

Promoting Algorithmic Thinking in an Introductory Programming Course

<https://doi.org/10.3991/ijet.v14i01.9061>

Sohail Iqbal Malik (✉), Mohanaad Shakir, Abdalla Eldow,
Mohammed Waseem Ashfaque
Al Buraimi University College, Al Buraimi, Oman
Sohail@buc.edu.om

Abstract—Computer programming is considered as a difficult area of study for novices. One of the reasons is the main focus of the curriculum presented in an introductory programming (IP) course which emphasizes more on the programming knowledge (syntax and semantic) of the programming language. This study introduced a new teaching curriculum in the IP course, which focuses on different skills required by the novices. We compared the IP course materials based on the traditional and new approaches against five categories. The result shows that the new approach encourages both the programming knowledge and problem solving strategies, and promotes deep learning. Furthermore, it discourages programming shortcut (Problem statement → Code), and support three-step approach (Problem statement → Solution Plans → Code) in solving a problem statement. The new approach also promotes algorithmic thinking in the IP course by paying equal attention on the problem solving strategies.

Keywords—Algorithmic thinking, Novice programmer, Problem solving strategies, Introductory programming

1 Introduction

The introductory programming course is considered as one of the difficult courses for novice programmers [1]. They have to focus on different skills, such as program comprehension, problem solving strategies, program design etc., at the same time. On the other side, the traditional teaching approach used in the IP course focuses more on the syntax of the programming language [2]. This study evaluated the teaching materials of the IP course based on the traditional approach. A new teaching approach based on the ADRI (Approach, Deployment, Result, Improvement) model was introduced in the IP course to address the shortcomings emerged from the traditional approach. To determine the impact of the new teaching approach, the teaching materials based on the traditional and new approaches were analyzed by the criteria proposed by [3].

The new teaching approach pays equal attention on algorithmic thinking skills besides programming knowledge (syntax and semantics) in the IP course. All the programming questions or examples presented in the IP course cover algorithmic thinking skills. Flowchart and pseudo code techniques are used to promote algorithmic

thinking skills. Novices use these techniques (Flowchart and pseudo code) to purpose a solution for the given problem statement.

This paper is organized into a number of sections. It starts with a literature review followed by a brief introduction to the new teaching approach. Research methodology and Results of the study were presented. The paper concludes with the research outcomes of the study.

2 Literature Review

Various studies have been conducted to investigate the different teaching approaches used in IP courses. In this section, we explore the literature and discuss different teaching approaches.

The teaching method is an approach by which educators deliver teaching materials to the students. A teaching strategy is one of the crucial factors that help to motivate students and ultimately play an important role in mastering the course content [4]. Students' achievement of the learning outcomes is affected by their learning style and motivation. Some students prefer a visual learning style, others prefer auditory or kinesthetic styles [5] and therefore it is important for teachers to incorporate different approaches to accommodate different learning styles in their teaching process.

Reference [6] interviewed instructors of an IP course to learn about the pedagogical and motivational strategies they used in the course for the students. They interviewed 18 instructors and concluded that 'student-teacher relationship, teachers' proximity with the students, class competitions, challenges and strategies in which the students are engaged actively and continuous assessment' (p. 7) are considerable motivational strategies. Moreover, most teachers' do not consider this course different from other courses and use the same traditional pedagogical strategies in it. Some teachers' emphasized that a good choice of exercises and teaching materials also play an important role in a successful teaching and learning process.

Reference [7] discussed that if instructors omit any one of these models ('data structure, program design, problem domain and data representation') from the teaching process of the programming then as a consequence 'the students will make up their own models of dubious quality' (p. 21).

Reference [8] introduced Assurance of Learning (AOL) process in an introductory programming (IP) course to align program-learning goals with the IP curriculum. The five stages of the AOL process helped them to identify shortcomings in the course and provided mechanism to address these drawbacks. They collected faculty feedback regarding the AOL process through a web-based survey. The overwhelming majority of the participants agreed that the AOL affected their teaching process in a positive way.

