

Pedagogical Approach to Effective Cybersecurity Teaching

Abdullahi Arabo and Martin Serpell

Department of Computer Science and Creative Technologies, University of the West of England, Coldharbour Lane, Bristol, BS16 1QY
Abdullahi.Arabo@uwe.ac.uk

Abstract. Initial research ruled out many factors that were thought may correlate to student academic performance. Finally, a strong correlation was found between their academic performance and their motivation. Following on from this research teaching practice was restructured to improve student motivation, engagement, and interest in cybersecurity by contextualizing teaching material with current real-world scenarios. This restructuring led to a very significant improvement in student academic performance, engagement and interest in cybersecurity. Students were found to attend more of their lectures and practical sessions and that this had a strong positive correlation with their academic performance.

Keywords: Education, Cybersecurity

1 Introduction

Learning as a concept can mean different things to different people. Some might see it as a way of exploring a giving fact and driving a new meaning or understanding of an existing theory or fact. While others might just see it as an abstract form of knowing or memorizing an existing fact without the need to analyze or re-interpreted that fact into a body of knowledge. Learning can also be different depending on the size of the class and facilities used to convey such a message to the audience.

The need for careful and deliberate considerations in designing and implementing educational and training activities to enhance cybersecurity teaching and adaptability is of paramount importance, so as to make it more of an interactive, engaging and exciting discipline. The current skills gap in the field of cybersecurity is growing. Some employers have to wait for up to six months to the recruit suitable candidate for a role. Therefore; there is a need for a change in the pedagogical approach to teaching cybersecurity so as to provide the necessary technical skills to reduce the skills gap. The current practice makes use of some available tools and textbook oriented teaching methodologies.

In this paper, we explored the use of real-world and live scenarios with aid of available tools to contextual learning and teaching in cybersecurity so as to provide the concept of students as co-creators and higher learners rather

than just presenting theoretical and textbook materials with some out of date scenarios.

This paper has drawn its data from a various practical approaches that have been tested with three sets of cohorts, where teaching materials are contextualized with real-world scenarios, judging the engagement of students within these cohorts in terms of their active engagement within and outside lectures with the teaching materials and teaching team. At the end of each cohort statistical data has been kept and evaluated to compare the end result both in terms of technical cybersecurity capability, obtaining placements/full-time work after graduation and final marks for taught cybersecurity and programming technical modules.

The rest of the paper is structured as follows: related background work in terms of teaching theories, pedagogy used in industrial controls systems and the use of available tools are presented. We then addressed the issues of technical programming skills and use of real-world examples to contextualize learning and its effect on enhancing teaching and learning in cybersecurity. The paper further presents our finding from a different set of cohort and compared this to available data when such methodologies have not been used. Finally, the paper then concludes with our concluding remarks and future study to enhance and provide a better road map and pedagogy for enhanced technical skills and students engagement for teaching cybersecurity.

2 Background

Ramsden (1992) classifies the concepts of learning into three main theories: teaching as a means of transmission, teaching as student activity and teaching as making learning possible [1]. The basic concept of the first theory lies in the fact that, the teacher is the main source of authoritative content/body of knowledge, while the student role is just a positive receiver of such knowledge; without the need of analyzing it. The theory utilize the input-output model without any processing in-between. It fosters a surface learning approach as explained by Biggs [2]. The theory is also of the opinion that there are good and bad students. If a student fails, it is entirely their fault but nothing to do with the teacher, programme or the institution. Biggs (1999) also discusses two concepts of constructive alignment: students get meaning from what they do to learn and teachers align such derivatives to planned learning activities with the expected learning outcomes [2]. Hence, constructive alignment is all about trying to associate assessment and outcomes. Ramsden's second theory was the principle that teaching is seen as a student activity [1]. The theory emphasis is on the need to establish procedural rules by improving teaching via the use of a set of strategies; the content not being important whereas the conditions of learning are. Ramsden's third theory focuses on the combination of the first two theories [1]. Saying that teaching is a process of making learning possible; hence, teaching, students, and subjects are inter-linked. Each takes their own responsibilities and work in harmony to enhance and provide a more positive environment. The major principle of this theory is that a teacher needs to en-

gage with and get involved with their students by; encouraging students interest, demonstrating concern and respect for students, providing appropriate feedback, presenting students with clear goals, giving students independence and modifying ones teaching practice in response to student learning outcomes.

