

Computational Thinking through Creative Programming in a Computer Science Course

Foteini Paraskeva, Aikaterini Alexiou, Stavros Emmanouil Panoutsos*,
Department of Digital Systems, University of Piraeus, 80 Karaoli and
Demetriou, 18534 Piraeus, Greece, *Corresponding Author

Creative and computational competencies are part of the so-called 21st century skills. The creative exploitation of digital systems is related to a predominant Computational Thinking for problem solving. This paper explores the enhancement of creative and Computational Thinking in the context of programming skills for pre-service computer science teachers in Higher Education. A blended learning environment was implemented, using web 2.0 technologies, for a course studying ‘how people learn with technology’. The learning environment incorporated the ‘Six Thinking Hats’ method along with a creative thinking model. An experimental procedure was conducted with pre-test and post-test measuring creative thinking factors, such as fluency, elaboration, flexibility and originality. The participants were 33 undergraduate students of a computer science department. Results showed that creative thinking increased during the experimental procedure. This study suggests that the proposed conceptual framework would support learners to engage in and enhance their creative thinking when teaching programming.

Key words: *Computational Thinking, Creative Programming, Computer Science, Higher Education.*

1 INTRODUCTION

It is widely considered that technology education in today’s society is of great importance, since, in almost every field, technological literacy is a valuable asset. In this context, important questions arise referring to the educational methods that are suitable in order to teach technology-related subjects, especially to computer science students.



Programming skills are at the core of technology education for they enable individuals to develop their very own solutions to a vast number of challenging situations. Furthermore, the knowledge of a programming language is not limited to mere software design, but can potentially provide far more benefits. Through programming, students collaborate with each other and start to think as a computer programmer, developing a new perspective, an alternative way of thinking. This new way of thinking related to computer and programming skills is also referred to as Computational Thinking (CT). In this context, the use of creative thinking may prove beneficial to students, for it can potentially enhance all features necessary for effective programming (Greenberg, 2007).

The purpose of this study was to design a conceptual framework that supported pre-service computer science teachers to enhance creative thinking when teaching programming. To this end, a blended learning environment was implemented, using web 2.0 technologies, in the context of a “Learning with Technology” laboratory course of a university department. Learning environment was orchestrated along the lines of a conceptual framework, which incorporated the ‘Six Thinking Hats’ method along with a creativity model (Applied Psychology Tools to Creativity And Computational Thinking – APT2C&CT). The use of such a framework makes it possible for students to create significant collaboration and design their own mental path in computer programming experiences thus motivating them to activate their mental processes underlying the creativity theoretical model.

There are a significant number of studies that underpin the blending of creative thinking and CT in learning environments. Recent findings suggest that creative thinking can be further enhanced in a learning context that includes CT factors (Kim, Song et al., 2014). Not only is creative thinking involved in scientific achievements (Barrett, Vessey et al., 2014), furthermore, the addition of exercises related to creative thinking in computer science courses can improve CT and also computer knowledge and skills (Shell, Hazley et al., 2014). However, the design of such a framework needs to be addressed with caution.

With all this taken into consideration, this study will focus on the following research question: Is it possible through the implementation of the conceptual framework (APT2C&CT) to support pre-service computer science teachers to bolster creative thinking when teaching programming?

2 THEORETICAL BACKGROUND

As it was already mentioned, the relation between creative thinking and CT in the context of a learning environment can prove beneficial to computer science students. Seminal studies suggest that a conceptual framework may strengthen creative thinking. The question however still remains; what is the optimal design for such a framework in order to be effective and reliable? The main factors need to be further analysed.



2.1 Creative Thinking

The term “Creative Thinking” is involved in many fields of human activity and is thus open to many different interpretations according to the very context in which it occurs. Although the exact definition of creativity is still a matter of debate, researchers agree that there are three different theoretical backgrounds depending on how creativity is defined.

Initially there was the notion that creativity is a personality trait of some individuals (personality based). The main advocate was Maslow (1968), who stated that creativity is a universal characteristic of a person during self-development. According to the second background supported by behavioural psychologists (behaviourist view) and mainly represented by Skinner, the right environment is what will eventually affect the appearance and further development of a person’s creativity. Finally, the most recent theoretical background mainly supported by Wallas, Osborn, Guilford, Gordon and Koestler, is one that defines creativity as a cognitive process. In other words, creativity is regarded as a mental process that can be learned like any other learning process, by everyone. In the ensuing section are listed some key definitions of creativity.