Mobile device supported learning provides additional support for accessing introductory programming teaching materials to students [9]. It provides teaching materials anywhere, anytime and eliminates space and time constraints for learners. It also allows interaction between teachers and learners without physical contact [10]. The work of [11] suggested that the retention of specific subject knowledge in first year

university students for a mobile learning group was better than for those who were not supported in this way. Surveys conducted with novice programmers concluded that the overwhelming majority of participants agreed that mobile device supported learning provides additional support to access and understand teaching materials. The teaching materials can be saved in mobile memory, which can be accessed at the learners’ convenience. Furthermore, the students can access recorded class lectures along with other teaching contents. It can help them to catch-up with missed classes [9]. Moreover, it helps to promote information communication technology as a teaching tool in the teaching and learning process [12] [13].

The ACM-IEEE Joint Task Force on Computing Curricula [14] identifies 18 knowledge areas for computing undergraduate programs. Programming knowledge (syntax and semantics) and problem solving strategies are given equal attention in these knowledge areas. The IP curriculum should promote both skills so that students will be well prepared for their career path and labor market. Reference [15] reported that curriculum organization and teaching methods were the two main factors for high failure rates in IP courses. Reference [16] analyzed forty programming books and concluded that most of the books emphasized the syntax of programming languages. Some authors provide a large number of examples but give less attention to problem solving skills. In some cases, authors discuss problem-solving skills in early chapters but these skills are not integrated in the remaining chapters.

Table 1 lists the factors and current practices used in teaching and learning process of an IP course. In this study, these factors were addressed by introducing a new teaching approach instead of the traditional approach to enhance the learning process.

The traditional approach emphasizes more on programming knowledge (syntax and semantics), promotes surface learning and programming shortcut (Problem Statement→Codes). Most of the programming editors used in the introductory programming also promote programming shortcut.

Table 1. Factors and practices in traditional approach

Factors	Practices in Traditional Approach
Learning to program	More emphasis is given on syntax and semantics of programming languages
Teaching strategy/model	Traditional approach spends more time on teaching syntax compared to problem solving strategies. Hence does not achieve students’ learning outcomes and high failure and dropout rates are reported
Programming books	Most textbooks discuss problem solving strategies in an initial chapter only and syntax and semantics are emphasized in rest of the chapters
Learning style	Traditional approach promotes surface learning instead of deep learning because problem solving strategies are not emphasized thoroughly in all topics
Lecture notes	Examples discussed in the lectures promote programming shortcut: (Problem Statement→Codes)
Lab exercises	Problems given in the exercises promote programming Shortcut: (Problem Statement→Codes)
Software development tools	Most of the editors used in the introductory programming promote programming shortcut: (Problem Statement→Codes)
Programming language practice	Most students struggle with completing programming exercises because they spend most of their time in debugging their programs

3 New Teaching Approach

The new teaching approach based on the four stages of the ADRI model was introduced in the IP course. The ADRI approach focuses on different skills (problem solving strategies, syntax, semantics and problem analysis) in the IP course. All the programming examples and problems were prepared based on the four stages of the ADRI approach as shown in table 2. The first part of the ADRI based example provides a problem statement. The second part (Approach) promotes algorithmic thinking by using problem solving skills such as pseudo code and flow chart. The third part (Deployment) focuses on the syntax and semantics of the programming language. The fourth part (Result) covers input, output, process used to solve the problem statement, and basic syntax and semantic errors in the programming language. The fifth part (Improvement) focuses on new programming constructs, syntax and semantics of the programming language by changing the requirements of the problem statement given in the first part.

