Proceedings of the 11th International Conference on Applied Informatics
Eger, Hungary, January 29–31, 2020, published at http://ceur-ws.org

Are Computational Thinking Skills Measurable? An Analysis

Loice Victorine Atieno[a], Richard Kwabena Akrofi Baafi[b], Turcsanyi-Szabo Marta[c]

[a]Eötvös Loránd University, Budapest, Hungary
Atienomunira04@gmail.com

[b]Eötvös Loránd University, Budapest, Hungary
richbaafius@yahoo.com

[c]Eötvös Loránd University, Budapest, Hungary
tszmarta@inf.elte.hu

Abstract

Acquisition of digital literacy constitutes, not only knowledge on the use of software and or hardware, but also knowledge of sophisticated skills (cognitive, motoric, sociological and emotional) which are essential for functioning effectively [1]. Computational Thinking (CT) skills and their acquisition by learners and assessing their competency level is important. Even though many countries have integrated CT into their education curricula, comprehensive assessment tools for these skills are still lacking [2]. This paper presents the analysis of various assessment tools and demonstrated the use of customised version of Cognitive Skills Framework (CSF), also known as Skill-based Theoretical Framework [1], to assess CT skills. The study was conducted among first year computer science international students with basic or no programming skills. Scratch programming language was used to develop projects and content delivered using design-based approach. The study showed that, as the learners progressed in the course, their competency also increased. The research concluded that CSF has the capability of assessing CT skills comprehensively.

Keywords: Computational Thinking Skills, Cognitive Skills Framework, Competency Assessment, Assessment tools, Analysis

1. Introduction

Computational Thinking (CT) skills are essential skills for the 21st Century [3] hence are equally important as reading, writing and arithmetic [4]. This has led to several countries seeking for ways of integrating computing and CT into their curricula [5]. Despite the high integration of computing into curricula, no consensus has been reached regarding: what constitutes CT hence its definition [6, 7, 8]; integration of CT into the education setup [9] ; concepts of CT to be learnt at different levels; teaching learners how to use CT in different disciplines; and how to assess CT skills [7, 10].

This paper therefore analyses different CT assessment tools and frameworks, and explores the use of Cognitive Skills Framework (CSF) in assessing CT skills and their mastery level [1].

2. Assessing Computational Thinking Skills

Although the integration of CT in the school curricula is an ongoing process, its assessment still remains a big concern [11]. Assessing CT is important as it enables teachers measure learners' level of learning [12] and mastery of the content which are necessary for a successful integration of CT into education curricula [6].

2.1. Computational Thinking Assessment Tools

Considering K-12 education, there are diverse views and classification of CT assessment tools [2, 7].

CT Summative Tools: Include aptitude test (measuring basic programming ability) [13] and content-knowledge tools (assessing different levels of combined taxonomy based on the SOLO (Structure of Observed Learning Outcomes) and Bloom's taxonomy) [14].

CT Formative-iterative Tools: Provide automatic feedback helping learners improve their CT skills and are designed for specific programming environment. Example includes Dr. Scratch which gives scores to CT and detects programming errors [15].

CT Skills-Transfer Tools: Provide assessment of skills transfer to solving different kinds of problems. Example includes the Bebras tasks (assesses transfer to various real-life problems) [16].

CT Perceptions-Attitudes Scales: Measure creativity, algorithmic thinking, cooperation, critical thinking, and problem solving. Example includes the computational thinking scale (CTS) for assessing creativity, problem solving, critical thinking, collaboration and algorithmic thinking [17]

CT Vocabulary Assessment: measures elements and dimensions of CT expressed verbally by the learners [18].

According to Gonzalez, et al [6], the use of only one type of tools mentioned above can mislead the learners in understanding CT skills to be acquired. For example, Brennan and Resnick [7] observed that assessing learners' projects alone is insufficient for a comprehensive CT competency assessment. In order to complement each other and conduct a comprehensive assessment, a combination of the tools is proposed.

