OPEN PRAXIS

Applying Threshold Concepts Strategies to Teaching Computing Students in an ODL Context

RESEARCH ARTICLE

JAROSLAW PAWEL ADAMIAK

INTERNATIONAL COUNCIL FOR OPEN AND DISTANCE EDUCATION

ABSTRACT

The academic success of first-year students' learning in science faculties is by no means assured, especially in an Open Distance Learning setting with its limited number of face-to-face encounters between students and lecturers or tutors. Therefore, such encounters should be highly efficient in view of the considerable amount of knowledge transfer to students. The University of South Africa (Unisa) makes provision for contact sessions of 15 hours per semester for selected modules in an attempt to elevate the pedagogical efficiency of these sessions by focusing on the threshold concepts as an innovative way of learning. This paper shows that tutorials adopting the threshold concepts approach have the potential to make students academically more successful. The focus of this study is an introductory information systems module that teaches the Python programming language. Our statistical analysis demonstrated that the year marks and final exam scores of the participating students were frequently higher than those of the students in the control group.

CORRESPONDING AUTHOR:
Jaroslaw Pawel Adamiak
University of South Africa, South Africa
eadamijp@unisa.ac.za

KEYWORDS:
open distance learning; threshold concepts; learning support; programming; innovative pedagogy

TO CITE THIS ARTICLE:
Adamiak, J. P. (2024). Applying Threshold Concepts Strategies to Teaching Computing Students in an ODL Context. *Open Praxis*, *16*(2), pp. 225–236. DOI: https://doi.org/10.55982/openpraxis.16.2.552

DOI: 10.55982/openpraxis.16.2.552

INTRODUCTION

Open Distance Learning (ODL) students' success rates are, on average, lower than those of students at contact universities. For example, the United Kingdom Open University's graduation rates for 2010 were at 22% for ODL students, at 82% for full-time students, and at 39% for part-time students. It was predicted that graduation rates could drop even further (Simpson, 2010). In other countries, students at major ODL universities did not perform better. Graduation rates were 14% at the Dr Ambedkar University in India; 6% at the University of South Africa (Unisa); 5% at the Athabasca University in Alberta, Canada; and only 0.5% at the Tele-university of Quebec (Simpson, 2013).

The above percentages indicate the urgent need to improve students' success rates. It is possible that prospective students may regard ODL institutions as a last resort rather than universities of choice. This downward trend may not only harm the standing of ODL degrees in the industry but could also deter high-potential students from registering at ODL institutions. It may even lead to a further decline in ODL institutions' performance.

At Unisa, the largest ODL university on the African continent, no weekly lectures or tutorials characteristic of the pedagogical approach at traditional contact universities, are offered. Students are expected to work without supervision and use only study materials provided by the university, loaned at its libraries, or bought at booksellers. Limited support is offered by online facilitators called e-tutors (Pitsoane et al., 2015) in the form of online materials produced or curated within the university and, to a quite limited extent, short workshops called face-to-face classes.

Because of a need to cut operational costs, these workshops are often presented by ad-hoc lecturers or subject experts who have nearly no academic or contractual ties with the university. The downside is that pedagogies that are not always aligned with the university educational philosophy may be applied during workshops. Also, the time constraints of 15 hours per semester render some of the efficient and popular pedagogies such as group work, mini-projects, or extensive discussions impossible. Moreover, the time that tutors devote to struggling students is often inadequate.

Additionally, it is difficult for isolated students stuck on a concept critical for understanding the rest of the material to get help, particularly in courses that require students to master an ample block of knowledge before they can get to grips with the next one. Science subjects are typical examples of such courses.

In this paper, we attempt to find an educational silver bullet, a pedagogy that, in an ODL environment, would significantly improve the passing rates of students. If successful, this methodology could be imparted to tutors before they educate students in workshops. Such a pedagogy might be based on the threshold concepts (TCs), a term that describes any challenging block of knowledge that must be mastered before a next topic can be dealt with.

According to Meyer & Land (2003), a TC can be defined as an idea in the curriculum that is:

- Transformative in that its learning leads to a changed perception of the subject or discipline.
- Integrative since it allows students to see relationships within a discipline that they were previously unaware of.
- Irreversible because once it is learned, it is difficult (or impossible) to unlearn or forget.
- Potentially troublesome for students because it may be conceptually difficult, alien, or counter-intuitive; may require explicit knowledge of a concept that has been tacit; or may require a deeper understanding of a concept than merely ritual knowledge.
- Bounded because a concept is understood only by disciplinary insiders and defines the boundaries of a discipline.

In the learning of TCs, students often find themselves in a liminal state, a state of being on the threshold. It is characterised by an oscillation between of the old and the new understandings and confusion about them, by an emotional response, and a feeling of being stuck (Meyer & Land, 2005).

PURPOSE OF STUDY

TCs originated from the notion of troublesome knowledge, knowledge that is conceptually difficult, counter-intuitive or alien (Perkins, 1999). Introduced by Meyer & Land (2003), TCs quickly gained popularity and was soon the subject of a great many papers and books (Meyer & Land 2006; Land, et al., 2008; Land at al., 2016; Meyer et al., 2010). After TCs had become ubiquitous in academia, Michael Thomas created a web page listing past and current developments in TCs (Flanagan, 2020). Since May 2022 the list of disciplines in which publications about TCs appeared, surpassed 200 entries.

