Towards a Brain-Compatible Approach for Online Programming Education Through CHAT

[1]Vuyolwethu Mdunyelwa, [1]Lynn Futcher, [1,2]Johan van Niekerk
[1]Nelson Mandela University, Gqeberha, South Africa
[2]Noroff University College, Kristiansand, Norway

vuyolwethu.mdunyelwa@mandela.ac.za
lynn.Futcher@mandela.ac.za
johan.vanniekerk@noroff.no

Abstract: During the outbreak of the COVID-19 pandemic, the majority of higher education institutions were faced with the challenge of adopting various online platforms as a means for teaching and learning. This added an additional level of complexity in teaching modules such as programming, which are difficult to teach even in traditional classroom environments. Teaching and learning programming is difficult since the teaching does not only focus on programming, but should also show the students how to solve complex programming problems. Also, many programming students struggle to understand programming since they would typically not put their programming knowledge into practice outside of the learning environment. Online learning adds additional challenges, since the teacher is not available to provide immediate feedback which is one of the critical aspects in programming education. This paper reports on a case which considers brain-compatible principles when creating online programming content for learning programming. Furthermore, it reflects on the use of the Cultural-Historical Activity Theory (CHAT) to delineate the various elements required for teaching and learning programming.

Keywords: Brain-Compatible Principles, Online Programming Education, Cultural-Historical Activity Theory.

1. Introduction

Higher education institutions offering software development and other computer related courses have experienced an increase in growth relating to student numbers in recent years (Abesadze and Nozadze, 2020). Many students registered for these courses are required to study some form of programming, either at introductory level or throughout their qualification (Fawcett and Juliana, 2015). This is certainly the case in software development qualifications since programming is the main skill which students need to be equipped with (Espinha Gasiba, Lechner and Pinto-Albuquerque, 2020). While this is the case, students studying programming often face difficulties in learning, which is evident in the high failure rate experienced in programming modules, especially at introductory and second year level (Waite and Sentance, 2021).

This high failure rate was experienced prior to the adoption of online education, where lecturers were able to teach in their traditional face-to-face lecturing environments. Numerous educational tools provide an effective educational experience in an online environment. However, if these tools are unfamiliar to lecturers and students, they cannot be used to provide effective programming education (Waite and Sentance, 2021). This paper reports on how online programming education was implemented at a South African university to address the challenges relating to online learning.

Section 2 presents the related literature relating to programming education, with a specific focus on the online environment. In addition, it presents brain compatible principles that should be considered in teaching and learning programming. A discussion of the Cultural Historical Activity Theory (CHAT) follows as it relates to planning for teaching and learning. Sections 3 presents the research context and design, while Section 4 highlights the research approach undertaken. Section 5 reflects on the use of CHAT and brain-compatible principles to enhance online programming education and Section 6 concludes the paper.

2. Related Literature

Due to the recent advances in information technology, the emphasis on programming in numerous disciplines has increased (Lee and Cho, 2017). This is the case in many computing related disciplines where students are required to learn programming at introductory level or throughout their qualification (Garov and Tabakova-Komsalova, 2017).

Students who do not intend to pursue programming further can obtain the introductory programming knowledge and skills required for their specific qualification (Robins, Rountree and Rountree, 2003). One issue

which is a common contributor to the loss of interest among programming students is the high failure rate, which often results in students dropping out of their qualification (Antón-Rodríguez *et al.*, 2020).

2.1 Programming Education

Although the failure rate may not be high in more advanced programming courses, many students continue to encounter challenges that make learning programming at introductory level difficult, particularly in classes that emphasise algorithms (Kori *et al.*, 2016). Such challenges include:

- The inability to relate learnt programming knowledge to contexts outside the classroom (Lee and Cho, 2017).
- Large student numbers causing less individual feedback to students.
- The lack of an effective approach for providing programming education to students (Santos et al., 2020).

These challenges are noted in programming education and are crucial in its success. Student learning is influenced by prior programming knowledge and continuous feedback on their work, which allows them to continuously improve their programming knowledge and skills (Bailey and Mentz, 2017).