2.2. Computational Thinking Assessment Frameworks

Systematic combination of different assessment tools (known as systems of assessments [6]) is recommended in understanding learners' CT abilities better [10, 11]. Some of the CT assessment frameworks include:

The three dimensions of CT assessment framework: Consists of three approaches for assessing the acquisition of CT concepts, practices and perspectives [7]. The approaches include project analysis (analysis done using Scrape tool); artefact-based interviews (interviewing Scratchers); and design scenarios (developing design scenarios). Despite the strengths of the approaches none proved enough to stand on its own. The approaches have no proof of assessing CT perspectives; being combined to complement each other; they comfortably assessed CT concepts; and assessment of CT practices was time consuming thus not viable where many leaners are involved.

Framework for assessing development of CT: use of code inspection and artefact- based interviews to assess the development of CT concepts and practices [8]. Apart from the framework being time consuming, it assisted in discovering conceptual gaps. The major boosts of the framework were based on its capability of being able to be repeated at various points in time, examination of projects, motivating learners to conceptualize their processes, and ability of the learners to define and use a concept.

Complementary CT assessment tools: look at convergent validity between CT Test (CTt) and Bebras and between CTt and Dr. Scratch [6]. The tools cannot be interchanged with the other but can complement each other leading to development of system of assessment. The tools were mapped to Bloom's taxonomy with the CTt mapped to the lowest constructs of the taxonomy (Remember and Understand); Bebras tasks mapped to the middle constructs (Apply and Analyse); and Dr. Scratch mapped to the top most constructs (Evaluate and Create). The use of different set of Babras tasks or set of projects in case of Dr. Scratch could lead to the realisation of different correlation values hence the need for further research.

System of assessment framework: a combined method that includes coding, learning and life skills is the most viable solution to assessing CT [2]. They further noted that coding performance and emotions can change at different times within the same activity; several activities can cause learning to occur; and changes to cognitive, social and life skills can only be because of different experiences over a period coupled with previous factors. There is therefore need for a system of assessment that incorporates the three points. The framework assesses not only the technical but non-technical skills too. The framework is in two parts hence the need to blend the two parts of the framework. It also requires scaling down to minimise time consumption and resources used if it is to be adopted in the classroom.

3. Cognitive Skills Framework

Considering the strengths and limitations of the tools and frameworks discussed, it is evident that they concentrate on assessing practical, cognitive and emotional skills, ignoring social and relational skills [17]. There is no clear indication of what is expected of the students if learning is to be counted to have occurred. For the development of a successful assessment framework, several suggestions should be considered. They include support for further learning, incorporating artefacts, illuminating processes, checking in at multiple waypoints, valuing multiple ways of knowing, and including multiple viewpoints [7].

The CSF [1], which is a type of digital literacy framework, presents an easily formattable assessment tool that can be used in different learning environments. The tool enables learners to know what constitutes mastering the required skills, learning activities that lead to the acquisition of the skills and the process of acquiring the skills [19]. Three different types of digital literacy frameworks were identified due to their alignment with high education instructional settings and goals. This included CSF, Short guide to digital humanities [20] and Association of College and Research Libraries (ACRL) Standards [21].

The selection criteria of the frameworks included alignment of the framework with academic content; flexibility and capability to be customised for different learning content; and articulation of the competencies that when paired with learning activities can be measured [19] . This research adopted CSF for assessment. The customised version of the framework excellently represents types of digital skills and methods taught to learners. The framework can enhance collaboration between learners and teachers as it provides a diagnostic and assessment tool that facilitates creation of accurate, user-directed projects [1].

4. Teaching and Assessing Computational Thinking Skills

The design-based learning approach [22, 23] was used which enabled the learners to create projects at the end of each learning activity. The syllabus was adopted from Creative Computing Curriculum [10], that was scaled down to meet the requirements of the course. The area adopted was the themes while the projects stemmed from students' own ideas. The main evaluation tool was the CSF which was customised to capture the CT skills and the activities required to acquire the skills. The study took place at the Faculty of Informatics, Eötvös Loránd University. The survey involved a total of 9 year-one computer science international students which was the total population of the target group.