Table 2. ADRI based programming example

Design an algorithm that will read name, balance and rate then calculate interest and print name and interest.	
Step 1: Approach – Problem-solving strategies Solve the given problem statement by pseudo code technique	Solve the given problem statement by flow chart technique
Step 2: Deployment – Programming knowledge Convert the proposed solution in step 1 into a computer program by using the programming language	
Step 3: Results Input: NAME : Naushad BALANCE: 20 RATE : 0.25 Process: INTEREST = 20 x 0.25 Output: Name: Naushad Interest: 5	Syntax Error: Missing Semicolon Example: 4 int feet; 5 int inches 6 double bigNum; Reason: Semicolon is expected on line 5
Step 4: Improvement Update the above algorithm to calculate RATE if INTEREST and BALANCE is entered by the user.	

An ADRI editor was prepared to support the four steps of the ADRI approach as shown in figure 1. The editor provides separate interface for each step of the ADRI approach. The editor was developed in Java program. It is compatible with most of the operating system and easy to use. All the programming questions are embedded in it. The user can choose question by topic.

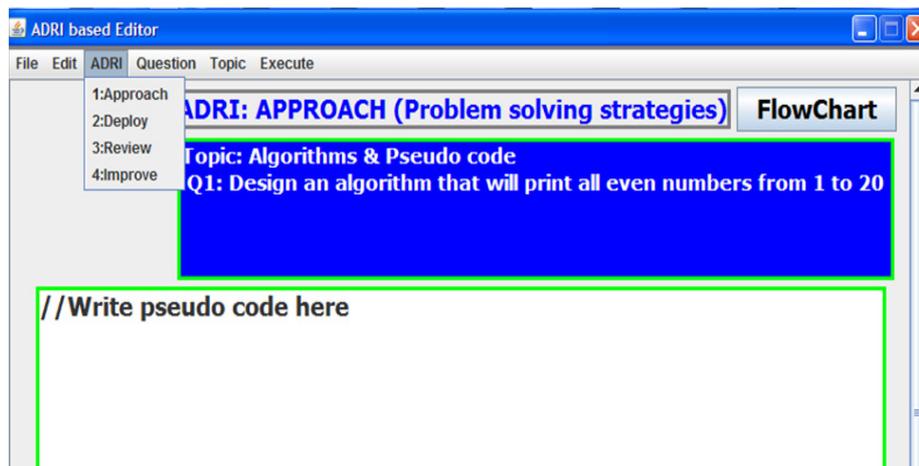


Fig. 1. ADRI based Editor [17]

4 Research Questions

It is evident from the literature that the teaching materials based on the traditional approach do not provide all the required skills to novices. This study purposes the new teaching materials based on the four stages of the ADRI approach in the IP course. Moreover, the study purposes the following research questions to determine the impact on the teaching materials based on the ADRI approach compared to the traditional approach.

RQ1. What is the impact of the ADRI approach compared to the traditional approach on the teaching materials of the introductory programming course?

RQ2. How the ADRI approach promotes algorithmic thinking in the introductory programming course?

5 Research Methodology

The research questions were probed by comparing the IP course materials based on the traditional and ADRI approaches against five categories such as Teaching topics for lecture sessions, Practical topics for lab sessions, Types of programming examples and problems, Presentation style of example and problems, and Four problem solving steps in examples and problems.

6 Results

This section describes the results of this research study and probes research questions 1 and 2. The first research question is as follows:

RQ1. What is the impact of the ADRI approach compared to the traditional approach on the teaching materials of the introductory programming course?

We compare the teaching materials based on the traditional and ADRI approaches to determine the impact of the ADRI approach on the course.

Table 3 below shows the comparison between the traditional and ADRI based teaching materials of the course. The teaching materials are compared in five different categories including: ‘Teaching topics for lecture sessions’, ‘Practical topics for lab sessions’, ‘Types of programming examples and problems’, ‘Presentation style of example and problems’, and ‘Four problem solving steps in examples and problems’. The comparison of the traditional and ADRI approaches in the first two categories (‘Teaching topics for lecture sessions’, ‘Practical topics for lab sessions’) are discussed in terms of the emphasis on the programming skills and problem solving strategies in each category of teaching materials. The remaining three categories (Types of programming examples and problems, Presentation style of example and problems, and Four problem solving steps in examples and problems) were analyzed on the criteria provided by [3].