Several techniques serve to identify a TC in a certain scientific area or discipline. Semi-structured interview analyses of exam responses and observation of classroom behaviour are two of them (Barradell, 2013). Another involves a survey amongst lecturers, followed by a focus group session and in-depth interviews (Galligan et al., 2010). Using a questionnaire for students, Baillie et al. (2006) identified complex numbers as a TC in the engineering field. Townsend et al. (2016) applied the Delphi method to a small group of educational experts to find and validate TCs in the domain of information literacy. Faculty workshops and student interviews were a source of TCs for Loertscher et al. (2014). Barradell (2013) maintains that the TC identification process should be rigorous and strengthened by involving participants outside the educational domain. She also advocates the consensus methodology in combination with other research methods.

Novice programmers find computer science and programming quite difficult (Mannila et al., 2006). It is understandable because they have to grasp dozens of TCs (Sanders & McCartney, 2016). Another reason is students' anxiety when they have to cross the liminal space of computer programming (Chetty & van der Westhuizen, 2013).

To our knowledge, TCs in computer science were first mentioned in the work of McCartney & Sanders (2005). Because of the frequent occurrence of TCs in mathematics and physics, they decided to identify TCs in their domain of expertise and come up with the most appropriate pedagogies for addressing them. Program-memory interaction was proposed as a TC and examined by Vagianou (2006). Eckerdal et al. (2006) dealt with TCs according to constructivist learning theory from a computer science point of view. According to them, TCs are a particularly troublesome part of a student's knowledge framework. They suggested abstraction and object-orientation as TCs. Khalife (2006) discusses the difficulties that novice programmers encounter and claims that the computer model itself is a TC. Consequently, students need to develop a concrete mental model of computer internals and operations during program execution to cross the liminal space. An interesting generalisation of TCs in programming education, called anchor concepts, is proposed by Mead et al. (2006). Anchor concepts should be either only foundational or both integrative and transformative. Owing to this relaxation, anchor concepts are easier to identify than TCs, and more promising and relevant in curriculum planning. The empirical identification of TCs was undertaken by Boustedt et al. (2007) by means of questionnaires destined for computer science instructors and students and interviews with them. They found evidence of TCs in computer science and identified two of them: pointers and object-oriented programming. Eckerdal et al. (2007) conducted interviews with graduate seniors in computer science to investigate the usefulness of the liminal-space metaphor. They found that although TCs are useful constructs, they cannot be easily ordered since students take different routes through liminal space. Students also reported that learning TCs takes a considerable time and may stimulate negative attitudes. Moström et al. (2008) asked a group of students to identify and describe the computing concepts that transformed their view of computer science. An analysis of their answers produced several concepts that could be classified as TCs, namely modularity, data abstraction, object-orientation, code reuse, design patterns, and complexity. Rountree and Rountree (2009) link TCs in computer science to student success as well as the type and timing of support during the learning process. They argue that the idea of recursion is an example of a TC in computer science since it meets all the TC definition requirements. Sorva (2010) introduced the notion of transliminal concepts, which are concepts that can be understood only when a student's perspective has been transformed by a TC. In programming, instantiation and pointers would be transliminal concepts. Sien & Chong (2012) analyse object-oriented modelling from a TCs angle and conclude that classes, generalisation-specialisation hierarchies, and object interaction should be added to the list of computer science TCs. Rountree et al. (2013) contend that any successful crossing of the liminal space requires not only the acquisition and internalisation of the knowledge, but also "its practical elaboration in the domains of applied strategies and mental models". Shinners-Kennedy & Fincher (2013) summarised the work of a group of researchers who had to identify and report TCs in computer science education. The authors predicted that looking for evidence of TCs in teachers' pedagogical presentations instead of in learners' knowledge acquisition would be the next step in the discipline. As part of their experimental approach to TCs by using Python programming language, Miller, Settle & Lalor (2015) conducted an exploratory study in an interactive programming environment. Five students doing inheritance exercises were requested to share everything they think during the programming process. The analysis indicated parameter passing and dot-object notation as TC candidates in Python programming. Sanders & McCartney (2016) give the most comprehensive overview of TCs in computer science. They conducted an extensive literature research and tabularised data TCs in computer science with citations. They also listed the instruments they have used for gathering data about TCs, and proposed an intervention that would facilitate liminal-space crossing.

Despite these substantial theoretical developments, few papers describe the practical implementation of TCs in the computer science classroom and offer a formal evaluation of students' progress. Khalife (2006) used TCs in introductory object-oriented programming in Java. He reports a significant improvement in student scores compared to their scores in previous courses on the same topic. Sien (2011) applies concept maps for teaching object-oriented analysis and contends that students subjected to the intervention produce a better class-and-sequence diagram than the control group.

While there is a significant body of literature focused on defining and identifying TCs across various academic fields, little research has been conducted on the impact of TC-related pedagogies on academic outcomes. Thus, our study aims to address the following research questions:

- Does a TC-based pedagogy improve formative test results?
- Does a TC-based pedagogy improve summative tests results?
- Does a TC-based pedagogy accelerate students' knowledge acquisition?