The study spanned over a period of 14 weeks (duration of the course) and Scratch programming used to implement the course. The choice of Scratch as was prompted by its enticing features. Even though students are familiar with informatics, teaching programming has proved challenging to most teachers [24]. Use of traditional ways to teach programming has yielded poor results raising questions on effective ways of executing the tasks [25]. Scratch has capability of changing the traditional way of teaching programming (viewed as boring) into a creative learning experience [7]; Is a media rich environment (use of text, audio, visual); Viewed as a way to other programming languages such a java, python etc; And provides a low threshold for programming enabling beginners understand programming structures and focus on logic problem before syntax [26].

The learning process was divided into 5 themes which included introduction to Scratch, exploring arts, digital storytelling, gaming and developing own choice projects (Table 1).

Themes	**Description**	**Competencies**	**Required Sessions**
Introduction and Integration	Introduction to creativity in computing and Scratch, using sample projects and hands-on experiences. Using basics in Scratch and self-guided tutorial to create an interactive postcard	Integration; communication and presentation skills; creativity and problem-solving skills; collaboration	6
Exploring Arts	Exploring arts by creating projects that include elements of music, design, drawing, and dance	Sequence; iterative and incremental	4
Digital Stories Telling	Exploring storytelling by creating projects that include characters, scenes, and narrative	Parallelism and events; reuse and remixing	4
Developing Games	Exploring games by creating projects that define goals and rules.	Conditionals; operators and data; testing and debugging	6
Final project – Own Choice Project	Developing independent projects by first identifying the suitable project, collaborating with others to improve the project, and presenting the project and its development process	Enhance the developed skills	8

Table 1: Course themes and competencies

Each activity begun with the explanation of each CT skills to be acquired and snippet of a programme created in class demonstrating its use. The learners were then required to create their own choice of project based on the theme using the CT skills learnt. They were also encouraged to collaborate with each other through sharing, questioning, presentation of their progress and peer review. The peer reviews and final assessment of each project was done using the assessment framework – customised based on Eshet Cognitive Skills Mastery Matrix (ECSMM)[1] – and it enabled the articulation of the skills to be developed at different mastery levels as shown in the excerpt in Table 2.

Themes	**Skills Categories**	**Skills**	**Level 1 Understanding /Remembering**	**Level 2 Apply/ Analyze**	**Level 3 Create/ Evaluate**
Introd. to Scratch	CT Concepts	Sequencing	Show understanding of sequencing,	Identify use of sequencing in sample projects	Create a simple scratch project using sequences to introduce themselves
	CT Practice	Incremental & Iterative	-	-	Describe process used to create their project
	CT	Connecting, Expressing	Be able to introduce themselves	Present on their introduction project giving the main points	Give an extensive presentation of their introduction project

Table 2: Skills Mapped to Mastery Levels

The framework provided an insight into the skills to be acquired and the learning activities associated with them. This was accomplished using the customized version of the framework's Google Sheets Mastery Matrix template[2]. The learning outcomes and activities were associated to expected skills for different mastery levels using the mastery metrices as shown in Table 3.

The learning activities begun by giving the learners an opportunity to envisage their desired projects and process needed to achieve their goal. The teacher played the role of facilitator and consultant. Apart from developing projects, various activities were carried out to enable comprehensive assessment. Every learning activity comprised of self-reporting that required learners to keep record of their own progress and sequence of steps leading to the completion of the project. The learners were therefore able to keep track of their progress, identify areas needing correction and refine their project. The learners were able to share ideas and challenges through collaboration, group projects participation, peer reviewing, project

[1] `https://docs.google.com/document/d/1OZiyKwaSXlM9cnOsz5dGDLjZm8tmF2wHNu8qPGqQa94/edit`