If implemented correctly, various educational approaches can enhance programming education by offering a standardised method of teaching programming (Kadar *et al.*, 2021). These approaches should be adaptable to diverse settings in which programming education takes place, including large classes. Furthermore, they should incorporate the context of students' learning and work in various educational environments such as online, blended, or traditional classroom education (Özden and Tezer, 2018).

2.2 Online Programming Education

For many higher education institutions, programming education continued to be online during the Covid-19 pandemic, until a blended-learning approach was adopted (Santos *et al.*, 2020). A key challenge in the online environment is that instructors have little control over students' learning (Tufekci and Demirel, 2009).

The objective of online education is to provide a learning experience that is comparable to that of a traditional classroom environment, but in a virtual environment. This can be readily accomplished in subjects that are primarily theoretical in nature. However, programming, being very applied, makes this task more complex. Learning programming seeks to address both knowledge and behavioural aspects, requiring online educational platforms to replicate the effectiveness of traditional classroom environments with benefits of online learning which allows students to explore their own learning styles and revisit the learning material at their own time (Mdunyelwa, Futcher and van Niekerk, 2019). Online programming education enables instructors to use learning management systems and other online educational tools to achieve the following:

- Upload lecture recordings for student content engagement.
- Conduct virtual online classes (Rawashdeh *et al.*, 2021);
- Engage in live coding sessions and grant students' access to the coding examples and video recordings.
- Allow students to utilise collaborative programming platforms such as GitHub.

Various technologies used to achieve the above can be effective in programming education if introduced in a manner suitable for student learning (Shen and Ho, 2020). Online learning approaches should seek to address the needs of the human brain by including content which is based on pedagogically sound educational approaches such as brain-based learning (van Niekerk and Webb, 2016).

2.3 Brain-Compatible Educational Principles

The concept of brain-based learning relates to educational approaches which are based on the underlying biology of learning instead of simply following traditional practices (Jensen, 2000; van Niekerk and Webb, 2016). This educational approach stems from a set of available brain-compatible principles as referred to by various researchers (Jensen, 2000). Although many of these principles are difficult to measure in a learning environment, they can positively impact student learning if considered as part of the learning process.

Comprehensive lists of brain-compatible principles are available the literature and have been used in education, however, due to space limitations, these cannot be depicted in their entirety and can be found in literature

(Caine and Caine, 1990; Jensen, 2000; van Niekerk and Webb, 2016)}. In this study, only the brain-compatible principles which are deemed most relevant to programming education were considered when developing the educational material for the Development Software II module. These included the following:

- There is no long term retention without rehearsal (Jensen, 2000);
- Short, focused learning activities are best (Caine and Caine, 1990; Jensen, 2000);
- Learning with specific context is best (Caine and Caine, 1990);
- Learning should be given choices to accommodate different learning styles (Caine and Caine, 1990; Jensen, 2000);
- Immediate feedback amplifies learning (Jensen, 2000);
- Learning is collaborative and influenced by interactions with others (Jensen, 2000).

In addition, CHAT was used to delineate the online programming educational elements which were considered in the delivery of the educational content.

2.4 Cultural-Historical Activity Theory

The Cultural Historical Activity Theory (CHAT) is a tool typically used for the observation and analysis of people's actions within an organisation, for planning and determining its success (Koszalka, 2004; Czerniewicz, Trotter and Haupt, 2019). CHAT is useful for ensuring a system's thinking view when identifying online programming educational elements. Chang et al (2006) provides a clear view for understanding the concept of system's thinking, where they define a system as a technical set of interrelated parts, all of which work together towards a defined goal (Chang, 2006).

The parts of the system depend on each other in pursuing the target and uses feedback to determine successes of the system (Dick, Carey and Carey, 2006). Chang et al (2006) also provides an analogy of an automotive braking system, where the driver is a less reliable system. They argue that mechanical failure is rarely the cause of accidents that are brake-related, rather that human beings (driver) are largely the contributors of such failures (Chang, 2006). This is also true in situation s where human physiological and psychological characteristics are key elements of a system; it becomes less predictable and more difficult to predict whether the targets will be reached, especially if a specific approach has not been identified.