In this study, we aim to address the knowledge gap surrounding the practicality of TCs in ODL pedagogies. To ensure scientific rigor, we established a quasi-experimental setting by conducting the experiment three times with different student groups in an almost identical environment, employing independent outcome measurement, and using widely accepted statistical methods to support our conclusions. The significance of this research lies in two areas: firstly, the pressing need to enhance student retention and success in ODL, and secondly, the effect size of the outcomes we obtained. We aspire to meet the needs of curriculum designers by providing efficient pedagogical strategies for developing ODL courses and producing a manual to support academics in delivering these courses.

METHOD

In response to the research questions, we applied quantitative research techniques to select a student cohort, provide a set of interventions, and perform a statistical analysis to determine whether the intervention affected students' assignment and exam results. Due to ethical concerns, which we could summarize as a purposeful non-delivery of an educational support to a group of students, we were unable to apply a full experimental setting with two randomly constituted student groups, so we decided to use a quasi-experiment instead. In the context of this study, according to Shadish et al. (2002) the employed design is called the "posttest-only control group design with non-equivalent groups".

The student group that participated in the quasi-experiment was not randomly selected but consisted of students who responded positively to the invitation to participate in the intervention workshops, thus the sample was affected by self-selection bias. Additionally, we could expect an additional bias caused by the fact that the participating students were in the geographical proximity of the campus where the interventions took place, what might lead to geographical bias (González Canché, 2019). The comparison group was the full cohort of students enrolled in the study unit but not participating in the intervention.

To safeguard the rights of participating students, the detailed plan research plan was submitted to the Unisa ethics committee before the investigation commenced. Only after the committee was satisfied that the research will be conducted according to the university's code of ethical research, permission 2017_PRC_REW_005_Ex was given. During the recruiting process and at the workshops, students were advised that they would be equipped with knowledge according to a method that has its roots in experimental pedagogy.

The experimental environment we decided on was the introductory module INF1511. It is part of the undergraduate curriculum for computer science and information systems and teaches students the Python programming language. This choice was dictated by, first, the popularity of this module, in other words, the substantial number of students registering for it and, second, the numerous TCs found in this module. During the second semester of 2018, we followed a similar approach in another undergraduate module in the programming language C++. Unfortunately, the small the number of students participating in the project made it impossible to generalise the outcomes and after one attempt, we decided not to continue with C++.

The workshops that were designed around programming TCs were presented to Unisa students located in the Western Cape Province. The number of students in this province taking the INF1511 module fluctuates around 100 every semester. We hosted four four-hour workshops in the second semester of 2017 (2017/S2) and five three-hour workshops in the first and second semesters of 2018 (2018/S1 and 2018/S2). Additionally, a portal with links arranged thematically to various relevant internet resources was created. Students participating in the intervention enrolled as users of this portal. Moreover, notes from the workshops, interesting results and insights, and the code used during the workshops were published on the portal.

Although we tried to cover a wide range of TCs as suggested by Sanders & McCartney (2016) and Shinners-Kennedy & Fincher (2013), limited time forced us to concentrate on only a few major ones. TCs we dealt with involved object orientation in various respects such as abstraction, classes, objects, modularisation, dot-notation, object interaction, class declaration, information hiding, inheritance, polymorphism, operator overloading, and code reuse. Less important TC themes included recursion, scope, and lambda functions. The variable-assignment statement in contrast to the mathematical equality sign, parameter passing, and complexity were also mentioned.

In the workshops, the following educational approaches were adopted. First, each concept was explained in various ways to cater for the diversity and various background of students. Second, cases and experiences students were familiar with were used to explain TCS. Third, students were required to change an existing code, add a new code, and program a simple solution from the very beginning.

Object-Oriented Programming was explained by dealing with the growing complexity of the data structures and the programming's suitability for a reflection of the real world. Students participated by building a fictitious University Tracking System containing various types of students, depending on their progress in academia. Hierarchies and inheritances were created by implementing student objects at various levels of university enrolment (undergraduate, master's, PhD, etc.). The classes and objects concepts started with the idea of an empty class in which attributes and methods were gradually added. Based on the class template, the objects were created, and their functionalities were reviewed and tested. This process-supported teaching included not only fundamental, but also sophisticated programming methods such as the application of constructors, decorators, static and class functions, as well as variables. The presentation concluded with the implementation of the student system hierarchies. If time allowed, examples illustrating the applicable ideas from various fields (such as physics, economics, and biology) were given. Recursion, known for its learning difficulties (McCauley et al., 2015), was approached by means of an analysis of the mathematical definition (after assurance that the students are suitably mathematics literate), a game in which each student performed a specific recursive function call, and the trace tables. The lambda functions were first explained by an analysis of the traditional mathematical function and functions used in Python. Then the lambda construct was presented. Several examples of the application of lambda functions to vectors were worked through. Finally, students were asked to design and implement their own lambda functions.

The reliability of the above method was ensured by retesting and the quasi-experiment was performed three times on different cohorts of students. The outcomes yielded consistent results and supported the research hypothesis. To ensure the validity, we used the construct approach. We wanted to find out if the students undergoing the TC-based intervention performed better than the general cohort of students.

In our study, we employed assignment and exam outcomes as the key measures of student performance. These measures are widely recognized as effective indicators of student learning and academic achievement. Assignments are an important tool for assessing student understanding of course material and their ability to apply the concepts they have learned. By completing assignments, students can demonstrate their mastery of the subject matter and their ability to think critically and creatively about course material (Black & Wiliam, 1998; Brookhart, 2010).