According to Torrance (1966), creativity is the skill that a person has when dealing with various problems, showing sensitivity and originality but also remaining in order and calm. Guilford (1975), proposed that creativity covers the most typical skills of creative people, which determine the probability of a person to express a creative behaviour, expressed with ingenuity and cohesion. Finally, according to Paraskevopoulos (2004), creative thinking is the ability of the human mind to seek and find multiple original-innovative alternatives for solving various problems, as well as to investigate long-term impacts resulting from a product (Whiting, 2017).

As it was mentioned before, creative thinking is related to computer science and programming skills. Collaboration among individuals and the search for innovative ways and ideas to address problems is quite often the norm in projects that involve programming. It is therefore important to address the main factors of CT.

2.2 Computational Thinking

The idea of CT is not new; it was initially proposed by Seymour Papert in 1996, in order to analyse effective educational methods for teaching mathematics to students. It was later popularised by Jeannette Wing (2006) who defined CT as “a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use”. Thus, CT makes use of cognitive mechanisms that are not limited to sole programming skills; they can be applied in almost every domain. Although a universally acceptable definition of CT is still a subject of debate, researchers agree that CT is consisted of the following elements (Grover & Pea, 2013):

- Abstractions and pattern generalisations
- Systematic processing of information
- Symbol systems and representations
- Algorithmic notions of flow of control
- Structured problem decomposition (modularising)
- Iterative, recursive and parallel thinking
- Conditional logic
- Efficiency and performance constraints
- Debugging and systematic error detection

The use of CT in education leads to the next challenge: what are the exact properties that programming environments should have in order to foster CT in computer science students? Since the development of the LOGO programming language, researchers agree that programming environments must have a “low floor-high ceiling” design, meaning that these tools should be suitable to both types of users, beginners and experienced, respectively. Moreover, they must have scaffold, enable transfer, support equity, and be systemic and sustainable (Repenning & Webb, 2010).

Although there is a significant number of programming environments that do fit these criteria, little has been written about the way they can be embedded in educational curricula, in order to teach computer science students and bolster CT. Important questions referring to the skills of computer science teachers, the introductory methods preceding these subjects, as well as ways to exert student’s motivation, need to be addressed.

First of all, the difficulties that students face when learning CT, have to be considered. One important issue is the lack of operational definitions of CT and the misconception that CT is related solely to programming languages, a fact that leads many students away from it (Lu & Fletcher, 2009). Furthermore, they don’t feel confident in dealing with complex situations and difficult, open-ended problems. It is also important to mention that the discipline of Computer Science (CS) demands a high level of collaboration among individuals, since large projects typically require teams of developers who have to work together, sometimes with discipline-specific experts. In this context, many students find it very challenging to set aside differences in order to work together, especially in inter-disciplinary environments (Barr & Stephenson, 2011).

According to the Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) (2016), there are certain ways to introduce CT in educational curricula. These methods include the extensive use of computational vocabulary and notation in order to describe problems and understand abstractions. Moreover, it is better for students to work in groups with explicit use of decomposition (breaking down problems



into small, easy-to-solve parts), abstraction (simplifying concepts), negotiation (working together within a team) and consensus building (group solidarity). Referring to the teachers, there is a need for resources in order to clarify the core concepts of CT and provide specific examples of how they can be applied into a range of curricular areas. To this end, teacher's professional development could be enhanced through demonstrating the role of CT in other non-CS disciplines, participation in learning communities, use of model simulations and activities for students and encouraging educational associations and policy makers to recognise the importance of CT.

In addition, CT related subjects already implemented and evaluated within educational contexts may provide valuable information. Seminal studies suggest that a subject encompassing problem-driven approaches focused on scientific discovery and computational principles may, in fact, enhance student's interest in computing (Hambruch et al., 2009). Results from introductory CT courses indicate that methods like the use of practical examples related to the student's experience, teaching in a problem-driven way and visualising concepts can prove beneficial. Furthermore, for developing programming skills there is a need to use a programming language that supports interactive environments, enables students to be engaged and quickly develop sophisticated programs, provides feedback, visualisation and it is widely recognised by the scientific community.