Development of reflective thinking and practice needs to be a cornerstone of each discipline and part of the pedagogy of teaching cybersecurity to students. This is, however, not an area of learning and teaching which is regularly discussed, despite being recognized across the literature as complex and challenging for lecturers for a variety of reasons teaching reflective practice is neither obvious or easy [3]. The range of reflective approaches available is extensive for example the work of [4], [5], [6], [7], [8], [9], [10], [11] and [16]. The challenges of introducing reflective practice are varied. For example, the literature suggests that theory is needed to underpin learning as without a guiding theory/structure, student reflections may remain shallow, not providing the anticipated depth of learning [12]. However, use of simplified and technicity prescriptions in the implementation of a theory or structure can mean that reflection becomes a checklist ticking or recipe following exercise [13]. There is potential tension between reflection as a personal, individual experience but also one that is undertaken to be assessed [14]. The risk of gamification and emotional performativity in reflective assessment writing can arise with more sophisticated students [15]. Most resonant is Wongs (2016) experience that despite the introduction of models and examples and provision of writing examples and templates, it never quite works for them [17] and that many students still write descriptively rather than reflectively.

Cybersecurity students need to be proficient in computer programming. Much research has already been done on how to teach computer programming. A tremendous variation in the ability to learn computer programming was reported by Bishop-Clark as long ago as 1995 [27]. Ranjeeth and Naidoo (2007) used of statistical analysis of students performances in computer programming related assessments tasks to conclude the same [28]. However it is believed that an aptitude for computer programming is closely related to an aptitude for mathematics [29].

Student motivation has been found to play a large part in students learning programming [20] [21]. Learning how to motivate students will improve teacher effectiveness and student learning [22]. The motivating factors individual attitude and expectation, clear direction and reward and recognition have been identified [23] as have the ability to make decisions that positively affect the quality of your work and the perception that your work is interesting and challenging [24]. However, it may not be easy measuring the motivation of new students rather it is something that must be instilled by the teacher; this aligns with the theories put forward by Ramsden (1992) [1].

Statistical analysis by Turley and Bieman (1995) found that exceptional computer programmers were more likely to be people that saw the big picture, had a bias for action, were driven by a sense of mission, exhibited and articulated strong convictions, played a proactive role with management, and helped other programmers [18]. They also found that programming ability improved with ex-

perience. Colley et al (1996) interviewed computer programmers and found that enjoying working with machines, liking technology and enjoying solving complex problems were linked to programming ability whereas being good at maths and being good at science were not [19].

3 Teaching Methodology

3.1 Introducing Computing and Cybersecurity

In their first year, students were introduced to computer architecture and cybersecurity via lectures supported by formative self-assessment tests. There were two pieces of programming coursework which were designed to be very different. As many of the students had never programmed before, the first piece of coursework had the students doing a lot of repetitive programming under the guidance of their module tutors. They were required to write a computer program in the C programming language to emulate the instruction set of an 8-bit microprocessor. The objective was to get students familiar with both computer programming and computer architecture. This piece of coursework was designed to be grueling whereas the next was designed to be fun. The second piece of programming coursework had the students working on an open problem. They had to write software that controlled a battleship that they could see on a large screen. Their battleship bot was in competition with their fellow students bots, marks were allocated based on how many of their fellow student's battleship bots that they had sunk. This piece of coursework was aimed at developing students networking and problem-solving skills. To stretch the better students they were encouraged to think of creative solutions to this task. Students were not punished for using underhanded techniques such as man-in-the-middle or denial-of-service attacks against their fellow students but rewarded with bonus marks instead.