[2] `https://docs.google.com/spreadsheets/d/1rP5FXq33CdFZPbr123hhMMC1R46Qz-TqxEfIeoMRSTQ/edit?usp=sharing`

Source	Theme	Competencies	Mastery Level 1:Remember/ Understand	Mastery Level 2:Apply/ Analyse	Mastery Level 3:Evaluate/ Create
Eshet, 2012 (Customized)	**Introd. to Scratch**	**Sequencing; Incremental; Iterative; Connecting; Expressing**	Understand the use of the skills	Identify the skills used in the project	Use the skills in a project
		Skill:*Learners should be able to...*	Understand the use of the skills	Identify the skills used in the project	Use the skills in a project
		Activity:*Learners will develop this skill by...*	Going through the demo project: All about myself	Going through the demo project: All about myself	Create own project
		Time:	20 Min.	20 Min.	3 hours

Table 3: Mapping of the Learning Outcomes and Activities to the Expected Skills

presentation and reflection. These activities facilitated development and enhancement of CT practices (testing and debugging, incremental and iterative) and CT perspective (connecting and expressing) which could not be assessed by analysing the complete projects. The results of the self-reporting assessment concurred with the peer reviewing assessment and the teacher's final evaluation.

5. Findings of the Study

The findings showed that CT skills can be assessed using the customised framework. This was possible by mapping the learning outcomes to the framework. Assessment of the projects revealed use of various CT concepts and practices as shown in Figure 1.

Figure1 demonstrated the acquisition of CT concepts (sequence, loops, conditional among others) and practices such as abstraction and modularisation. Scratch provided an environment for developing and enhancing the skills and the mastery level of the skills acquired was determined using the ECSMM.

Likewise, the results of the survey carried out showed that there was improvement in the acquisition and enhancement of other skills such as testing and debugging, and collaboration. This is evident in the comparison of the survey carried out at the beginning and the end of the course as shown in Figures 2 and 3, and Figures 4 and 5 respectively. Based on the survey the learners were able to transfer the skills to other area. For example, 89 percent used collaboration in other subjects.