Similar cases from informal education also provide useful insights. For instance, projects involving simulation, models, robotics and computer game design helped students to better understand abstraction, automation and analysis (Leonard et al., 2017). Key concepts of CT like iterative design, refinement and reflection underlie all these activities, concepts which, according to Resnick (2007), are central not only to computational but to creative thinking as well.

One effective approach to this issue is the use of a creativity strategy in the teaching process, in order to underpin collaboration among students and encourage their motivation. To this end, the "Six Thinking Hats" model by Edward de Bono was incorporated in the educational method. Through this method students will plan thinking processes together, in a detailed and cohesive way, adopting different perspectives and developing their collaborative skills (De Bono, 1985).

2.3 The Six Thinking Hats Method

According to the Six Thinking Hats method, six metaphorical hats, each with a different colour, are being used by an individual in order to indicate the type of thinking that is being used. The white hat focuses on information and facts, the red hat covers feeling, intuition and emotion, while the black hat is for judgement and caution. The yellow hat represents positive thinking, the green is for creativity, alternatives or proposals, and finally, the blue one covers process overview and meta-cognitive thinking. Depending on the topic to be addressed, it is



possible for formal hat sequences to be used. Alternatively, the order of the hats may vary according to the specific situation. One advantage of the six thinking hats method is that it allows its users to switch hats (and thus thinking modes) easily and flexibly, a fact that may allow many different aspects of a discussed topic to emerge, hence boosting creativity (De Bono, 1995).

The Six Thinking Hats method was applied in APT2C&CT since students were expected to create certain deliverables in each phase of the Lab. The Six Thinking Hats Method enables students to develop many aspects about a topic thus enhancing creative thinking especially when learning involves web technologies (Gregory & Masters, 2012).

3 METHOD

3.1 Research Method

The research process included an experimental procedure with only one team of participants (identical-subjects design). The purpose of this procedure was to observe the impact that the experimental process had over the students, in relation to the assigned variables. In this case, researchers conducted repeated measurements to the subjects that belonged to only one group. This group was used as an experimental and as a controlled group as well.

3.2 Research Question

This study will focus on the following research question: Is it possible through the implementation of the conceptual framework (APT2C&CT) to support pre-service computer science teachers to bolster Creative Thinking when teaching programming?

3.3 Participants

The sample of the research study composed of 33 undergraduate university computer science students. The experimental process was conducted during winter semester and was part of the laboratory of the elective course “Learning with Technology”. The enrolment of the students was optional. From the 33 participants, 26 were male (78.8%) and 7 were female (21.2%). All participants have signed a consent form prior to the course and completed the activities successfully.

3.4 Towards a course of “Learning with Technology” with Creativity and Computational Thinking

This research focuses on designing and developing a conceptual framework based on a creative thinking model aligned with the Six Thinking Hats method, in the context of a Web



2.0 learning environment, in order to prepare pre-service computer science teachers to be able to teach programming following and applying creative thinking principles.

To accommodate the needs of this experiment, a conceptual framework was designed based on the creative thinking model developed by Clinton & Hokanson (2012). This model is consisted of the following phases: Problem Identification, Preparation, Incubation, Illumination and Elaboration.

This model was combined with the Six Thinking Hats creativity method developed by Edward de Bono (1985). In this way, each phase of the model was aligned to the corresponding “Thinking Hat” of de Bono’s method.

In the first phase of “Problem Identification” students were asked to use the white hat in order to think with objectivity, without their own personal opinions, focus on the events and ask questions. During the second phase of “Preparation” students were asked to use the green hat in order to think more creatively, finding new and alternative ideas. At phase three, “Incubation”, students had to combine different hats, starting from the green one and then proceed to the yellow hat, in order to think about the advantages of an idea. Finally, they were asked to use the black hat, in order to address the idea with disbelief and think about the disadvantages. In the fourth phase of “Illumination”, students had to use a combination of hats, starting from the green one and then proceed to the red hat, in order to think using their very own emotions and insights. No reasoned thinking and no justification was used at this point. Finally, during the “Elaboration” phase, students were told to use the blue hat, for they had to consolidate the upcoming parts of their project and focus on the objectives.