Exams, on the other hand, are often used as a comprehensive measure of a student's knowledge and understanding of a particular subject. Exams are designed to assess a student's ability to recall and apply the concepts they have learned throughout the course (Brown et al., 2013). In our statistical analysis, we utilized both assignments and exams to evaluate the impact of the intervention on student performance. By examining the changes in these measures over time, we were able to assess the effectiveness of the intervention and determine whether it had a significant impact on student learning and achievement.

To address the potential for confounding variables and ensure that our findings were robust, we employed several controls in our study. First, we used a comparison group consisting of the full cohort of students enrolled in the study unit. This allowed us to compare the performance of students who received the intervention to those who did not, providing a baseline for evaluating the effectiveness of the intervention. Second, we took measures to ensure that the intervention was implemented consistently across all participants. We provided clear instructions and guidelines to the intervention group to ensure that the intervention was delivered in a standardized manner.

Since the sample sizes are rather small and the samples themselves do not exhibit a strict normal distribution, we decided to perform the Wilcoxon test, as it is a non-parametric statistical procedure often employed in hypothesis-testing setups involving a single sample (Sheskin, 2003). In research, the effect size is crucial as it measures the magnitude of difference between groups, moving beyond merely identifying if the groups differ statistically (Cohen, 2013). We have furnished this metric for the comparative analysis conducted. The trend of improvement in the assignment marks of the students over the course of the study was evaluated with help of the Pearson correlation that measures the degree and the direction of the linear relationship between two variables (Gravetter et al., 2020).

FINDINGS

Following the announcement of the exam results, we extracted the necessary data from our institutional databases, which included two academic indicators that were used to assess students' academic success. These indicators were the year marks, which represent the average of all assignments submitted by students during the semester, and the final exam marks obtained at the end of the course.

Both of these indicators were used as input data for our subsequent statistical analysis. It's worth noting that we did not use the final course score, which is a computed score comprising 20% of the year mark and 80% of the exam mark. Instead, we used the year mark and the final exam mark as separate indicators to ensure a more accurate assessment of student performance.

The sizes of the control groups, which comprised all students taking the course but not participating in our interventions, and of the samples, which comprised students participating in our interventions, together with the statistical results for the year mark and exam results for three teaching periods are summarised in Table 1.

We also reported on the effect size calculated according to Cohen's method (Cohen, 2013). The effect sizes we calculated, according to Sullivan & Feinn (2012) can be classified as being between small (0.2) and medium (0.5). Thus, although the effect observed in the study is not extremely large or significant, it is still meaningful and noteworthy.

YEAR/SEMESTER	2017/S2 (4 × 4 hrs)	2018/S1 (5 × 3 hrs)	2018/S2 (5 × 3 hrs)
Control #	547	550	500
Sample #	21	15	21
Year Mark			
Control mean	66.3	60.9	59.1
Sample mean	74.8	68.2	70.3
P-value	0.044	0.094	0.052
Effect size	0.32	0.25	0.38
Exam			
Control mean	47.1	45.7	59.7
Sample mean	58.7	48.8	68.2
P-value	0.018	0.112	0.011
Effect size	0.45	0.25	0.44

Table 1 Control and sample groups sizes, year mark and exam results (both in percentage points) for the TC quasi-experiment. Data is listed for the three consecutive semesters when the interventions took place.

After analysing the data on the comparative assignment results, we sought to estimate the speed at which students improved their subject mastery when the TC pedagogy was applied. We focused on the changes in students' assignment marks over time as a way to track their progress. Our analysis revealed a trend of improvement in the assignment marks of the students over the course of the study. This suggests that the TC pedagogy was effective in promoting student learning and mastery of the subject. We were able to observe a consistent increase in the assignment marks of the students, as depicted on Figures 1, 2 and 3, indicating that they were making progress towards their learning goals.