Consider the case of the adoption of online education in South Africa. With the onset of Covid-19, this was often done without a proper analysis of the elements required for teaching and learning. CHAT provides a more concise theoretical framework to delineate such educational elements and tools.

CHAT was successfully used in a study at a South African university during the #FeesMustFall campaign where teaching and learning had to take place online (Czerniewicz, Trotter and Haupt, 2019). Similarly, CHAT was used in this study to delineate the key elements required for online programming education.

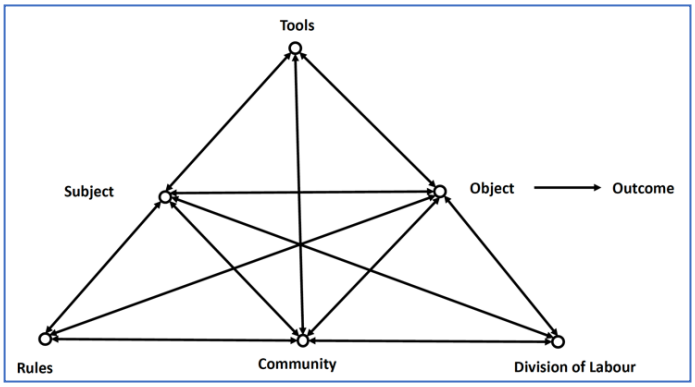

Figure 1: Cultural-Historical Activity Theory (Czerniewicz, Trotter and Haupt, 2019)

Figure 1 presents CHAT in its basic form, where the key elements are presented in each of the triangle ends. The elements include the *tools, subject, rules, community, division of labour, object and outcomes* as discussed below:

- **Tools**- refers to the resources used to achieve the specific outcomes in the planned activities(Czerniewicz, Trotter and Haupt, 2019).

- **Subject** – represents the various parties included in the activities and are required to perform the necessary objectives (Andriani, Priskananda and Budiraharjo, 2022).
- **Rules** - reflects the formal and informal demands which the organisation should perform under (Czerniewicz, Trotter and Haupt, 2019).
- **Community** - includes the stakeholders involved in the activity system (Czerniewicz, Trotter and Haupt, 2019) .
- **Division of Labour** - refers to the persons involved and the duties they are required to perform (Havnes, 2010) .
- **Object** - refers to the assigned task which people are required to work towards (Czerniewicz, Trotter and Haupt, 2019).
- **Outcome** - refers to the target which the activity system would seek to achieve (Havnes, 2010).

These elements interact dynamically and influence each other within a socio-cultural context. It provides a framework for analysing complex activity systems and understanding the socio-cultural dynamics that influence human behaviour and learning.

3. Research Context and Design

This paper is presented in the form of a case study since it involves an in-depth examination of a particular phenomenon or situation within its real-life context (Creswell, 2007). Case studies are particularly useful for exploring complex issues and understanding how different factors interact to shape a particular outcome (Creswell, 2007). According to Creswell, the researcher must clearly define the focus and scope of the research in the research objectives (Creswell, 2007).

This study was conducted at the Nelson Mandela University, School of Information Technology within the second year Development Software module which forms part of the Software Development Diploma. In this module, students are introduced to database programming, Object Oriented Programming, and Web Development. The students are required to write their programs using the .NET framework for Windows Forms and ASP.NET using C#, and databases using Microsoft SQL Server.

This paper presents the theoretical underpinnings and literature that informed the study in Section 2, followed by an exploration of the real-life context of the research in Sections 4, and 5.

The focus of the study is presented in the form of two specific objectives, namely:

- **Objective 1:** To reflect on an approach for delineating online programming educational elements;
- **Objective 2:** To reflect on the brain-compatible principles used to enhance online programming education;

The following section discusses how each of these objectives were achieved and their relevance in online programming education.

4. Research Approach

This study followed a phased approach, comprising of two phases, where each phase represents the objectives identified in Section 3. Objective 1, which seeks to identify an approach for delineating the educational elements required for teaching and learning programming was achieved by the use of the Cultural-Historical Activity Theory (Grimalt-Álvaro and Ametller, 2021). Objective 2, which seeks to identify the principles required for effective online education was achieved by considering the most relevant brain-compatible principles (Caine and Caine, 1990; van Niekerk and Webb, 2016).