3.5 The Web 2.0 Environment: APT2C&CT Environment

An appropriately configured Web 2.0 learning environment provided by a wiki-based platform was designed and developed. ‘APT2C&CT environment’ was designed in order to provide a robust platform for supporting the development of the conceptual framework. Towards this, the ‘APT2C&CT Environment’ orchestrates the learning process along the lines of the creative thinking model (Clinton & Hokanson, 2012). The environment is structured based on the phases: Problem Identification, Preparation, Incubation, Illumination and Elaboration. Each phase consists of several collaborative activities described in further detail in the ensuing section.

3.6 Experimental Procedure

There are certain limitations that have to be considered. 33 participants enrolled for the course “Learning with Technology”. The enrolment for the course was optional, a fact that limited random sampling.



During the first session (Phase 1), the first meeting of the students was made. Students were introduced to the objectives of the course and to the concepts of creativity and computational thinking. The tutor then demonstrated the use of 'APT2C&CT Environment' and all participants were asked to form teams of 2 to 3 persons and create personal accounts. Team formation was crucial since, in this context, students may discuss a wider range of topics thus enhancing creative thinking (McMahon, Ruggeri et al., 2016). The maximum number of team members was limited to 3 persons based on the difficulty of the required deliverables and also on other factors concerning creative thinking. Relatively small teams are more aware of team goals, more committed and eager to communicate, discuss and debate when addressing an issue (Bradner & Mark, 2005). This increased flow of ideas that is emerging within small teams is crucial for creative thinking.

Students had then access to the learning resources and were asked to create a team page in the 'APT2C&CT Environment', where they had to present their team in a creative way. Only team members had access to the team page and had to upload all deliverables there. The first deliverable that students had in session 1 was to design a poster under the topic: "The advantages of programming".

The second session (Phase 2) was performed online and students had to deliver a conceptual map under the topic: "What are the objectives of programming and how they can be aligned to educational programming environments?" Students successfully uploaded the deliverables for both sessions 1 and 2. For the development of these deliverables, certain online tools were proposed by the tutor such as Glogster (Glogster S.A.) and Cmaps (IHMC).

In the third session (Phase 3) the second meeting of the students was made. The tutor presented various educational programming environments, giving examples so that the students could further try for themselves. Examples of these environments included Scratch (Scratch Foundation), Lego Mindstorms (LEGO) and Code & Conquer. In addition, the concepts of the educational activity and creativity strategy were presented, followed by a list of programming objectives for the fifth and sixth grade of elementary school. The deliverable of session 3 was a document under the topic: "Advantages and Disadvantages of Educational Programming Environments". Finally, a template was given to the students in order to structure their next deliverable (Deliverable 4), a document titled: "Development of an educational activity using a creativity strategy and an educational programming environment for the course of programming". Students uploaded all deliverables successfully.

During the fourth session (Phase 4) students studied learning materials referring to good practices in order to make a presentation. The deliverable of this session was a "Presentation of the educational activity of Deliverable 4". Another meeting with the students was made and each team presented the educational activity and received feedback.

In the fifth and last session (Phase 5) students were asked to upload updated versions of their deliverables based on the given feedback. For this reason, they had to submit final versions for both Deliverables 5 and 6. All participants completed the activities successfully. The alignment of sessions, phases, thinking hats, interaction modes and deliverables is presented in Figure 1.