Student performance in both pieces of coursework was recorded along with their responses to a questionnaire, 76 students took part. The questionnaire contained questions relating to students programming experience, technology experience, problem-solving experience, work plans and programming attitude.

- Programming experience
 - Number of years programming at school?
 - Number of years programming at work?
- Technology experience
 - Do you like Science Fiction [Y/N]?
 - Do you run Linux at home [Y/N]?
 - Do you own a Raspberry Pi or other development board [Y/N]?
 - Have you built a web-site before [Y/N]?
 - Have you installed an Operating System before [Y/N]?
 - Have you built a PC from component parts before [Y/N]?
- Problem solving experience
 - Do you like solving puzzles e.g. Sudoku [Y/N]?
 - Are you good at, or have ever been good at, chess [Y/N]?

- Work plans
 - Do you intend to go on a placement year [Y/N]?
 - Do you have a specific career path in mind [Y/N]?
 - Do you have a part-time job [Y/N]?
 - Do you want to work in a technical role when you leave university [Y/N]?
- Programming attitude
 - How much do you enjoy computer programming (Please answer 0 to 10 where 0=not at all, 10=very much)?
 - How many hours a week did you spend doing your battleship bots assignment?

Similarly student performance in both pieces of coursework was recorded against their aptitude test results. A number of students volunteered to take the following aptitude tests; numeric (30), verbal (28) and non-verbal (25). The lessons learnt, from the student survey and aptitude tests, were then fed back into their teaching in following years.

3.2 Cybersecurity In-depth

Ramsden's (1992) third theory is applied to teaching cybersecurity in-depth. Teachers make use of a well-established framework rather than relying on an authoritative body of knowledge or an encyclopedic transmission of information to students. This theory works better if the teacher is able to be dynamic, reflective, make use of the wealth of experience from students and link the concept of teaching, student and subject by applying to real world and current examples to help students conceptualize the principles and how it can be applied in the world of work (WOW), we refer to this as a WOW factor. Being dynamic is part of the framework of teaching, using these principles helps students in understanding ideas and linking of ideas to form meaning wholes rather than just as a surface of possible facts and concepts. It also makes and identifies the big ideas that structured the course, as this might not be in any one part of the lectures or seminars or practicals. By way of linking this concept from session to session and pointing out the possible applications in life, it stimulates discussion and engagement with students in both large and small groups. Coursework marks and student attendance before and after the introduction of the new WOW teaching framework were recorded for later analysis.

vbensk et al [25], has pointed out adversary thinking as an essential skill that is required for cybersecurity experts. These is enable them to understand possible cyber-attacks as well as to set up effective defences. Their study is based on a set of practical classroom scenarios that enables participants to cope with numerous interdisciplinary tasks throughout the semester while at the same time exercising a broad spectrum of technical and soft skills such as: system administration, penetration testing, game design, teamwork, project planning, communication, and presentation.

Rob Byrd [26], has also conducted a study that indicates/clarifies the importance of certain math topics as a core set of skills that is essential or crucial for

the developing and can be helpful for a program of study in cybersecurity which will eventually that will enable such individual with this skill set to succeed in the profession and evolving cybersecurity environment.

4 Results

4.1 Introducing Computing and Cybersecurity in the First Year

Student Prior Programming Experience It was reasonable to believe that those students with prior programming experience, either while at school or at work, would outperform the other students in their programming coursework. However, tests using Person Correlation showed that there was no significant correlation between the experience of programming at school or work and programming performance in their coursework, see Table 1. This was unexpected and is most likely explained by the students without prior experience quickly catching up with those that had prior experience.

Question	Test Result	8-bit Emulator	Battleship Bots
Number of Years	Pearson Correlation	0.170	0.142
Programming at School	Sig. (2-tailed)	0.131	0.258
Number of Years	Pearson Correlation	0.216	0.060
Programming at Work	Sig. (2-tailed)	0.100	0.391

Table 1. Correlation between student prior programming experience and their first year coursework performance

Student Prior Technology Experience Certain pass-times, like running Linux at home, liking science fiction, owning a development board, building websites, installing operating systems and building PCs can be indicative of a type of person who enjoys handling technology. This type of student might be expected to do well in programming assignments, however, the Mann-Whitney test, see Table 2, showed that none of these activities correlated to better performance in either piece of coursework.

