),* this paper considered how the design features of each tool could be effective in promoting the strategies used for SRL as proposed by Zimmerman (1989).

Firstly, the most common SRL strategy promoted by the tools analysed appears to be *Self-Evaluation* of one's learning. Students' self-evaluation is presented in varied forms through tools such as PoliCAT, WPGA, LET[J]S and OLP. These tools provide task quizzes and assess student code while providing targeted feedback. For the *Organising and Transforming* strategy, SRL-LMS presents learning content in the dashboard, and the information is organised in an easy-to-navigate way. These tools might regulate student cognition as programming requires proper analysis of the given problem (Garcia, Falkner and Vivian, 2018). Regarding the *Goal Setting and Planning* strategy, Doubtfire++ and SRL-LMS are tools used for goal-setting and the creation of study plans through built-in design features**.** LET[J]S and E-Journal permit students to set their own learning goals suitable for their study pace while they track time which in turn may provide the needed information to modify their behaviour.

Although the next strategy, *Seeking Information*, is associated with learning through the internet, SRL-LMS allow students to navigate the built-in material and learning content (Lee and Tsai, 2011). The system consists of information for learning online with unlimited availability allowing students to review the learning content when needed. Concerning *Keeping records and Monitoring*, students are expected to use the available resources to analyse the increase in their learning efficiency (Manso-Vázquez and Llamas-Nistal, 2015). LET[J]S, Doubtfire++, E-Journal, SNAP, SRL-LMS and DSLAB can help with keeping records and monitoring progress. E-Journal and SRL-LMS keep track of the reviewed content and the targeted learning goals to regulate behaviour on studied content or learning goals.

The *Environmental Restructuring* strategy, which is usually related to a physical learning space, may be acquired through system tools (Dembo and Seli, 2016; Garcia, Falkner and Vivian, 2018). For this strategy, TA and LET[J]S allow students to identify the most suitable learning environment. This may be suitable for context regulation during the forethought phase of learning as students identify suitable learning spaces (Dembo and Seli, 2016). TA-HELP.ME supports the *Seeking Social Assistance* strategy in order to enquire about or supplement learning content to attain learning goals and seek assistance from peers and students through social media. OLP allows students to seek help directly from other sources to regulate their cognition and behaviour when subjected to challenging programming concepts.

The *Reviewing Records* strategy was demonstrated by OLP which helps students to read their notes directly from the system without the need to consult external sources, while LET[J]S records students learning data for access and analysis at a later learning stage. Students may also use the E-Journal to track learning activities and content. However, while the *Rehearsing and memorising* strategy has a positive effect on learning programming, there was no emerging tool that supports students towards this strategy that was identified within the systematic review. In terms of the *Self-consequence* strategy, SRL-LMS is reported as a tool with potential to support this stratagem although there are few details explaining this function.

In response to *RQ2* (*What design features should be considered for the adoption of a technological tool to support SRL in a Flipped Programming classroom?*), this study highlighted some significant technology features that may be required for SRL that could be applied to the FCM. Design features are proposed (see Table 5) with corresponding strategy deployment and context in the FCM. The following linkages between SRL strategies and the design features and tools that could be used within a system to support FCM were drawn. The tools could involve online self-assessment quizzes, targeted feedback, transforming programming code into simpler formats (such as UML), online study plans, tracking of the code editing data, providing hints on code errors, e-journaling and defining a suitable learning environment. An important design feature is easy navigation of the system tools and content. Additionally social media tools could be useful in supporting students to seek assistance with content knowledge.

Kitsantas (2013) proposed tools concerning the SRL that includes regulation stages such as forethought, performance, and the self-reflection phase. Forethought phase tools may include tools such as web journals (i.e., blogs) and podcasts. These tools may be useful for self-reflection, self-monitoring, self-efficacy and modelling. Performance phase tools include social networks, virtual worlds, administrative tools and online marking tools. The functions and uses of these tools may include communication, collaboration, feedback, simulations, and records administration. The self-reflection phase tools include online grading systems and wikis which are useful for knowledge sharing, record keeping, feedback, collaborations, and discussions. These tools might be useful for self-evaluation, self-satisfaction, peer modelling and help-seeking (Kitsantas, 2013). Garcia, Falkner and Vivian (2018) who conducted a systematic analysis towards identifying computer science SRL strategies, found that self-evaluation emerged as the most supported strategy via automated assessment and feedback mechanisms as was found in this study. However, while the usage of these tools is targeted at the e-learning environment, their usage and application in the FCM remain inconclusive.

7. Limitations and Delimitations

This study is limited to programming self-regulation strategies. Specific programming languages were not taken into consideration; however, the general concepts are applicable to programming. The study is confined to SRL strategies and technological tools. The findings are limited to the databases selected.