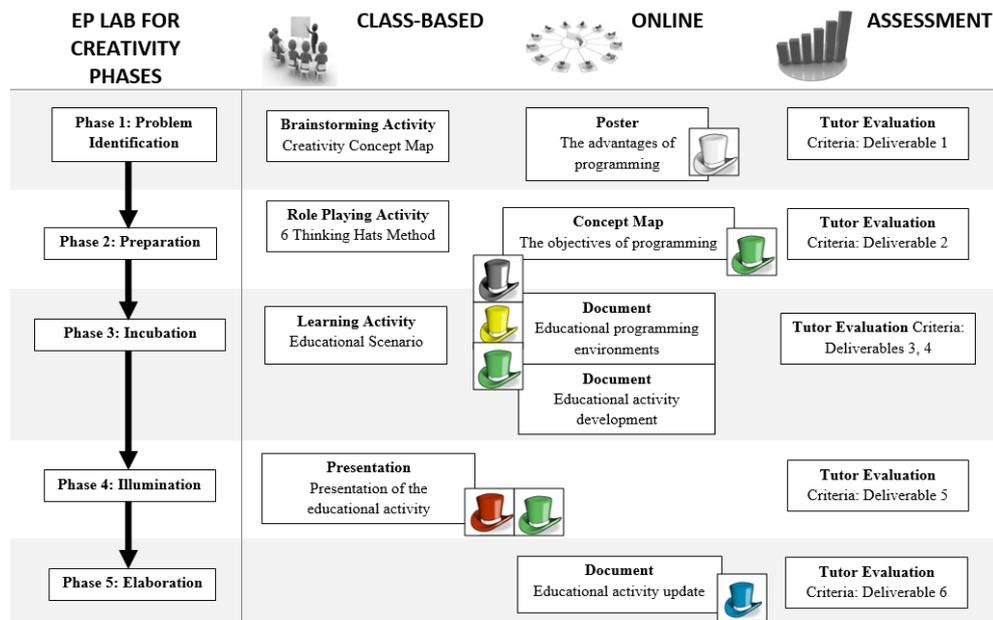


Figure 1: Sessions, Phases, Thinking Hats, Interaction Modes and Deliverables Alignment

3.6 Instruments

According to the research objective, this study focused on only one variable: Creativity. In order to measure Creativity, both quantitative and qualitative measurements were taken. During these measurements the following factors were used.

For the quantitative measurement a rubric with a 1-5 scale was used. The design of this rubric and its components was based on the work of Miller (2009) and included six factors, namely Incubation, Perspective, Metaphorical Thinking, Brainstorming, Imagery and Flow. This rubric was delivered to the students twice, before and after the completion of the activities. For the qualitative measurement, a rubric including four factors, namely Fluency, Elaboration, Flexibility and Originality was used. The design of this rubric and its components was based on the work of Torrance & Ball (1984) and Guilford (1997) and provides significant information about individual's creativity (Metwaly, Castilla et al., 2018). This rubric was used by the tutor only once, after the completion of the activities.

3.7 Results

As it was mentioned before, this study will focus on the following research question: Is it possible through the implementation of the conceptual framework (APT2C&CT) to support pre-service computer science teachers to bolster creative thinking?

Analysis was made using SPSS software, version 22.0 (IBM Corporation). During the quantitative measurements, a paired samples t-test was used in order to examine the mean difference before and after the completion of the activities. The null hypothesis stated that there was no mean difference, while the experimental hypothesis stated that there was a significant increase between pre-test and post-test means. Results are shown in Table 1.

Table 1: Paired Samples T-Test Differences for Creativity

	<i>M</i>	<i>SD</i>	Standard Error	95% Confidence Interval Lower Upper	<i>t</i>	<i>df</i>	Significance (one-tailed)
Pre-Test Mean	-0.029	0.178	0.031	-0.092 0.033	-0.950	32	0.174
Post-Test Mean							

The one-tailed paired samples t-test using an α level of 0.05 failed to reveal a statistically significant difference between the mean number of pre-test and post-test results of the students, $t(32) = -0.950$, $p = 0.174$.

In addition, an analysis of the correlation among creativity factors was performed using Spearman's r . The analysis yielded strong correlation among most of the factors, a fact suggesting that if one individual performs well with respect to one factor (e.g. Flow), it is expected to perform also well in other factors (e.g. Incubation). Results are shown in the following Table 2.