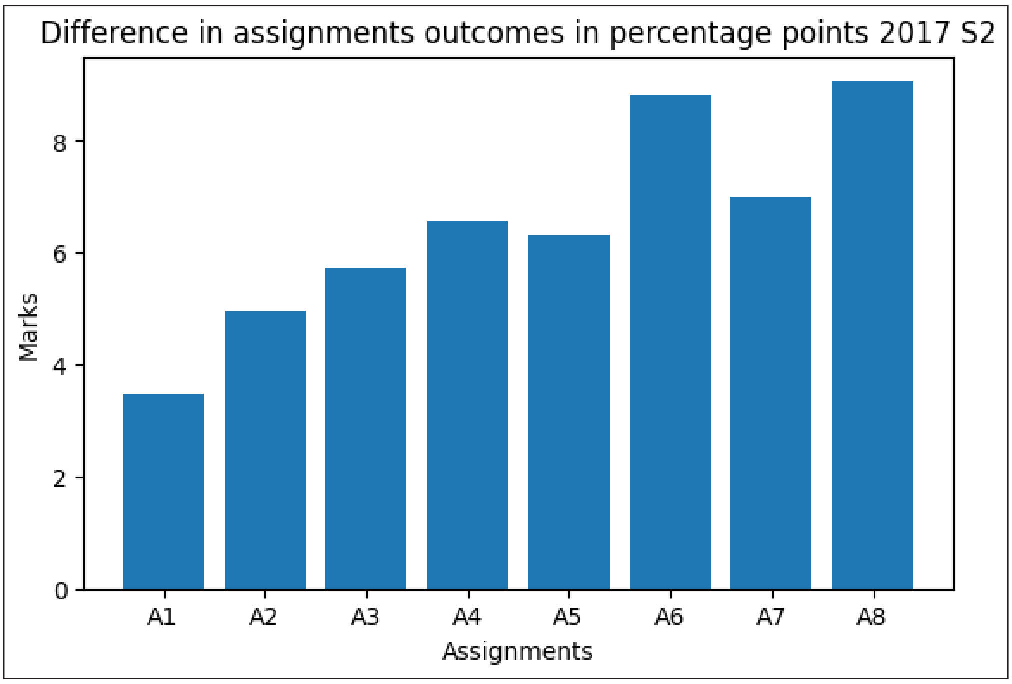

Figure 1 The difference in marks (equivalent to the percentage points) in consecutive assignments during the second semester of 2017.

The difference in marks between experimental and control groups is almost always positive and increases with time. We calculated the Pearson correlation coefficient and the p-value for testing non-correlation, which are presented in Table 2.

These results indicate that students who were taught according to the TC-centred method mastered the knowledge faster than students in the control groups. However, it must be noted that in 2018, semester 1, the correlation is not significant ($p = 0.465$), this is possibly a side effect of the small sample size.

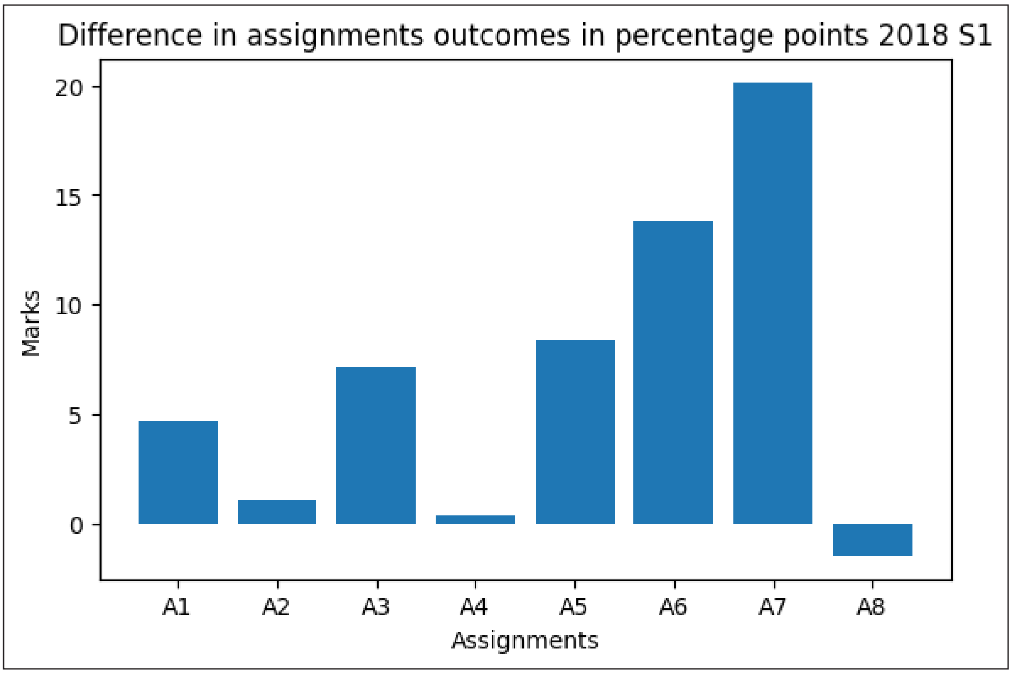

Figure 2 The difference in marks (equivalent to the percentage points) in consecutive assignments during the first semester of 2018.