Student Prior Problem Solving Experience Solving puzzles and playing chess involve the ability to apply concentration. The results of the Mann-Whitney Tests were very interesting, see Table 3. They showed that a like for solving puzzles did not help students with their first piece of coursework ($U = 541, p = 0.914$) but it was very helpful with their second piece of coursework ($U = 309.5, p = 0.004$) and similarly being good at Chess did not help students with their first piece of coursework ($U = 595, p = 0.497$) but it did help with their second piece of coursework ($U = 470, p = 0.039$). This is most likely

Question	Test Result	8-bit Emulator	Battleship Bots
Do you like Science Fiction	Mann-Whitney U	379.500	426.000
	Wilcoxon W	484.500	2317.000
	Z	-0.648	-0.014
	Asymp. Sig. (2-tailed)	0.517	0.989
Do you run Linux at home	Mann-Whitney U	506.500	564.500
	Wilcoxon W	1682.500	942.500
	Z	-1.566	-0.922
	Asymp. Sig. (2-tailed)	0.117	0.356
Do you own a Raspberry Pi or other development board	Mann-Whitney U	445.000	562.000
	Wilcoxon W	1930.000	2047.000
	Z	-1.712	-0.367
	Asymp. Sig. (2-tailed)	0.087	0.714
Have you built a web-site before	Mann-Whitney U	483.000	461.000
	Wilcoxon W	2136.000	2114.000
	Z	-0.704	-0.966
	Asymp. Sig. (2-tailed)	0.482	0.334
Have you installed an Operating System before	Mann-Whitney U	312.500	322.000
	Wilcoxon W	2523.500	2533.000
	Z	-0.270	-0.123
	Asymp. Sig. (2-tailed)	0.787	0.902
Have you built a PC from component parts before	Mann-Whitney U	539.500	605.000
	Wilcoxon W	890.500	956.000
	Z	-1.213	-0.493
	Asymp. Sig. (2-tailed)	0.225	0.622

Table 2. Correlation between student prior technology experience and their first year coursework performance

explained by the nature of the two pieces of coursework, the first requiring a lot of directed repetitive programming and the second undirected problem-solving. This agrees with the findings of Colley et al (1996) were "enjoying solving complex problems" was one of the attributes linked to programming ability [19].

Student Plans for Work Students planning their work career has little effect on their performance in their first-year coursework, see Table 4, only the wish to work in a technical role had any correlation. Wishing to work in a technical role did not help with the first piece of coursework ($U = 93.5, p = 0.094$) but did help with the second piece of coursework ($U = 80.5, p = 0.050$). This is a similar result to enjoying solving puzzles and being good at chess. It may be those students who wish to work in a technical role enjoy the challenges that come with solving technical problems, again as found by Colley et al (1996) [19].

Student Attitude Towards Programming It was believed if a student enjoyed programming then they would be more motivated when carrying out the programming components of their piece of coursework. The Pearson Correlation

Question	Test Result	8-bit Emulator	Battleship Bots
Do you like solving puzzles e.g. Sudoku	Mann-Whitney U	541.000	309.500
	Wilcoxon W	751.000	519.500
	Z	-0.108	-2.883
	Asymp. Sig. (2-tailed)	0.914	0.004
Are you good at, or have ever been good at, chess	Mann-Whitney U	595.000	470.500
	Wilcoxon W	1456.000	998.500
	Z	-0.680	-2.064
	Asymp. Sig. (2-tailed)	0.497	0.039