Table 2: Correlation between Creativity Factors Post-Test (Spearman's r)

Factor	Incubation	Perspective	Met. Thinking	Brainstorming	Imagery	Flow	Mean
Incubation	1	0.444*	0.496*	0.090	0.638*	0.415*	0.669*
Perspective	0.444*	1	0.657*	0.366*	0.572*	0.613*	0.793*
Met. Thinking	0.496*	0.657*	1	0.309*	0.600*	0.547*	0.779*
Brainstorming	0.090	0.366*	0.309*	1	0.274	0.393*	0.541*
Imagery	0.638*	0.572*	0.600*	0.274	1	0.665*	0.854*
Flow	0.415*	0.613*	0.547*	0.393*	0.665*	1	0.842*

Note. *Correlation is significant at the 0.05 level (one-tailed).

During the qualitative measurements, a chi square test for goodness of fit was used in order to examine the frequency of the students' responses. The null hypothesis stated that there was

no difference, while the experimental hypothesis stated that there was a significant difference between the observed frequency of the responses and a predetermined level.

The chi square test using an α level of 0.05 revealed a statistically significant difference between the frequencies of the responses that the students have, $\chi^2(1) = 9.308$, $p = 0.002$. Therefore, the null hypothesis can be rejected since significantly less student teams have been classified to the first ($N = 0$) or second ($N = 1$) category, compared to the third ($N = 12$). Results are shown in the following Table 3, where T1-T13 refers to each student team.

Table 3: Frequency Table for Creativity Factors

Factor	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
Fluency	30	38	23	24	26	29	32	31	24	21	30	33	23
Elaboration	14	12	14	8	24	13	17	16	10	14	16	23	26
Flexibility	16	19	13	12	12	15	13	14	17	14	17	17	14
Originality	5	7	4	2	6	6	4	4	2	5	2	1	3
Total	65	76	54	46	68	63	66	65	53	54	65	74	66
Category	3	3	3	2	3	3	3	3	3	3	3	3	3
Frequency				1*								12**	
Percent				7.7*								92.3**	

Note. *Category 2 **Category3

In addition, a frequency histogram for Creativity Factors clearly reveals the classification of student teams, showing that 12 teams were classified to the third category, while only one student team was classified to the first category.

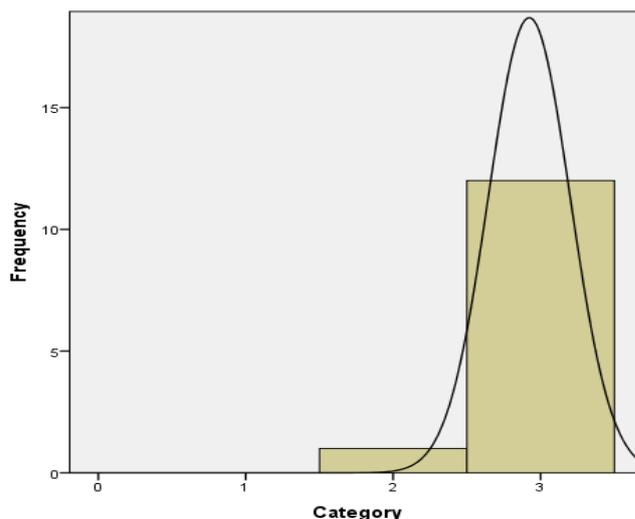


Figure 2: Frequency Histogram for Creativity Factors

4 CONCLUSION

In conclusion, the purpose of this study was to design a conceptual framework that supported pre-service computer science teachers to enhance creative thinking when teaching programming. A blended learning environment was implemented in the context of a conceptual framework including the Six Thinking Hats method along with a creative thinking strategy. The experimental procedure included pre-test and post-test measurements regarding creative thinking factors.

Findings suggest that a blended learning course incorporating the Six Thinking Hats method is indeed enhancing creative thinking among computer science students. All measurements yielded statistically significant results, except for the quantitative measurement for creative thinking factors.

These outcomes are consistent with the scientific literature, which suggests that a creativity strategy is, in fact, increasing creativity and collaboration among students, as it is more likely for novel ideas to emerge in a collaborative environment. The implementation of the Six Thinking Hats method was indeed found to be beneficial for creativity and information exchange in computer mediated classrooms of tertiary education (Belfer, 2001). In addition, results from previous studies have shown that the Six Thinking Hats method offered important improvements in corporate environments as well, boosting the generation of novel ideas and increasing productivity (De Bono, 1995).