Phase 1 incorporated the use of CHAT within the Development Software II module, while Phase 2 highlights the most relevant brain-compatible principles considered in the creation of online educational content for the Development Software II module.

The next section provides a list of brain-compatible principles which were deemed relevant for creating online programming educational material.

5. Using CHAT and Brain-Compatible Principles to Enhance Online Programming Education

CHAT was used to identify the various elements required for teaching and learning programming online within the Development Software II module. This was deemed relevant since programming education requires specific tools and rules for it to be implemented successfully. This is important, especially for online programming education to ensure that students know what is required to learn programming online. The basic elements of CHAT are provided in Section 2. Figure 2 provides an example of the process used for creating the educational material and how the students interacted with the module content. The university's Learning Management System (LMS), Moodle, was instrumental in the teaching and learning of this module. Moodle was adopted by the university before the onset of Covid-19 as its tool for managing student and lecturer interactions. The Development Software II Moodle site is broken down into topics which relate to various learning units within the module. An example of such a topic is Files and Streams which relates to programmatically interacting with file contents, files and folders. This learning unit was split into 3 videos which were uploaded on Moodle using HTML5 Package (H5P).

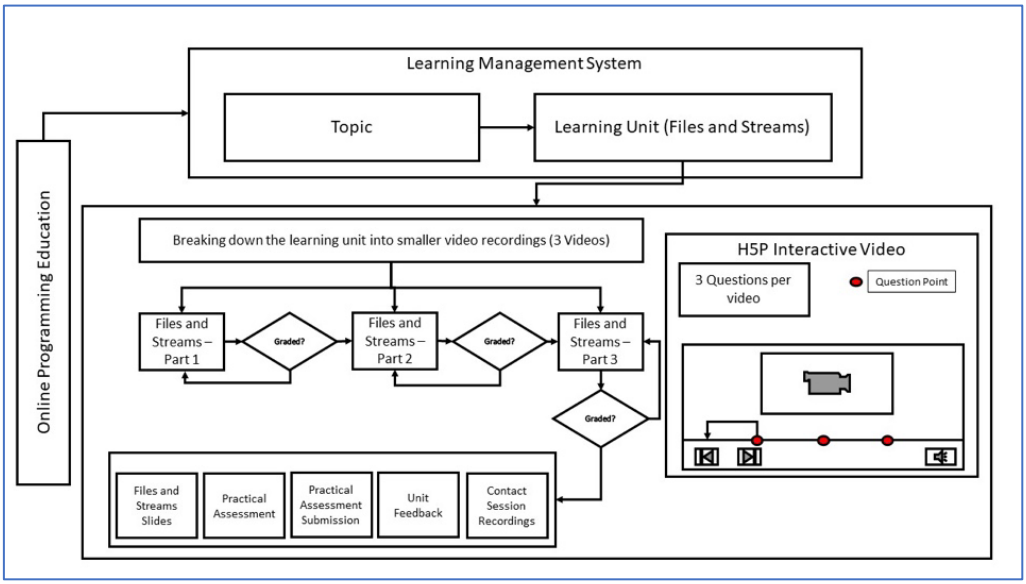

Figure 2: Online Programming Education Steps.

Figure 2 provides the process used for creating the educational material and how the students interact with the module. A learning Management System is instrumental in the teaching and learning of this subject. The learning management system used is Moodle, which has been adopted by the university as its tool for managing student and lecturer interactions. Moodle includes topics which relate to a learning unit in a module, such Files and Streams which relates to programmatically interacting with file contents, files and folders. The learning unit is split into 3 videos where which are uploaded on Moodle using HTML5 Package (H5P).

H5P is a content collaboration package which aims to ease interactive content (Mekterovic *et al.*, 2020). This includes a number of collaborative elements such as interactive videos, course presentations and branching scenarios (Mekterovic *et al.*, 2020). This study made use of interactive videos which allows adding questions to a video for students to answer while watching the video. Various techniques were included such as configuring the video to rewind to the relevant content if the student did not get a specific mark in the question.