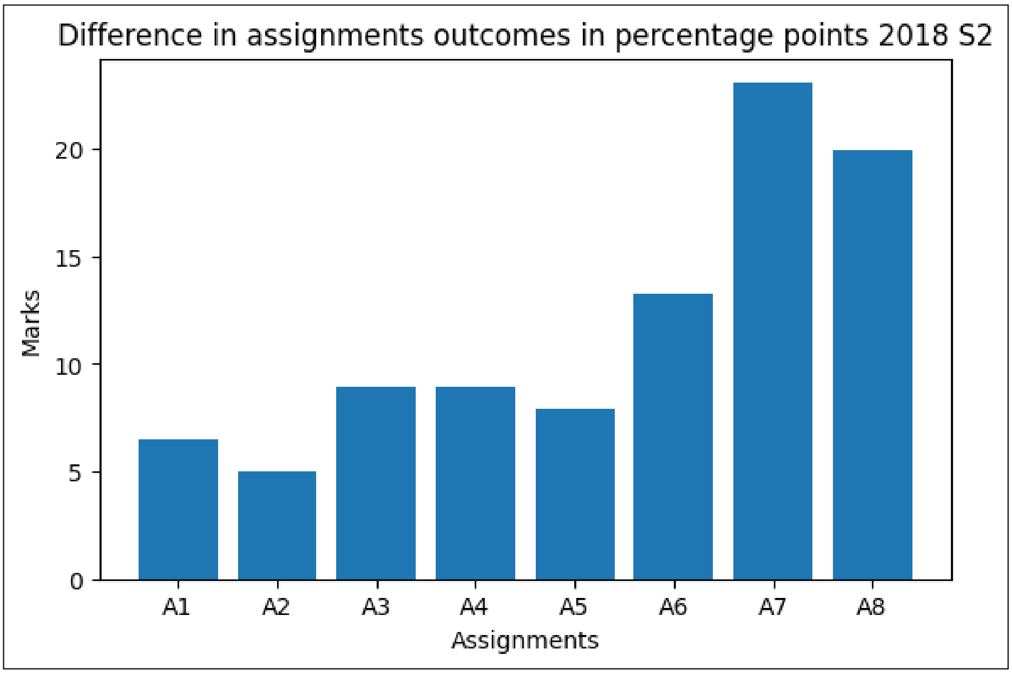

Figure 3 The difference in marks (equivalent to the percentage points) in consecutive assignments during the second semester of 2018.

Table 2 Linear approximation of results growth (the difference between the average sample mark and the average control group mark) over a series of consecutive assignments submitted by students in the sample.

YEAR/SEMESTER	2017/S2 (4 × 4 hrs)	2018/S1 (5 × 3 hrs)	2018/S2 (5 × 3 hrs)
Pearson	0.91	0.30	0.87
P-value	0.002	0.465	0.005

DISCUSSION AND CONCLUSIONS

In this work, we tested the TC-based pedagogy designed by Meyer & Land (2003) in a quasi-experimental setting and found comparable outcomes to those obtained by Khalife (2006) and Sien (2011). We have shown that 15 to 16 hours of tutorials, during which only TCs were taught, improved students' knowledge of computer science and increased the probability of their passing the course.

In the 2017/S2 experimental setup, the research hypothesis was confirmed: the sample's year mark and exam outcomes surpassed those of the general cohort, with a moderate effect size. In 2018/S1, although the results were positive, they were not statistically significant—this lack of significance was likely due to the small sample size. In 2018/S2, the exam results supported the hypothesis, but the year mark, though positive, did not reach statistical significance.

When measuring the speed at which students assimilated knowledge, we observed positive outcomes. During both 2017/S2 and 2019/S2, students in the test sample produced superior assignments, and the gap between them and the control group increased with each assignment. This suggests that knowledge growth in the intervention sample outpaced that in the control group. However, the 2018/S1 results were less conclusive in terms of statistical significance. Therefore, with a moderate level of confidence, we can assert that our research questions have been affirmatively addressed.

Although the workshops were offered to all students in the region (there are nine regions in South Africa), only some students opted to participate. Therefore, the results may suffer from a self-selection bias and therefore one should remain cautious when generalising them. Also, we engaged with a discipline that, fortunately, abounds with particularly well-defined TCs, which is not always true of other academic subjects. Consequently, the following recommendations are made:

- The full experiment rather than the quasi-experiment should be conducted to make the argument for TC applicability stronger.
- Should the execution of the full experiment prove infeasible, conducting a quasi-experiment with a substantially increased number of participants might lead to more conclusive results.
- The TC pedagogy should be tested in the same workshop setting for other disciplines and if the outcome of a such tests support our conclusions, this pedagogy should be extended to other academic subjects.

These recommendations, however, must be treated with caution from the generalisability and transferability points of view. First, a full experiment requires a placebo sample, which, in our case, was a group of students taught according to traditional methods. This approach raises some ethical issues, for example, the denial of potentially superior tutoring services to some of the students. Therefore, creating a full experimental setting seems to be quite difficult. What would strengthen our argument and improve the conclusion's generalisability, is a repetition of the quasi-experiment in a few other university's regional centres with the infrastructure and capabilities to conduct the workshops. Second, TCs are not as well defined in other than computer science disciplines, which severely limits the transferability of the outcomes. To remedy this, much preparatory work should be done to promote the recognition of the TCs and constructing appropriate interventions before TC pedagogy is applied to learning in those disciplines.

Overall, the adoption of TC pedagogy in higher learning could lead to improved academic success and better-prepared graduates. By exploring the potential implications of our findings, we hope to encourage further discussion and research on the application of TC pedagogy in higher education.

DATA ACCESSIBILITY STATEMENT

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

ETHICS AND CONSENT

To safeguard the rights of participating students, the detailed plan research plan was submitted to the Unisa ethics committee before the investigation commenced. Only after the committee was satisfied that the research will be conducted according to the university's code of ethical research, permission 2017_PRC_REW_005_Ex was given. During the recruiting process and at the workshops, students were advised that they would be equipped with knowledge according to a method that has its roots in experimental pedagogy.

COMPETING INTERESTS

The author has no competing interests to declare.

AUTHOR CONTRIBUTIONS (CRediT)

Adamiak, Jaroslaw Pawel: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. The author has read and agreed to the published version of the manuscript.

DOI: 10.55982/openpraxis.16.2.552

AUTHOR(S) NOTES

This paper was reviewed and refined with the assistance of OpenAI's GPT4 (Version as of February 2024), complementing the human editorial process. The human author critically assessed and validated the content to maintain academic rigor. The author also assessed and addressed potential biases inherent in the AI-generated content. The final version of the paper is the sole responsibility of the human author.