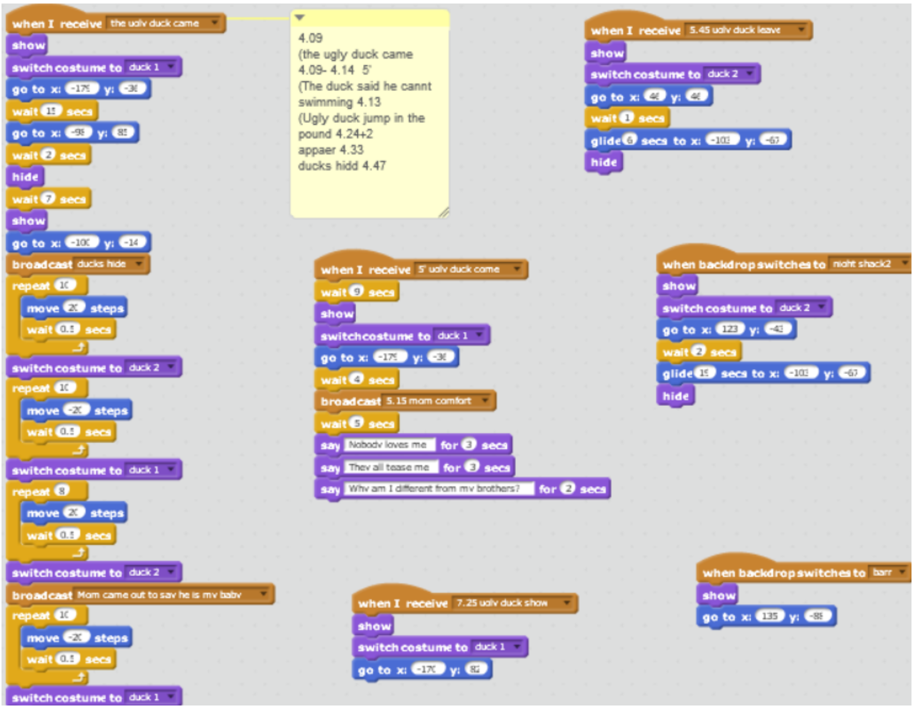

Figure 1: Blocks Showing use of Various CT Concepts and Practices

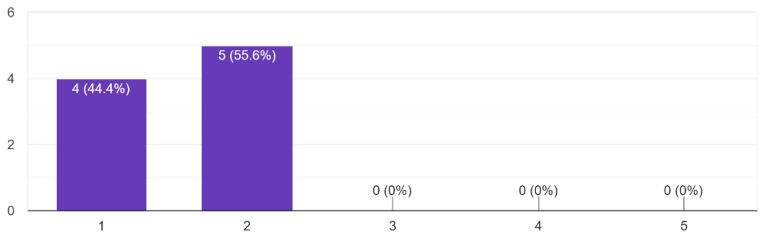

Figure 2: Competency in Testing and debugging skills at the beginning of the course

6. Discussion

The results of the study showed that CSF can comprehensively assess CT skills. Based on its flexibility (that enables customisation) and structure, the activities carried out could easily be assessed.

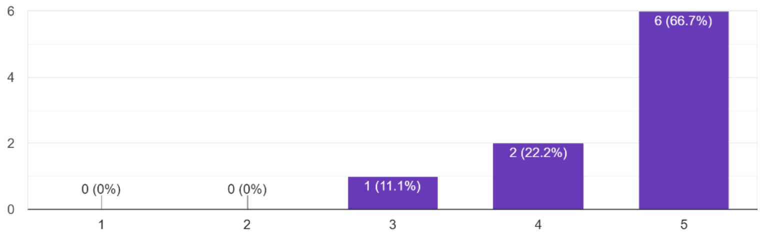

Figure 3: Competency in Testing and Debugging based at the end of the Course

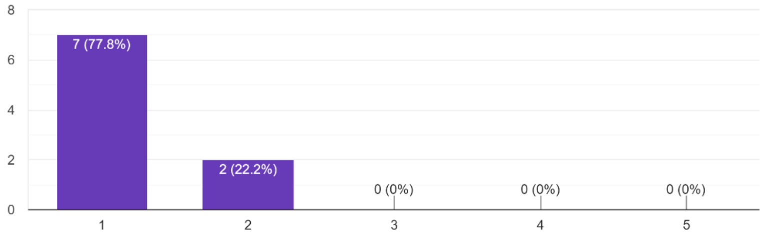

Figure 4: Competency in Collaboration based on self-assessment at the beginning of the Course

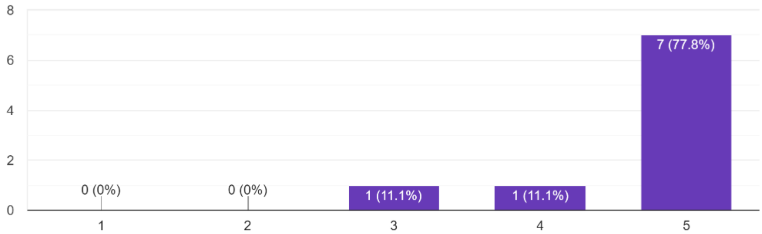

Figure 5: Competency in Collaboration based on self-assessment at the end of the Course

a) Computational thinking concepts

The CT concepts well displayed in the project which made it easy to map the outcome to the framework. Various ways have been used to assess the skills in projects. For example analysis of projects portfolio using Scrape tool [7]. The learners' projects were subjected to three levels of evaluation. These were peer reviewing (gave the learners opportunity to evaluate themselves); internal evaluation (carried out by the course facilitators using the framework); and external evaluation (done by lecturers in the department).