Such techniques were included in this exercise to ensure that students understand the content. Since the video was split into 3 parts (Part 1, Part 2, and Part3), a student's access to the next video was restricted by only allowing access to the following video if they got 80% in the previous video. Once students completed Part 3 of the video, they were granted access to the practical assessment, slides, submission and a unit feedback activity for the learning unit. On completion of the learning unit, they were granted access to the next learning unit. Below is a discussion of how these elements relate to online programming education.

5.1 Phase 1: Identifying Online Programming Elements Through CHAT

Figure 2 below provides the elements which relate to the basic CHAT elements as discussed in Section 2.3.

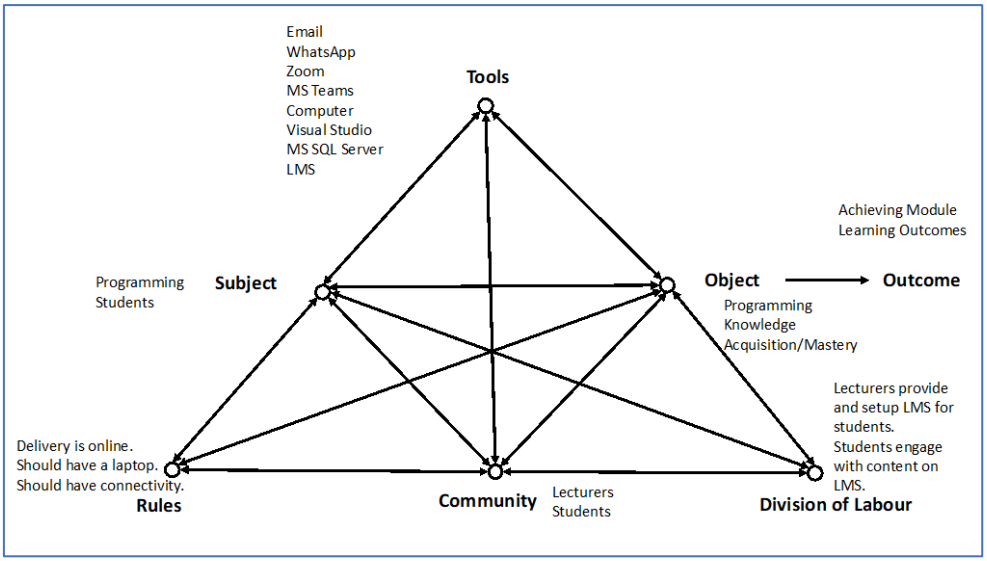

Figure 3: CHAT for Teaching Online Programming Education.

It is important to identify elements which are deemed necessary, relevant or accepted in online environments generally. In this context, CHAT was applied to online programming education to delineate the required elements and understand the complex interactions involved as discussed below:

- **Subject:** In online programming education, the subject refers to the learners who are engaged in learning programming concepts. It includes their prior knowledge, experiences, motivations. Understanding this is crucial for designing effective online programming education (Havnes, 2010).
- **Object:** The object in online programming education is the mastery of programming knowledge and skills. The object can include specific learning outcomes, such as understanding programming concepts, problem-solving abilities, or database programming which is the goal for the Development Software II module. Defining clear and meaningful learning objectives is essential to guide the development of programming content for online teaching and learning (Andriani, Priskananda and Budiraharjo, 2022).
- **Tools:** Online programming education relies on various tools and technological means. These include programming languages, integrated development environments (IDEs) such as Visual Studio or MS SQL Server, code editors, simulation environments, online learning platforms (LMS), discussion forums (WhatsApp, Email), and collaborative coding tools. The choice and use of appropriate tools significantly influence learners' engagement and learning outcomes (Havnes, 2010).
- **Rules:** In online programming education, rules involve instructional guidelines, pedagogical strategies, and assessment criteria. They can include coding standards and problem-solving methodologies. However, since this happened during the Covid-19 pandemic, the rules also include that the delivery of the module should be online without interaction or access to the university facilities. Therefore, student were required to have a reliable internet connection and a laptop. Establishing clear rules and guidelines helps learners understand expectations and promotes consistent and effective learning experiences.
- **Community:** The online programming education community consists of learners, instructors, teaching assistants, and fellow students. It also includes online forums, coding communities, and social media groups where learners can interact, collaborate, and seek help. There were used by both students and the instructor to seek and to provide feedback for students. Fostering a supportive and collaborative community can enhance engagement, knowledge sharing, and peer learning \cite{Gasiba2020}.
- **Division of Labour:** The division of labour in online programming education refers to the distribution of roles and responsibilities among learners and instructors. It involves structuring learning activities, providing instructional materials, facilitating discussions, and offering feedback.