Reference [3] presented a content analysis of computer programming examples or problems of sixteen textbooks taught in high school in Taiwan. They categorized the programming problems into five different types such as math problem (finding the even integers), syntax-oriented problems (demonstrating the different programming constructs), real-life problems (how to convert Fahrenheit and Celsius temperatures, calculation of bank loans), graphics problems, and others. Graphic problems are not taught in the IP course and so were not considered in our analysis.

Reference [3] discussed a qualitative analysis of three different presentation styles used for the programming examples in textbooks. A brief explanation of these presentation styles as follows:

Problem Statement → Pseudo-code or Flowchart: The problem statement is pursued by pseudo-code or flowchart

Problem Statement → Codes: The problem statement is pursued by code with a brief explanation.

Problem Statement → Solution Plan → Coding: The problem statement is pursued by solution plan (problem analysis, input, output, process, program design, variables etc.) and then complete code is given with a brief explanation.

Reference [3] demonstrated problem solving as a 4-step process as follows:

The problem analysis step: The problem statement is described so that programmers can understand it. There is no particular method used for this step.

The solution-planning step: It is demonstrated by a textual description such as algorithm design, variables, main algorithm etc.

The coding step: A complete code is given with a brief explanation or comments. A program is also provided with a sample run.

The testing/debugging step: It shows to students how to debug syntax, semantic and run-time errors by presenting programs with planted errors in it.

Table 3. Comparison of the traditional and ADRI based teaching materials

No	Area of concern	Traditional approach		ADRI approach	
1.	Teaching topics for lecture sessions	More emphasis on syntax and semantics compared to problem solving strategies		Pay equal attention on syntax & semantics and problem solving strategies	
		Syntax & Semantics	Problem Solving Strategies	Syntax & Semantics	Problem Solving Strategies
		86.6%	13.4%	100%	100%
2.	Practical topics for lab sessions	More emphasis on syntax and semantics compared to problem solving strategies		Pay equal attention on syntax & semantics and problem solving strategies	
		Syntax & Semantics	Problem Solving Strategies	Syntax & Semantics	Problem Solving Strategies
		86.6%	13.4%	100%	100%
3.	Types of programming examples and problems	Syntax-oriented examples and problems are dominant in reading materials (50%) and lecture notes (49%)		Math (62.7%) and daily-life (25.6%) examples and problems are dominant	
4.	Presentation style of example and problems	Problem Statement→Codes Presentation style is dominant in lecture notes (77%) and reading materials (70%)		Problem Statement→Solution Plans→Codes Presentation style is dominant in lectures (83.3%) and labs (84%)	
5.	Four problem solving steps in examples and problems	Coding step is dominant in <i>lecture notes</i> (74%) and <i>reading materials</i> (61%)		All the four problem solving steps are addressed	

It is clear from table 3 that in ‘Teaching topics for lecture sessions’ category, the ADRI approach pays equal attention to syntax & semantics and algorithmic thinking skills (problem solving) compared to the traditional approach which only emphasizes the syntax & semantics. The same trend is depicted in ‘Practical topics for lab sessions’ category. For ‘Types of programming examples and problems’ category, syntax-oriented examples are more dominant in the traditional approach compared to math and daily-life which are dominant in the ADRI approach. Problem Statement →Codes style is dominated in the traditional approach. Under ‘Presentation style of example and problems’ category compared to the ADRI approach, Problem Statement→Solution Plans→Codes style is dominated. Lastly in ‘Four problem solving steps in examples and problems’ category, coding is dominated in the traditional approach compared to the ADRI approach where all the four problem solving steps are dominated. Although the change in emphasis from coding dominating to a more equal presentation of problem solving and coding was the aim of the redesign and modification of the teaching materials, this comparison demonstrates that the goal was successfully achieved.