AUTHOR AFFILIATIONS

Jaroslaw Pawel Adamiak orcid.org/0000-0001-6535-6831
University of South Africa, South Africa

REFERENCES

Baillie, C., Goodhew, P., & **Skryabina, E.** (2006). Threshold concepts in engineering education-exploring potential blocks in student understanding. *International Journal of Engineering Education*, *22*(5), 955. https://www.ijee.ie/articles/Vol22-5/06_ijee1823.pdf

Barradell, S. (2013). The identification of threshold concepts: A review of theoretical complexities and methodological challenges. *Higher Education*, *65*(2), 265–276. DOI: https://doi.org/10.1007/s10734-012-9542-3

Black, P., & **Wiliam, D.** (1998). Assessment and classroom learning. *Assessment in Education: principles, policy & practice*, *5*(1), 7–74. DOI: https://doi.org/10.1080/0969595980050102

Boustedt, J., Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., & **Zander, C.** (2007, March). Threshold concepts in computer science: do they exist and are they useful? *ACM SIGCSE Bulletin*, *39*(1), 504–508. DOI: https://doi.org/10.1145/1227504.1227482

Brookhart, S. M. (2010). How to assess higher-order thinking skills in your classroom. Alexandria, VA: ASCD. https://lccn.loc.gov/2010018842

Brown, G. A., Bull, J., & **Pendlebury, M.** (2013). *Assessing student learning in higher education*. Routledge. DOI: https://doi.org/10.4324/9781315004914

Chetty, J., & **van der Westhuizen, D.** (2013, June). "I hate programming" and Other Oscillating Emotions Experienced by Novice Students Learning Computer Programming. *EdMedia: World Conference on Educational Media and Technology*, *2013*(1), 1889–1894. https://www.learntechlib.org/primary/p/112226/

Cohen, J. (2013). Statistical power analysis for the behavioral sciences. Academic press, 2013. DOI: https://doi.org/10.4324/9780203771587

Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., & **Zander, C.** (2006, June). Putting threshold concepts into context in computer science education. *ACM SIGCSE Bulletin*, *38*(3), 103–107. DOI: https://doi.org/10.1145/1140124.1140154

Eckerdal, A., McCartney, R., Moström, J. E., Sanders, K., Thomas, L., & **Zander, C.** (2007, September). From Limen to Lumen: computing students in liminal spaces. *Proceedings of the third international workshop on Computing education research* (pp. 123–132). ACM. https://dl.acm.org/doi/abs/10.1145/1288580.1288597

Flanagan, M. (2020, May 08). *Threshold Concepts: Undergraduate Teaching, Postgraduate Training, Professional Development and School Education*. Accessed December 12, 2019. https://www.ee.ucl.ac.uk/~mflanaga/thresholds.html

Galligan, L., Wandel, A. P., & **Hartle, R. T.** (2010). Scaffolding distance learning in mathematics for engineering: Identifying key troublesome knowledge. *Proceedings of the 1st Science, Technology, Engineering and Mathematics in Education Conference (STEM 2010)*. Queensland University of Technology. https://core.ac.uk/download/pdf/11047354.pdf

González Canché, M. S. (2019). Repurposing Standardized Testing for Educational Equity: Can Geographical Bias and Adversity Scores Expand True College Access? *Policy Insights from the Behavioral and Brain Sciences*, *6*(2), 225–235. DOI: https://doi.org/10.1177/2372732219861123

Gravetter, F. J., Wallnau, L. B., Forzano, L. A. B., & **Witnauer, J. E.** (2020). *Essentials of statistics for the behavioral sciences*. Cengage Learning.

Khalife, J. T. (2006, June). Threshold for the introduction of programming: providing learners with a simple computer model. *28th International Conference on Information Technology Interfaces, 2006*. (pp. 71–76). IEEE. DOI: https://doi.org/10.1109/ITI.2006.1708454

Land, R., Meyer, J. H., & **Flanagan, M. T.** (Eds.), (2016). *Threshold Concepts in Practice*. Rotterdam: Sense Publishers. DOI: https://doi.org/10.1007/978-94-6300-512-8

Land, R., Meyer, J. H., & **Smith, J.** (2008). *Threshold concepts within the disciplines*. Rotterdam: Sense Publishers. DOI: https://doi.org/10.1163/9789460911477

Loertscher, J., Green, D., Lewis, J. E., Lin, S., & **Minderhout, V.** (2014). Identification of threshold concepts for biochemistry. *CBE—Life Sciences Education*, *13*(3), 516–528. DOI: https://doi.org/10.1187/cbe.14-04-0066

Mannila, L., Peltomäki, M., & **Salakoski, T.** (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, *16*(3), 211–227. DOI: https://doi.org/10.1080/08993400600912384

McCartney, R., & **Sanders, K.** (2005, November). What are the "threshold concepts" in computer science. *Proceedings of the Koli Calling 2005 Conference on Computer Science Education* (p. 185).

McCauley, R., Grissom, S., Fitzgerald, S., & **Murphy, L.** (2015). Teaching and learning recursive programming: a review of the research literature. *Computer Science Education*, *25*(1), 37–66. DOI: https://doi.org/10.1080/08993408.2015.1033205

Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., Clair, C. S., & **Thomas, L.** (2006). A cognitive approach to identifying measurable milestones for programming skill acquisition. *ACM SIGCSE Bulletin*, *38*(4), 182–194. DOI: https://doi.org/10.1145/1189136.1189185

Meyer, J., & **Land, R.** (2003). *Threshold concepts and troublesome knowledge: Linkages to ways of thinking and practicing within the disciplines* (pp. 412–424). Edinburgh: University of Edinburgh.