This included gamifying the learning by requiring students to unlock various stages in the various topics to allow students to proceed in the learning units by engaging with the content and answering the various questions. Effective division of labour ensures a balanced workload and supports learners' progression and achievement (Czerniewicz, Trotter and Haupt, 2019).

- **Outcome:** The outcome in online programming education can encompass both tangible and intangible aspects. Tangible outcomes may include learners' developed coding projects, completed assignments. This also was measured through the students' end of year assignment which the students develop to demonstrate the various learning outcomes for the module. Intangible outcomes can include increased problem-solving skills, computational thinking abilities, or improved self-efficacy in programming. Continuously assessing and evaluating learning outcomes helps refine instructional strategies and improve the learning experience (Czerniewicz, Trotter and Haupt, 2019).

By applying CHAT to online programming education, educators can gain a comprehensive understanding of the interplay between these elements and design more effective and engaging learning experiences in preparation to create the online programming educational content as discussed in Section 5.2.

5.2 Phase 2: Brain-Compatible Principles Considered in Online Programming Education

Once the required elements for teaching programming online have been identified, the brain-compatible principles identified in Section 2.3 will be used to reflect on the use in creation online programming education for the Development Software module as discussed below:

- **There is no long term retention without rehearsal** - This was achieved using questions in the H5P interactive content video. While the students were watching the video, a question would appear and they would be required to answer the question. This allowed them to assess their understanding of the content covered in the video.
- **Short, focused learning activities are best** - The learning units for the modules were split into 3 videos which were between 7 and 10 minutes in length. This allowed for the students to be able to focus on specific tasks which would be integrated in the last part of the video. This was important since it allowed students to digest and understand the detail of the learning unit.
- **Learning with specific context is best** - This module is based on the .NET framework and C\# was used as the programming language which the students were required to learn and code in.
- **Learning should be given choices to accommodate different learning styles** - The main benefits of online education is that it allows for a variety of students to study at their own pace. The content for this module was made available to the students without limits on time. The students were allowed to work through the content and revise whenever they wanted to.
- **Immediate feedback amplifies learning** - Feedback is important in programming, especially since students' learning builds on concepts learned in prior topics. In this module, students were provided with feedback in their interactive content when watching the videos. This occurred when students selected incorrect answers when answering multiple choice questions. Students were also required to submit practicals for each learning unit,. The lecturer discussed the solution to the practicals in online classes and provided the students with a solution to apply to their work.
- **Learning is collaborative and influenced by interactions with others** - This principle relates to allowing students to work together in understanding the module. This was achieved in two ways. Firstly, through having contact sessions with the students where the lecturer allowed students to come up with a scenario which could be solved using a relevant programming topic using live coding. Students received access to the recording enabling them to go through the video and practice in their own time. Secondly, providing students with a practical assessment at the end of each learning unit, where the students were allowed to work in groups making use of version control tools and encouraging them to practice pair programming.

The identified brain-compatible learning principles were effective in online programming education. While implementing this approach was time consuming, it assisted students in successfully meeting the learning outcomes of the Development Software I module in the online environment.