b) Computational Thinking Practice

The learners were encouraged to learn and use ideas from other projects either among themselves or from online Scratch community. This is a common practice in design and problem solving [7]. The projects likewise displayed abstraction and modularisation. Skills such as testing and debugging were used frequently as they worked on the projects. Testing and debugging strategies used in Scratch included: trial and error, learning from existing projects and getting support from those with knowledge on the task [7]. These strategies were utilised by the learners.

c) Computational Thinking Perspective

One of the non-technical skills enhanced by CT is connecting and expressing. Developing projects using Scratch is further facilitated by sharing [7]. The learners developed collaboration skills by participating in group projects; presenting progress of on going project in order to solicite more and unique ideas from their peers; and peer-reviewing which helped in making a critical analysis of their projects. This gave the learners confidence to express themselves fully.

The use of customised CSF has shown that, apart from cognitive skills, social skills can be assessed using this framework. In this research, the assessment was made possible by engaging in several activities leading to one common goal and that is development of complete projects. A comprehensive assessment system should be able to support further learning, incorporate artefacts, highlight processes, creating multiple intervention points during the learning period, appreciating multiple ways of knowing and considering multiple viewpoints [7]. The CSF clearly supports all these suggestions in its assessment roadmap.

Studies have called for the adoption of different types of complementary assessments tools if comprehensive understanding of learners' CT skills is to be reached [11, 24]. This has been seen to be with the various CT assessment frameworks. With the customised CSF it is possible to map the activities, processes and deliverables to the learning objectives set at the beginning of the course. This made it possible to assess CT skills comprehensively using a single tool.

7. Conclusion

The aim of this research was to identify a comprehensive CT assessment tool and demonstrate how it can be used in the assessment of CT skills. Using a customised CSF, the assessment of different CT skills was made flexible and learner-centred. The CT skills were developed using Scratch based on its features that makes it appealing even to non programmers. The framework stipulated the requirements for achieving competencies at various levels of learning leading to enhancement of the learners' self-organised learning. The detailed nature of the framework enabled the learners to envisage the expected learning outcomes paving way to self-assessment.

This is important especially in the current era where constructionism and learner-centred approaches to learning are highly encouraged. Apart from the learners, the flexibility of the framework may be useful to teachers in carrying out comprehensive assessment of CT skills. The study has proven that CT skills both technical and non-technical skills are measurable. Likewise, comprehensive assessment of the skills can be carried out using a single assessment tool. The study was carried on a small population. Further research is therefore recommended to validate the framework on a larger population and on different learning contexts.

Acknowledgements. My sincere gratitude goes to Eötvös Loránd University, Faculty of Informatics for presenting with the opportunity to carry out this research. The faculty provided an innovation lab that gave the students a user-friendly environment to work in. My gratitude also goes to my colleagues in the department who created time to evaluate the students' final projects. Lastly to my students who made this research a success.

References

[1] Eshet, Y. Y., Thinking in the digital era: A revised model for digital literacy, *Issues Informing Sci. Inf. Technol.,* Vol. 9 (2012), 267–276.

[2] Basso, D., Fronza, I., Colombi, A., Pahl, C., Improving Assessment of Computational Thinking Through a Comprehensive Framework, *The 18th Koli Calling International Conference* (2018), 1–5.

[3] Wing, J. M., Computational thinking - What and why?,*Link Mag.,*(2010).

[4] Wing, J. M., Computational Thinking - The begening, *Commun. ACM,* Vol. 24 (2006), 33.

[5] V. Dagiene, V., Sentance, S., Stupuriene, G., Computational Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics, *Informatica,* Vol. 28 (2017), 23–44

[6] Roman-Gonzalez, M., Moreno-Leon, J., Robles, G., Complementary Tools for Computational Thinking Assessment, *International Conference on Computational Thinking Education,*(2017).