Meyer, J. H., & **Land, R.** (2005). Threshold concepts and troublesome knowledge (2): Epistemological considerations and a conceptual framework for teaching and learning. *Higher education*, *49*, 373–388. DOI: https://doi.org/10.1007/s10734-004-6779-5

Meyer, J., & **Land, R.** (2006). *Overcoming barriers to student understanding: Threshold concepts and troublesome knowledge*. London: Routledge. DOI: https://doi.org/10.4324/9780203966273

Meyer, J. H., Land, R., & **Baillie, C.** (Eds.), (2010). *Threshold concepts and transformational learning* (pp. 303–316). Rotterdam: Sense Publishers. DOI: https://doi.org/10.1163/9789460912078

Miller, C. S., Settle, A., & **Lalor, J.** (2015, September). Learning object-oriented programming in python: Towards an inventory of difficulties and testing pitfalls. *Proceedings of the 16th Annual Conference on Information Technology Education* (pp. 59–64). DOI: https://doi.org/10.1145/2808006.2808017

Moström, J. E., Boustedt, J., Eckerdal, A., McCartney, R., Sanders, K., Thomas, L., & **Zander, C.** (2008). Concrete examples of abstraction as manifested in students' transformative experiences. *ICER'08, September 6–7, 2008,* Sydney, Australia (pp. 125–136). ACM. DOI: https://doi.org/10.1145/1404520.1404533

Perkins, D. (1999). The many faces of constructivism. *Educational leadership*, *57*(3), 6–11.

Pitsoane, E., Mahlo, D., & **Lethole, P.** (2015). "UNISA e-tutors' perceptions, experiences & views of active learning." *International Journal of Educational Sciences*, *9*(1), 29–36. DOI: https://doi.org/10.1080/09751122.2015.11890291

Rountree, J., Robins, A., & **Rountree, N.** (2013). Elaborating on threshold concepts. *Computer Science Education*, *23*(3), 265–289. DOI: https://doi.org/10.1080/08993408.2013.834748

Rountree, J., & **Rountree, N.** (2009, January). Issues regarding threshold concepts in computer science. *Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95* (pp. 139–146). Australian Computer Society, Inc..

Sanders, K., & **McCartney, R.** (2016, November). Threshold concepts in computing: past, present, and future. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 91–100). ACM. DOI: https://doi.org/10.1145/2999541.2999546

Shadish, W. R., Cook, T. D., & **Campbell, D. T.** (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Boston, MA: Houghton Mifflin.

Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical procedures*. Chapman and Hall/CRC. DOI: https://doi.org/10.4324/9780203489536

Shinners-Kennedy, D., & **Fincher, S. A.** (2013, August). Identifying threshold concepts: from dead end to a new direction. *Proceedings of the ninth annual international ACM conference on international computing education research* (pp. 9–18). ACM. DOI: https://doi.org/10.1145/2493394.2493396

Sien, V. Y. (2011). Teaching object-oriented modelling using concept maps. *Electronic Communications of the EASST*, 34. DOI: https://doi.org/10.14279/tuj.eceasst.34.590

Sien, V. Y., & **Chong, D. W. K.** (2012). Threshold concepts in object-oriented modelling. *Electronic Communications of the EASST*, 52. DOI: https://doi.org/10.14279/tuj.eceasst.52.763

Simpson, O. (2010). 22%-can we do better? CWP Retention Literature Review. *Report to the Centre for Widening Participation*, Open University. DOI: https://doi.org/10.13140/RG.2.2.15450.16329

Simpson, O. (2013). Student retention in distance education: are we failing our students? *Open Learning: The Journal of Open, Distance and e-Learning*, *28*(2), 105–119. DOI: https://doi.org/10.1080/02680513.2013.847363

Sorva, J. (2010, October). Reflections on threshold concepts in computer programming and beyond. *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (pp. 21–30). ACM. DOI: https://doi.org/10.1145/1930464.1930467

Sullivan, G. M., & **Feinn, R.** (2012). Using effect size—or why the P value is not enough. *Journal of graduate medical education*, 4(3), 279–282. DOI: https://doi.org/10.4300/JGME-D-12-00156.1

Townsend, L., Hofer, A. R., Hanick, S. L., & **Brunetti, K.** (2016). Identifying threshold concepts for information literacy: A Delphi study. *Communications in Information Literacy*, *10*(1), 23. DOI: https://doi.org/10.15760/comminfolit.2016.10.1.13

Vagianou, E. (2006, February). Program working storage: a beginner's model. *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006* (pp. 69–76). DOI: https://doi.org/10.1145/1315803.1315816

TO CITE THIS ARTICLE:
Adamiak, J. P. (2024). Applying Threshold Concepts Strategies to Teaching Computing Students in an ODL Context. *Open Praxis*, *16*(2), pp. 225–236. DOI: https://doi.org/10.55982/openpraxis.16.2.552

Submitted: 06 April 2023
Accepted: 05 February 2024
Published: 03 April 2024

Open Praxis is a peer-reviewed open access journal published by International Council for Open and Distance Education.

INTERNATIONAL COUNCIL FOR OPEN AND DISTANCE EDUCATION