6. Conclusion

This paper reported on a case study aimed at improving online educational content for programming education. The paper focused primarily on the creation of content used for student engagement. The study consisted of two phases. Firstly, CHAT was be used as a lens to provide a system thinking overview to improve understanding of the interaction between various educational elements when designing online programming educational content. The use of CHAT allowed for a clear delineation of educational elements through which educators gained a comprehensive understanding of the interplay between these elements. This allowed the educators to address possible problems proactively. Subsequently, the study used brain compatible learning principles as a pedagogical basis for the created online educational material. The use of brain compatible principles was found to be effective for the creation of such content. The authors are not implying that CHAT and brain compatible principles specifically are needed for effective online programming educational design, but rather that using an existing theory, such a CHAT, allows educators to accurately define the scope, interactions, and requirements for learning activities. Once defined, in order to be effective, such activities should be implemented through the use of established educational principles, for example through the use of brain compatible principles.

References

Abesadze, S. and Nozadze, D. (2020) 'Make 21st Century Education: The Importance of Teaching Programming in Schools', *International Journal of Learning*, 6(3), pp. 158–163. Available at: https://doi.org/10.18178/IJLT.6.3.158-163.

Andriani, E., Priskananda, A.A. and Budiraharjo, M. (2022) 'A Cultural-Historical Activity Theory (CHAT) Analysis on Educational Psychology Class: The Challenges in Delivering a Fully Online Classroom Environment', *Journal of Foreign Language Teaching and Learning*, 7(1), p. PRESS. Available at: https://doi.org/10.18196/ftl.v7i1.13196.

Appana, S. (2008) 'A review of benefits and limitations of online learning in the context of the student, the instructor, and the tenured faculty', *International Journal of E-Learning*, 7(1), pp. 5–22.

Bailey, R. and Mentz, E. (2017) 'The value of pair programming in the IT classroom', *Independent Journal of Teaching and Learning*, 12(1), pp. 90–103.

Caine, R.N.R. and Caine, G. (1990) 'Understanding a brain-based approach to learning and teaching', *Educational Leadership*, (October), pp. 66–70.

Chang, S.L. (2006) 'The Systematic Design of Instruction', *Educational Technology Research and Development*, 54(4), pp. 417–420. Available at: https://doi.org/10.1007/s11423-006-9606-0.

Creswell, J.W. (2007) *Qualitative enquiry & research design, choosing among five approaches*, *Book*. Available at: https://doi.org/10.1016/j.aenj.2008.02.005.

Czerniewicz, L., Trotter, H. and Haupt, G. (2019) 'Online teaching in response to student protests and campus shutdowns: academics' perspectives', *International Journal of Educational Technology in Higher Education*, 16(1), pp. 1–22. Available at: https://doi.org/10.1186/s41239-019-0170-1.

Dick, W., Carey, L. and Carey, J. (2006) *The Systematic Design of Instruction*, *Pearson*.

Espinha Gasiba, T., Lechner, U. and Pinto-Albuquerque, M. (2020) 'Sifu - a cybersecurity awareness platform with challenge assessment and intelligent coach', *Cybersecurity*, 3(1). Available at: https://doi.org/10.1186/s42400-020-00064-4.

Fawcett, G. and Juliana, M. (2015) *Teaching in the Digital Age*, *Designing Instruction for Technology-Enhanced Learning*. Available at: http://opentextbc.ca/teachinginadigitalage/%5Cnhttp://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-930708-28-0.ch004.

Garov, K.A. and Tabakova-Komsalova, V.V. (2017) 'Learning content of educational tasks in computer programming training for 10-11 year old children', *TEM Journal*, 6(4), pp. 847–854. Available at: https://doi.org/10.18421/TEM64-26.

Grimalt-Álvaro, C. and Ametller, J. (2021) 'A Cultural-Historical Activity Theory Approach for the Design of a Qualitative Methodology in Science Educational Research', *International Journal of Qualitative Methods*, 20, pp. 1–12. Available at: https://doi.org/10.1177/16094069211060664.

Havnes, A. (2010) 'Cultural-historical activity theory', *International Encyclopedia of Education*, pp. 491–497. Available at: https://doi.org/10.1016/B978-0-08-044894-7.00464-4.

Jensen, E. (2000) 'Brain-based learning: A reality check', *Educational Leadership*, 57(7), pp. 76–80. Available at: https://doi.org/10.1080/00220973.2013.813367.