8. Conclusion

This study sought to investigate the technology-based self-regulation strategies that students harness to improve their learning. Kumar et al. (2005) note that effective SRL strategies can be a predictor of student performance in programming. Husin, Judi, Hanawi and Amin (2020) argue for technology integration toward learning to program. Learning through technology simplifies learning and evokes the desire to learn (Husin *et al,* 2020). Educators may benefit from this overview of design features and tools to consider when adopting technology based SRL within FCM. Talbert (2014) suggests the following requirements for an effective FCM – open lines of communication, accountability where students take responsibility and demonstrate dedication and well-planned assessments to determine whether learning has taken place before, during and after contact learning sessions. These strategies for FCM should be integrated within the SRL strategies as well to ensure student success.

References

Alhazbi, S. (2014) "Using e-journaling to improve self-regulated learning in introductory computer programming course", in *2014 IEEE Global Engineering Education Conference*. Istanbul, Turkey 2-5 April: IEEE, pp. 352–356.

Bergin, S., Reilly, R. and Traynor, D. (2005) "Examining the role of self-regulated learning on introductory programming performance", in *Proceedings of the 2005 international workshop on Computing education research - ICER '05. the 2005 international workshop*, New York, United States,1-2 October: ACM Press, pp. 81–86. Available at: https://doi.org/10.1145/1089786.1089794.

Breymann, L.E.I., Mader, A.H. and Kok, H.M. (2020) "How can technology enhanced learning improve the efficiency and quality of help seeking and giving for programming tutorials?", in. *SEFI 48th Annual Conference Engaging Engineering Education, Proceedings*, Enschede, University of Twente, 20-24 September: Societe Europeenne pour la Formation des Ingenieurs, pp. 668–677.

Çakıroğlu, Ü. and Öztürk, M. (2017) "Flipped Classroom with Problem Based Activities: Exploring Self-regulated Learning in a Programming Language Course", *Educational Technology & Society*, 20, pp. 337–349.

Daradoumis, T., Marquès Puig, J.M., Arguedas, M. and Calvet Liñan, L. (2021) "A distributed systems laboratory that helps students accomplish their assignments through self-regulation of behavior", *Educational Technology Research and Development*, 69(2), pp. 1077–1099. Available at: https://doi.org/10.1007/s11423-021-09975-6.

Dascalu, M., Bodea, P., Melania, N., Alecu, G., Marin, I., Mitrea, D. and Ana Maria, N. (2017) *Supporting self-regulated learning via technology tools: a case study in an engineering degree program in Romania*, pp.5054- 5061.

Dembo, M.H. and Seli, H. (2016) *Motivation and Learning Strategies for College Success: A Focus on Self-Regulated Learning*. 5th edn. New York: Routledge.

Derus, S.R.M. and Ali, A.M. (2012) "Difficulties in learning programming: Views of students", in *1st International Conference on Current Issues in Education*. Yogyakarta, Indonesia, 15-16 September: Yogyakarta State University, pp. 74–79.

Falkner, K., Vivian, R. and Falkner, N.J.G. (2014) "Identifying computer science self-regulated learning strategies", in *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14*. Uppsala, Sweden, 21-25 June: Association for Computing Machinery, pp. 291–296.

Fetaji, M., Fetaji, B., Sukic, C., Gylcan, A. and Ebibi, M. (2016) "Case Study Analyses of the Impact of Flipped Learning in Teaching Programming Robots", *TEM Journal*, 5(4), pp. 401–406. Available at: https://doi.org/0.18421/TEM53-22.

Garcia, R., Falkner, K. and Vivian, R. (2018) "Systematic literature review: Self-Regulated Learning strategies using e-learning tools for Computer Science", *Computers & Education*, 123, pp. 150–163.

Hsiao, I.-H., Huang, P.-K. and Murphy, H. (2020) "Integrating Programming Learning Analytics Across Physical and Digital Space", *IEEE Transactions on Emerging Topics in Computing*, 8(1), pp. 206–217.

Huang, T.-C., Shu, Y., Chang, S.-H., Huang, Y.-Z., Lee, S.-L., Huang, Y.-M. and Liu, C.-H. (2014) "Developing a self-regulated oriented online programming teaching and learning system", in *2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering*. Wellington, New Zealand, 8 - 10 December: IEEE, pp. 115–120.

Husin, N.F., Judi, H.M., Hanawi, S.A. and Amin, H.M. (2020) "Technology integration to promote desire to learn programming in higher education", *International Journal on Advanced Science, Engineering and Information Technology*, 10(1), pp. 253–259.

Keung, J., Xiao, Y., Mi, Q. and Lee, V.C. (2018) "BlueJ-UML: Learning object-oriented programming paradigm using interactive programming environment", in *2018 International Symposium on Educational Technology (ISET)*. Osaka, Japan, 31 July - 2 August: IEEE, pp. 47–51.

Kitchenham, B. and Charters, S. (2007) "Guidelines for performing Systematic Literature Reviews in Software Engineering", *Technical Report EBSE 2007-001*, 2.

Kitsantas, A. (2013) "Fostering college students' selfregulated learning with learning technologies", *Hellenic Journal of Psychology*, 10, pp. 235–252.

Kumar, V., Winne, P., Hadwin, A., Nesbit, J., Jamieson-Noel, D., Calvert, T. and Samin, B. (2005) "Effects of self-regulated learning in programming", in *Fifth IEEE International Conference on Advanced Learning Technologies*. Kaohsiung, Taiwan, 5-8 July: IEEE, pp. 383–387.