6.1 ADRI approach promotes algorithmic thinking

This section describes the result for research question 2. The second research question is as follows:

RQ2. How the ADRI approach promotes algorithmic thinking in the introductory programming course?

As discussed above in table 2, the step 1 (Approach stage) of the ADRI approach uses two different problem solving skills (pseudo code and flowchart) to purpose a solution for the given problem statement. All the programming examples or questions presented in the IP course based on the ADRI approach includes the step 1, which helps novices to focus on algorithmic thinking skills throughout the IP course.

The results depicted for ‘Teaching topics for lecture sessions’ and ‘Practical topics for lab session’ categories in table 3 that the ADRI approach pays equal attention on algorithmic thinking skills and programming knowledge (syntax and semantics). Moreover, the ADRI approach discourages programming shortcut (Problem Statement→Codes) and promotes the three-step approach (Problem Statement→Solution Plans→Codes) which encourages algorithmic thinking.

7 Discussion

The IP teaching materials based on the traditional and ADRI approaches were analyzed. The comparison shows that the ADRI approach enhanced the students’ knowledge and confidence in the programming domain. They realized and grasped the programming concepts in a better context, which helps them in developing their programming skills. The multi-stage approach to develop a solution helped students in developing their flow of logic in a structured way and to achieve their objectives in the course. This finding is consistent with [18].

In the ADRI based teaching materials, math and daily-life programming examples and problems are dominant which helped in developing the students’ interest in the course. The students already had a good knowledge of maths from their school and current courses so it was comparatively easy for them to solve the problem statement based on it. Moreover, it was easy for the students to understand the requirements of the problem statements based on daily-life scenarios. Syntax-oriented type of programming examples and problems are dominant in the traditional approach, which gives good understanding of the programming knowledge to the students, but it does not promote problem-solving skills. On the other side, computer science students considered problem solving as one of the important skills required for a computer professional [19].

The traditional approach style (Problem Statement →Codes) promotes programming shortcuts. However, for novices it is very important to practice program design as suggested in some research studies [7][14][16][20]. The ADRI approach discourages programming shortcut and promotes the three-step approach (Problem Statement→Solution Plans→ Coding) which develops program design, language features and program comprehension skills. The purpose of the whole process is to promote deep learning of the programming domain to the novices. Moreover, ‘Solution Plans’

step of the three-step approach promotes algorithmic thinking skills. The three-step approach offered by ADRI is consistent with that suggested by [21]-[25].

The step 3 (Result) of the ADRI approach covers input, output, process used to solve the problem statement, and basic syntax and semantic errors in the programming language. This process helps novices in understanding the given problem statement in a better way and promotes deep learning. Common programming errors are presented to novices, which help them in writing the code of the program with less errors and saving their debugging time. Furthermore, novices can focus more on the problem statement.

8 Conclusion

The introductory programming (IP) course is considered as one of the difficult courses for novice programmers. The teaching materials used in the IP course focuses more on programming knowledge (syntax and semantic) of the programming language. Moreover, it promotes surface learning, and programming shortcut. This study re-designed the IP teaching materials based on the four stages of the ADRI model. The ADRI editor was prepared to support the four stages of the ADRI approach in the IP course.

The IP teaching materials based on the traditional and ADRI approaches were compared against five categories such as Teaching topics for lecture sessions, Practical topics for lab sessions, Types of programming examples and problems, Presentation style of example and problems, and Four problem solving steps in examples and problems. The comparison shows that the ADRI approach promotes both the programming knowledge (syntax and semantics) and algorithmic thinking skills (problem solving strategies) compared to the traditional approach who emphasizes more on programming knowledge. The traditional approach promotes programming shortcut presentation style (problem statement \rightarrow Codes) whereas the ADRI approach reduces it by following presentation style (Problem Statement \rightarrow Solution Plans \rightarrow Codes) which promotes deep learning of the programming domain.