Table 3. Correlation between student prior problem solving experience and their first year coursework performance

test, shown in Table 5, shows a correlation between programming enjoyment and the marks received for the programming portion of each piece of coursework. For the first piece of coursework ($r = 0.447, p = 0.000$) the enjoyment of programming contributed 20% to the mark given and for the second piece of coursework ($r = 0.303, p = 0.008$) enjoyment contributed 9%. However we cannot say which is the dependent variable, it could be that when a student finds that they are good at programming the task becomes enjoyable. This result again agrees with the findings of Colley et al (1996) [19]. It is also reasonable to expect that the longer a student spent working on a piece of coursework the higher the mark that they would achieve. The Pearson Correlation test, see Table 5, showed that the time spent on the Emulator positively correlated ($r = 0.292, p = 0.010$) to the marks achieved, emulating an Intel 8080 microprocessor involved a lot of repetitive programming and the more time spent would expectantly lead to a higher mark. This link between student motivation and programming achievement was reported by Jiau et al (2009) and Serrano-Cmara et al (2014) [20] [21]. The contribution time made to the mark achieved, however, was not great ($< 1\%$). There, however, was no strong correlation between the time spent working on the Battleship Bots piece of coursework and the mark achieved. This is most likely explained by the fact that the second piece of coursework required a brain, not brawn.

Student Aptitude Test Results Aptitude tests are frequently used in industry as a means of selecting suitable applicants for jobs that entail computer programming. In the literature positive correlations have been found between the ability to score highly in numeric aptitude tests and the ability to programme computers. The Pearson Correlation test was used to determine the strength of the relationship between the students aptitude test results and their coursework performance; the results are shown in Table 6. No significant correlations were found between the numeric, verbal or non-verbal aptitudes of the students and their coursework performance. A possible correlation ($< 1\%$) between the non-verbal aptitude of the students and the second piece of coursework may indicate that a strength in non-verbal reasoning may help with problem solving.

Question	Test Result	8-bit Emulator	Battleship Bots
Do you intend to go on a placement year	Mann-Whitney U	163.500	227.000
	Wilcoxon W	191.500	255.000
	Z	-1.316	-0.139
	Asymp. Sig. (2-tailed)	0.188	0.890
Do you have a specific career path in mind	Mann-Whitney U	564.000	669.000
	Wilcoxon W	1384.000	1335.000
	Z	-1.628	-0.531
	Asymp. Sig. (2-tailed)	0.104	0.596
Do you have a part-time job	Mann-Whitney U	654.500	677.500
	Wilcoxon W	1395.500	1418.500
	Z	-0.516	-0.270
	Asymp. Sig. (2-tailed)	0.606	0.787
Do you want to work in a technical role when you leave university	Mann-Whitney U	93.500	80.500
	Wilcoxon W	108.500	95.500
	Z	-1.676	-1.956
	Asymp. Sig. (2-tailed)	0.094	0.050

Table 4. Correlation between student plans for work and their first year coursework performance

Question	Test Result	8-bit Emulator	Battleship Bots
Enjoy Computer Programming	Pearson Correlation	0.447	0.303
	Sig. (2-tailed)	0.000	0.008
Hours Spent on Coursework	Pearson Correlation	0.292	0.202
	Sig. (2-tailed)	0.010	0.080

Table 5. Correlation between student attitude towards programming and their first year coursework performance

4.2 Cybersecurity In-depth

The coursework marks achieved by the 2015-16 cohort of year 2 students was compared to that of the 2016-17 cohort of year 2 students. The 2015-16 cohort was taught prior to the introduction of the new teaching framework, the WOW factor and the 2016-17 cohort after. The coursework marks were compared using the Wilcoxon Signed Ranks Test, it showed with 99.9% confidence that the marks were significantly different. Table 7 shows that the average coursework mark is significantly better for the 2016-17 cohort.