Lai, C.-L. and Hwang, G.-J. (2016) "A self-regulated flipped classroom approach to improving students" learning performance in a mathematics course", *Computers & Education*, 100, pp. 126–140.

Law, C.-Y., Grundy, J., Cain, A., Vasa, R. and Cummaudo, A. (2017) "User Perceptions of Using an Open Learner Model Visualisation Tool for Facilitating Self-Regulated Learning", in *Proceedings of the Nineteenth Australasian Computing Education Conference*. New York, USA, 31 January - 3 February: Association for Computing Machinery, pp. 55–64.

Lazar, T., Sadikov, A. and Bratko, I. (2017) "Rewrite rules for debugging student programs in programming tutors", *IEEE Transactions on Learning Technologies*, 11(4), pp. 429–440.

Lee, S.W.-Y. and Tsai, C.-C. (2011) "Students' perceptions of collaboration, self-regulated learning, and information seeking in the context of Internet-based learning and traditional learning", *Computers in human behavior*, 27(2), pp. 905–914.

Maher, M.L., Latulipe, C., Lipford, H. and Rorrer, A. (2015) "Flipped Classroom Strategies for CS Education", in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. Kansas City, Missouri, 4-7 March: Association for Computing Machinery, pp. 218–223.

Manso-Vázquez, M. and Llamas-Nistal, M. (2015) "A monitoring system to ease self-regulated learning processes", *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 10(2), pp. 52–59.

Marwan, S., Shabrina, P., Milliken, A., Menezes, I., Catete, V., Price, T.W. and Barnes, T. (2021) "Promoting Students' Progress-Monitoring Behavior during Block-Based Programming", in. *Koli Calling '21: 21st Koli Calling International Conference on Computing Education Research*, Joensuu, Finland, 18-21 November: ACM, pp. 1–10.

Moher, D., Liberati, A., Tetzlaff, J. and Altman, D.G. (2010) "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement", *International Journal of Surgery*, 8(5), pp. 336–341. Available at: https://doi.org/10.1016/j.ijsu.2010.02.007.

Panadero, E. (2017) "A Review of Self-regulated Learning: Six Models and Four Directions for Research", *Frontiers in Psychology*, 8. Available at: https://doi.org/10.3389/fpsyg.2017.00422.

Panadero, E. and Alonso-Tapia, J. (2014) "How do students self-regulate? Review of Zimmerman's cyclical model of self-regulated learning", *Anales de Psicología*, 30, pp. 450–462.

Pintrich, P.R. (2000) "Chapter 14 - The Role of Goal Orientation in Self-Regulated Learning", in M. Boekaerts, P.R. Pintrich, and M. Zeidner (eds) *Handbook of Self-Regulation*. San Diego: Academic Press, pp. 451–502. Available at: https://doi.org/10.1016/B978-012109890-2/50043-3.

Selvi, S. and Panneerselvam, K. (2012) "A Self-regulated Learning approach for programming language using cloud-based Learning Management System", in *2012 International Conference on Recent Trends in Information Technology*. Madras, India, 19-21 April: IEEE, pp. 191–196. Available at: https://doi.org/10.1109/ICRTIT.2012.6206810.

Sletten, S.R. (2017) "Investigating Flipped Learning: Student Self-Regulated Learning, Perceptions, and Achievement in an Introductory Biology Course", *Journal of Science Education and Technology*, 26(3), pp. 347–358.

Strayer, J.F. (2012) "How learning in an inverted classroom influences cooperation, innovation and task orientation", *Learning Environments Research*, 15(2), pp. 171–193.

Yamada, M., Goda, Y., Imamura, R. and Egi, H. (2019) "The Effects of Seat Location-based Teaching Assistant Support System on the Awareness of Self-Regulated Learning and Learning Performance", in *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*. Yogyakarta, Indonesia,10 - 13 December: IEEE, pp. 625–631.

Ye, D. and Pennisi, S. (2022) "Using trace data to enhance Students' self-regulation: A learning analytics perspective", *The Internet and Higher Education*, 54, p. 100855. Available at: https://doi.org/10.1016/j.iheduc.2022.100855.

Zimmerman, B.J. (1989) "A social cognitive view of self-regulated academic learning.", *Journal of Educational Psychology*, 81(3), pp. 329–339. Available at: https://doi.org/10.1037/0022-0663.81.3.329.

Zimmerman, B.J. (1990) "Self-regulated learning and academic achievement: An overview", *Educational psychologist*, 25(1), pp. 3–17.

Zimmerman, B.J. (2000) "Chapter 2 - Attaining Self-Regulation: A Social Cognitive Perspective", in M. Boekaerts, P.R. Pintrich, and M. Zeidner (eds) Handbook of Self-Regulation. San Diego: Academic Press, pp. 13–39. San Diego, CA: Academic Press.

Zimmerman, B.J. and Martinez-Pons, M. (1988) "Construct validation of a strategy model of student self-regulated learning.", *Journal of educational psychology*, 80(3), p. 284.