9 References

- [1] I.M. Sohail, and C.N. Jo, "Impact of a new teaching and learning approach in an introductory programming course, *Journal of Educational Computing Research*, SAGE, vol. 55, No. 6, pp. 789-819, 2017 <https://doi.org/10.1177/0735633116685852>
- [2] S. Iqbal and O.K. Harsh, "A self-review and external review model for teaching and assessing novice programmers", *International Journal of Information and Education Technology*, Vol. 3, No. 2, 2013, pp. 120-123 <https://doi.org/10.7763/IJET.2013.V3.247>
- [3] C.C. Wu, J.M.C. Lin, and K.Y. Lin, "A content analysis of programming examples in high school computer textbooks in Taiwan", *Journal of Computers in Mathematics and Science Teaching*, Vol. 18, No. 3, 1999, pp. 225-244
- [4] S. Mohorovicic, and V. Strcic, "An overview of computer programming teaching methods", *Proceedings of Central European Conference on Information and Intelligent Systems*, CECIIS, Croatia, 2011, pp. 47-52

- [5] Hu, “A case study in teaching adult students computer programming”, Proceedings of the NACCQ conference, NZ, 2003, pp. 287-290
- [6] A. Gomes, and A. Mendes, “A teacher’s view about introductory programming teaching and learning: difficulties, strategies and motivations”, proceedings of IEEE Frontier in Education (FIE), Madrid, 22-25 Oct, 2014, 2014, pp. 1-8
- [7] L.E.Winslow, “Programming pedagogy—a psychological overview”, ACM SIGCSE Bulletin, Vol. 28, No. 3, 1996, pp.17-22. <https://doi.org/10.1145/234867.234872>
- [8] I. Sohail, M. Roy, C. Morshed, and C.N. Jo, “Impact of Assurance of Learning in programming course for novices”, Proceedings of the 2nd British Computer Society (BCS) International IT Conference, Abu Dhabi, UAE. 2014, DRO link: <http://dro.deakin.edu.au/view/DU:30073423>
- [9] S. Iqbal, M. Chowdhury, and O.K. Harsh, “Mobile devices supported learning for novice programmers”, Proceeding of the 2nd international conference on E-learning and E-technologies in education, IEEE, Poland, 2013, pp. 277-282
- [10] F.J. Tapia-Moreno, H.A. Villa-Martinez, C.A. Lopez-Miranda “Elaboration of statistics learning objects for mobile devices”, International Journal of Interactive Mobile Technologies, 2012, Vol. 6, No. 2.
- [11] D. Mcphee, P. Thomas, M. Ware, and N. Thomas, “An investigation into the effectiveness of the use of m-learning in the teaching of multimedia to first year university students”, International Journal of Emerging Technologies in Learning, 2006, Vol. 1, No. 1.
- [12] E. Abdalla, S. Mohanaad, Y.A. Mohamad, I.M. Sohail, “Using Information Communication Technology as a Teaching tool in Sudanese Governmental Universities of Khartoum State”, International Journal on Informatics Visualization, 2017, Vol. 1, No. 4, Pages. 150-156
- [13] P. Gurunathan, M. John, I. Sohail, “Bringing Technology into the classroom: Receptivity and Satisfaction level of faculty members Using Smart board – An Experimental Study” In proceeding of ICAICT 2014, Elsevier, pp. 274-279.
- [14] ACM-IEEE Joint Task Force on Computing Curricula 2013, Computer Science Curricula 2013, ACM Press and IEEE Computer Society press <http://dx.doi.org/10.1145/2534860>.
- [15] J. Tavares, I. Brzezinski, I. Huet, A. Cabreal, and D. Neri, “Having coffee with professors and students to talk about higher education pedagogy and academic success”, Proceedings of the 24th International HERDSA conference, 2001, Newcastle, Australia
- [16] M. De Raadt, M. Toleman, and R. Watson, , Textbooks Under Inspection, University of Southern Queensland, Australia, 2005, retrieved May 2013, <http://eprints.usq.edu.au/167/1/TechReport_Draft_10.pdf>
- [17] I.S. Malik, “Enhancing practice and achievement in introductory programming using an ADRI editor”, In Proceedings of the IEEE International Conference on Teaching, Assessment and Learning for Engineering, 2016, pp. 32-39, IEEE, Thailand, 7-9 December, <https://doi.org/10.1109/TALE.2016.7851766>
- [18] I.S. Malik, and J. Coldwell-Neilson, “Comparison of traditional and ADRI based teaching approaches in an introductory programming course”. Journal of Information Technology Education: Research, 16, pp. 267-283. 2017, <http://www.informingscience.org/Publications/3793> <https://doi.org/10.28945/3793>
- [19] I.M. Sohail, and A.E. Mustafa, “Social factors influence on career choices for female computer science students”. International Journal of Emerging Technologies in Learning, 2018, vol. 13, issue 5, pp. 56-70
- [20] T. Jenkins, “Teaching Programming – A Journey from Teacher to Motivator”, Proceedings of 2nd Annual LTSN-ICS Conference, London, 2002, pp.65-71.