This improvement was explained by students being more motivated and their improved attendance at both lectures and practical sessions where discussion of the latest cybersecurity news, and its implications, took place. This improvement agrees with results published in [20] [21]. A Pearson's Correlation Test between attendance (the independent variable) and the 2016-17 cohort coursework mark was carried out. It showed that with > 99.9% confidence that a positive correlation ($R = 0.750$) exists between student attendance and the coursework mark

Question	Test Result	8-bit Emulator	Battleship Bots
Numeric Aptitude	Pearson Correlation	0.221	0.172
	Sig. (2-tailed)	0.240	0.364
Verbal Aptitude	Pearson Correlation	0.090	0.212
	Sig. (2-tailed)	0.649	0.364
Non-verbal Aptitude	Pearson Correlation	-0.002	0.348
	Sig. (2-tailed)	0.993	0.088

Table 6. Correlation between student aptitude test results and their first year coursework performance

Year	Mean	Std. Dev.
2015-16	36.32	20.15
2016-17	58.85	32.25

Table 7. Average coursework marks for the year 2015-16 and 2016-17 cohorts

that they achieve. Further, it implies that 56.3% of the coursework mark can be attributed to students attending their lectures and practicals. Figure 1 shows a scatter plot of coursework marks by student attendance.

Feedback from students has been highly positive. This includes students always eager to contribute their understanding of materials and applying it to current news/trends. Wanting to contribute in the lectures on how the topics linked to what is happening with the week. Waiting for lectures to discuss current trends, emailing staff with current trends in cybersecurity and talking to staff in the corridors. Some of the feedbacks include

- “Lectures and Labs are interesting, I am always engaged, we cover interesting news which is useful.”
- “Good examples used/analogies and interactive in class”.
- “Excellent contextualization of materials with real-world scenarios”

5 Conclusion

Very little correlation was found between students academic performance and their prior use of technology or the results of their aptitude tests. Where correlation was found it tended to relate to student motivation; this agreed with previously published work that student motivation was a key factor in their academic success. Items of interest that linked current topics being taught to current real-world events were gleaned from the news and other sources. Incorporating this material into a new teaching framework for the second-year students has significantly improved their academic performance. This success is attributed to their increased engagement and interest in applying their skills to real-world scenarios.

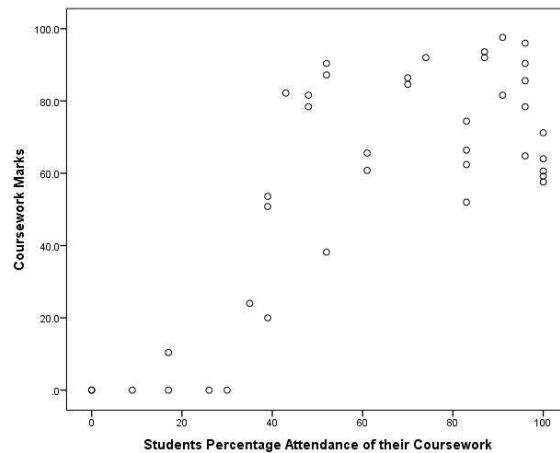


Fig. 1. Correlation between student attendance and coursework marks

References

1. Ramsden, Paul: *Learning to teach in higher education* (2003) Routledge
2. Biggs, John: What the student does: Teaching for enhanced learning. *Higher education research & development* 18, 1, 57-75 (1999)
3. Casey, Ashley: Models-based practice: Great white hope or white elephant?. *Physical Education and Sport Pedagogy* 19, 1, 18-34 (2014)
4. Boud, David: *Problem-based learning in education for the professions*. Higher Education Research and Development Society of Australasia (1985)
5. Dewey, John: The need for a philosophy of education (1934). *Schools* 7, 2, 244-245 (2010)
6. Fook, Jan, and Fiona Gardner: *Critical reflection in context: Applications in health and social care*. Routledge (2012)
7. Ghaye, Tony: *Teaching and learning through reflective practice: A practical guide for positive action*. Routledge (2010)
8. Kolb, David: *Experiential education: Experience as the source of learning and development*. Englewood Cliffs, NJ (1984)
9. Peterson, Kay, Lisa DeCato, and David A. Kolb: Moving and learning: Expanding style and increasing flexibility. *Journal of Experiential Education* 38, 3, 228-244 (2015)
10. Mezirow, Jack: Conclusion: Toward transformative learning and emancipatory education. *Fostering critical reflection in adulthood: A guide to transformative and emancipatory learning* 354-376 (1990)
11. Schn, Donald A: *Educating the reflective practitioner: Toward a new design for teaching and learning in the professions*. Jossey-Bass (1987)
12. Hume, Anne: Promoting higher levels of reflective writing in student journals. *Higher Education Research & Development* 28, 3, 247-260 (2009)
13. Boud, David, and David Walker: Promoting reflection in professional courses: The challenge of context. *Studies in higher education* 23, 2, 191-206 (1998)