Kadar, R. *et al.* (2021) 'A Study of Difficulties in Teaching and Learning Programming: A Systematic Literature Review', *International Journal of Academic Research in Progressive Education and Development*, 10(3), pp. 591–605. Available at: https://doi.org/10.6007/ijarped/v10-i3/11100.

Kori, K. *et al.* (2016) 'The Role of Programming Experience in ICT Students' Learning Motivation and Academic Achievement', *International Journal of Information and Education Technology*, 6(5), pp. 331–337. Available at: https://doi.org/10.7763/ijiet.2016.v6.709.

Koszalka, T.A. (2004) 'A cultural historical activity theory [CHAT] analysis of technology integration: Case study of two teachers', *Association for Educational Communications and Technology*, pp. 1–11. Available at: http://files.eric.ed.gov/fulltext/ED485000.pdf.

Lee, Y. and Cho, J. (2017) 'The Influence of Python Programming Education for Raising Computational Thinking', *International Journal of u- and e- Service, Science and Technology*, 10(8), pp. 59–72. Available at: https://doi.org/10.14257/ijunesst.2017.10.8.06.

Mdunyelwa, V., Futcher, L. and van Niekerk, J. (2019) 'An Educational Intervention for Teaching Secure Coding Practices', in *IFIP Advances in Information and Communication Technology*, pp. 3–15.

Mekterovic, I. *et al.* (2020) 'Building a comprehensive automated programming assessment system', *IEEE Access*, 8, pp. 81154–81172. Available at: https://doi.org/10.1109/ACCESS.2020.2990980.

Mukhtar, K. *et al.* (2020) 'Advantages, limitations and recommendations for online learning during covid-19 pandemic era', *Pakistan Journal of Medical Sciences*, 36(COVID19-S4), pp. S27–S31. Available at: https://doi.org/10.12669/pjms.36.COVID19-S4.2785.

van Niekerk, J. and Webb, P. (2016) 'The effectiveness of brain-compatible blended learning material in the teaching of programming logic', *Computers and Education*, 103, pp. 16–27. Available at: http://dx.doi.org/10.1016/j.compedu.2016.09.008.

Özden, C. and Tezer, M. (2018) 'The effect of coding teaching on students' self-efficacy perceptions of technology and design courses', *Sustainability (Switzerland)*, 10(10), pp. 1–29. Available at: https://doi.org/10.3390/su10103822.

Rawashdeh, A.Z. Al *et al.* (2021) 'Advantages and disadvantages of using E-learning in university education: Analyzing students' perspectives', *Electronic Journal of e-Learning*, 19(2), pp. 107–117. Available at: https://doi.org/10.34190/ejel.19.3.2168.

Robins, A., Rountree, J. and Rountree, N. (2003) 'Learning and teaching programming: A review and discussion', *International Journal of Phytoremediation*, 21(1), pp. 137–172. Available at: https://doi.org/10.1076/csed.13.2.137.14200.

Santos, S.C. *et al.* (2020) 'Innovative approaches in teaching programming: A systematic literature review', *CSEDU 2020 - Proceedings of the 12th International Conference on Computer Supported Education*, 1(Csedu), pp. 205–214. Available at: https://doi.org/10.5220/0009190502050214.

Shen, C. wen and Ho, J. tsung (2020) 'Technology-enhanced learning in higher education: A bibliometric analysis with latent semantic approach', *Computers in Human Behavior*, 104(November 2018), p. 106177. Available at: https://doi.org/10.1016/j.chb.2019.106177.

Tufekci, S. and Demirel, M. (2009) 'The effect of brain based learning on achievement, retention, attitude and learning process', *Procedia - Social and Behavioral Sciences*, 1(1), pp. 1782–1791. Available at: https://doi.org/10.1016/j.sbspro.2009.01.316.

Waite, J. and Sentance, S. (2021) 'Teaching programming in schools: A review of approaches and strategies Raspberry Pi Foundation', (November). Available at: www.raspberrypi.org.

Wang, Y. (2019) 'Research on Programming Practice Based on CDIO Concept', 298(Essd), pp. 246–248. Available at: https://doi.org/10.2991/essd-19.2019.54.