- [21] I.S. Malik, “Role of ADRI model in teaching and assessing novice programmers”, PhD thesis, Deakin University, 2016, <http://hdl.handle.net/10536/DRO/DU:30088862>
- [22] M. Webster, Overview of programming and problem solving, Merriam-Webster’s Collegiate Dictionary, Tenth Edition, 1994, retrieved on July15, 2013: computer-science.jbpub.com/vbnet/pdfs/mcmillan01.pdf
- [23] I.M. Sohail, and C.N. Jo, “A model for teaching an introductory programming course using ADRI”, *Education and Information Technologies*, Springer, vol. 22, issue 3, pp. 1089–1120, <https://doi.org/10.1007/s10639-016-9474-0>
- [24] I.S. Malik, “Improvements in introductory programming course: action research insights and outcomes”, *Systemic Practice and Action Research*, DOI: 10.1007/s11213-018-9446-y, 2018, Springer <https://doi.org/10.1007/s11213-018-9446-y>
- [25] I.M. Sohail, and C.N. Jo, “Gender differences in an introductory programming course: New teaching approach, students’ learning outcomes, and perceptions”, *Education and Information Technologies*, Springer, 2018 <https://doi.org/10.1007/s10639-018-9725-3>

10 Authors

Sohail Iqbal Malik obtained his Ph.D. in Computer Science from Deakin University, Australia in 2016. He has been working as an Assistant Professor at Information Technology Department, Buraimi University College, Oman (Academic collaboration with California State University, Northridge, USA) since February 2007. His research interest includes Algorithmic Thinking, Computer Education, Technology in Education, and Mobile Learning.

Mohanaad Shakir is an Assistant Professor at Al-Buraimi University College, Oman. He holds a B.S.C Degree in Computer Science from the University of Al-mamoon, Baghdad, Iraq; Post Diploma in Computer Security from University of Technology, Iraq and M.Sc. in Information Technology (MIT) from the University of Tenaga National (UNITEN), Putrajaya, Malaysia. He is a Ph.D. candidate in Information Communication Technology at the University of Tenaga National (UNITEN), Putrajaya, Malaysia. His research interests include Cipher Security, Data classification, Computer-Aided Learning and Cloud Computing Security. He can be contacted at mohanaad@buc.edu.com

Dr. Abdalla Eldaw is an academic staff member at Buraimi University College (BUC) – Oman. He is a member of research committee in this college and he was selected to be an evaluator member for last BUC conference. Also, he is a computer security specialist and university curriculum expertise.

Mohammed Waseem Ashfaque completed his graduation and post-Graduation in Computer Science. He has been working as a lecturer in department of IT at Buraimi university college (BUC), Al-Buraimi, Oman since 2015. He has contributed in many research articles.

Article submitted 15 June 2018. Resubmitted 17 July 2018. Final acceptance 05 August 2018. Final version published as submitted by the authors.