14. Laurillard, Diana: Rethinking University Education: A conversational framework for the effective use of learning technologies. RoutledgeFalmer, London (2002)
15. Macfarlane, Bruce, and Lesley Gourlay: The reflection game: Enacting the penitent self. *Teaching in Higher Education* 14, 4, 455-459 (2009)
16. Comber, Chris, Ann Colley, David J. Hargreaves, and Lisa Dorn: The effects of age, gender and computer experience upon computer attitudes. *Educational Research* 39, 2, 123-133 (1997)
17. Wong, Harry K., Rosemary T. Wong, Stacey Allred, and Jenn David-Lang: The first days of school: implementation guide. HARRY K. WONG PUBLICATIONS, INC. (2016)
18. Turley, Richard T and Bieman, James M: Competencies of exceptional and non-exceptional software engineers. *Journal of Systems and Software* 28, 1, 19-38 (1995) Elsevier
19. Colley, Ann and Henry, Olivia and Holmes, Stephen and James, Louise: Perceptions of ability to program or to use a word processor. *Computers in human behavior* 12, 3, 329-337 (1996) Elsevier
20. Jiau, Hewijin Christine and Chen, Jinghong Cox and Ssu, Kuo-Feng: Enhancing self-motivation in learning programming using game-based simulation and metrics. *IEEE Transactions on Education* 52, 4, 555-562 (2009)
21. Serrano-Cámara, Luis Miguel and Paredes-Velasco, Maximiliano and Alcover, Carlos-María and Velazquez-Iturbide, J Ángel: An evaluation of students motivation in computer-supported collaborative learning of programming concepts. *Computers in Human Behavior* 31, 499-508 (2014) Elsevier
22. Hawi, Nazir: Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education* 54, 4, 1127-1136 (2010) Elsevier
23. Law, Kris MY and Lee, Victor CS and Yu, Yuen-Tak: Learning motivation in e-learning facilitated computer programming courses. *Computers & Education* 55, 1, 218-228 (2010) Elsevier
24. Feldt, Robert and Angelis, Lefteris and Torkar, Richard and Samuelsson, Maria: Links between the personalities, views and attitudes of software engineers. *Information and Software Technology* 52, 6, 611-624 (2010) Elsevier
25. Valdemar vbensk, Jan Vykopal, Milan Cermak, Martin Latovi?ka, Enhancing Cybersecurity Skills by Creating Serious Games, ACM ITiCSE 2018 conference 2018
26. Rob Byrd Cybersecurity: 1) what math is necessary and 2) developing ubiquitous cybersecurity in current computing programs *Journal of Computing Sciences in Colleges* Volume 33 Issue 4, April 20 Pages 53-59
27. Bishop-Clark, C.: Cognitive style, personality, and computer programming. *Computers in Human Behavior*, 11, 2, 241-260 (1995)
28. Ranjeeth, S., & Naidoo, R.: An investigation into the relationship between the level of cognitive maturity and the types of errors made by students in a computer programming course. *College Teaching Methods & Styles Journal (CTMS)*, 3, 2, 31-40 (2011)
29. Martin, A., Maria, V., Beena, A., & Lucia, T.: Prediction of Association among Numerical Aptitude, Programming Skills, Trait Emotional Intelligence on Students Performance. *International Journal on Computer Science & Engineering*, 4, 9 (2012)