[7] Brennan, K., Resnick, M., New frameworks for studying and assessing the development of computational thinking, *Paper presented at the meeting of AERA 2012, Vancouver, CB, Canada,*(2012).

[8] Fronza, I., Ioini, N.E., Corral, L., Teaching Computational Thinking Using Agile Software Engineering Methods, *ACM Trans. Comput. Educ.,* Vol. 17 No. 4 (2017), 1–6.

[9] Lye, S.T., Koh, J.H.L., Review on teaching and learning of computational thinking through programming: What is next for K-12?, *Comput. Human Behav.,* Vol. 41 (2014), 51–61.

[10] K. Brennan, K., Chung, M., Hawson, J.,Creative Computing: A Design-based Introduction to computational Thinking, (2011.

[11] Grover, S., Cooper, S., Pea, R., Assessing computational learning in K-12, *Proceedings of the 2014 conference on Innovation and technology in computer science education,*(2014), 57–62.

[12] Grover, S., Pea, R., Computational Thinking in K–12, *Educ. Res.,* Vol. 42 no.1 (2013), 38–43.

[13] Muhling, A., Ruf, A., Hubwieser, P., Design and First Results of a Psychometric Test for Measuring Basic Programming Abilities, *Proceedings of the Workshop in Primary and Secondary Computing Education,*(2015), 2–10.

[14] Meerbaum, O., Armoni, M., Ben-Ari, M., Learning computer science with scratch, *Comput. Sci. Educ.,* Vol. 23 no.3 (2013), 239–264.

[15] Moreno-Leon, J., Robles, G., Roman-Gonzalez, M., Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking, *RED-Revista de Educación a Distancia.N,* Vol. 46 (2015).

[16] Dagiene, V., Stupuriene, G., Informatics Education based on Solving Attractive Tasks through a Contest, *KEYCIT 2014 – Key Competencies in Informatics and ICT proceedings,*(2015), 51–62.

[17] Korkmaz, O., Cakir, R., Ozden, M.Y., A validity and reliability study of the computational thinking scales (CTS), *Comput. Human Behav.,* Vol. 72 (2017), 558–569.

[18] Grover, S., Robotics and Engineering for Middle and High School Students to Develop Computational Thinking Robotics and Engineering for Middle and High School Students to Develop Computational Thinking, *Annual Meeting of the American Educational Research Association New Orleans,*(2011), 1–15.

[19] Appert L., Digital Literacy Competency Calculator-Rationale and Process, *DLCC White paper*, (2017).

[20] Burdick, A., Drucker, J., Lunenfeld, P., Presner, T., Schnapp, J., A Guide to the Digital Humanities, *MIT Press,*(2012), 121–136.

[21] ACRL Board, Framework for Information Literacy for Higher Education, (2015).

[22] Gómez Puente, S. M., Design-based learning: exploring an educational approach for engineering education, (2014).

[23] Gomez Puente, s. M., Van Eijck, M., Jochems, W., A sampled literature review of design-based learning approaches: A search for key characteristics, *Int. J. Technol. Des. Educ.,* vol. 23 no. 3, (2013), 717–732.

[24] Nikiforos, S., Kontomaris, C., Chorianopoulos, K.,MIT Scratch: A Powerful Tool for Improving Teaching of Programming, *in Conference on Informatics in Education,* (2013).

[25] Tew, A.E., Guzdial, M.,Developing a Validated Assessment of Fundamental CS1 Concepts, *in SIGCSE '10 Proceedings of the 41st ACM technical symposium on Computer science education, ACM New York,* (2010), 97–101.

[26] Malan, D.J., Leitner, H.H.,Scratch for Budding Computer Scientists Terms of Use Scratch for Budding Computer Scientists, *ACM SIGCSE Bull,* Vol. 39 No. 1 (2007), 223–227.

[27] Grover, S., Pea, R.D., " Systems of Assessments" for Deeper Learning of Computational Thinking in K-12, *Annual Meeting of the American Educational Research Association, Chicago,*(2015).