However, there are a small number of studies yielding non-significant statistical results when implementing the Six Thinking Hats method in educational curricula. For instance, Carl (1996) found no significant improvements in ideas exchange and debate among small groups. These facts underpin the need for careful and detailed design when implementing the Six Thinking Hats method, in order to overcome potential limitations and exploit the real advantages of the method.

Among the limitations of the current study is the relatively small and not randomly selected sample size. Furthermore, time constraints may have caused trouble for the use of Thinking Hats during class-based sessions.

Regarding to future work, the same creativity model could be applied to a bigger sample of students as, in this way, more reliable results referring to creativity could be obtained. Furthermore, a framework incorporating a different creativity strategy, other than the Six Thinking Hats method, could be used in order to examine the behaviour of students under different thinking methods.



REFERENCES

- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.
- Barrett, J. D., Vessey, W. B., Griffith, J. A., Mracek, D., & Mumford, M. D. (2014). Predicting Scientific Creativity: The Role of Adversity, Collaborations, and Work Strategies. *Creativity Research Journal*. 26(1), 39–52.
- Belfer, K. (2001). De Bono's Six Thinking Hats Technique: A Metaphorical Model of Communication in Computer Mediated Classrooms. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2001*, 113-116.
- Bradner E., Mark G., & Hertel T. D. (2005). Team size and technology fit: Participation awareness and rapport in distributed teams. *IEEE Transactions on Professional Communication*, 48(1), 68-77.
- Carl III, W. J. (1996). Six Thinking Hats: Argumentativeness and Response to Thinking Model. *Annual Meeting of the Southern States Communication Association*.
- Clinton, G., & Hokanson, B. (2012). Creativity in the training and practice of instructional designers: the Design/Creativity Loops model. *Educational Technology Research and Development*, 60, 111–130.
- Creativity Research Journal*, 28(3), 247-257.
- De Bono, E. (1985). *Six Thinking Hats: An Essential Approach to Business Management*, Little, Brown, & Company.
- De Bono, E. (1995). Serious creativity. *Journal for Quality and Participation*, 12.
- Greenberg, S. (2007). Toolkits and interface creativity. *Multimedia Tools and Applications*, 32(2), 139-159.
- Gregory, S., & Masters, Y. (2012). Six hats in Second Life: Enhancing preservice teacher learning in a virtual world. *Proceedings of iCTLT (International Conference on Teaching and Learning with Technology)*.
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43.
- Guilford, J. P. (1975). Creativity: A quarter century of progress. *Perspectives in creativity*, 37-59.
- Guilford, J. P. (1997). *Way beyond the IQ*, Creative Education Foundation: Buffalo, NY.
- Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *SIGCSE '09*. 41(1), 183-187.
- Kim, S. H., Song, K. S., & Park, S. Y. (2014). Exploring the Technological Factors Affecting Creativity in Computational Thinking-Centered Learning Context. *WSEAS Transactions on Advances in Engineering Education*, 14, 70-79.
- Leonard, J., Barnes-Johnson, J., Mitchell, M., Unertl, A., Stubbe, C. R., & Ingraham, L. (2017). Developing teachers' computational thinking beliefs and engineering practices



- through game design and robotics. *North American Chapter of the International Group for the Psychology of Mathematics Education*.
- Lu, J. J., & Fletcher, G. H. L. (2009). Thinking about Computational Thinking. *SIGCSE' 09* 41(1), 260-264.
- Maslow, A. H. (1968). *Toward a Psychology of Being*, D. Van Nostrand Company: New York.
- McMahon, K., Ruggeri, A., Kämmer, J. E., & Katsikopoulos, K. V. (2016). Beyond Idea Generation: The Power of Groups in Developing Ideas.
- Metwaly, S. S., Castilla, B. F., Kyndt, E., & Van den Noortgate, W. (2018). The Factor Structure of the Figural Torrance Tests of Creative Thinking: A Meta-Confirmatory Factor Analysis. *Creativity Research Journal*, 30(4), 352-360.
- Miller, A. (2009). *Cognitive process associated with creativity: Scale development and validation*, PhD Thesis, Ball State University.
- National Educational Technology Standards for Students (NETS) (2nd edition), International Society for Technology in Education (ISTE) (2016). Retrieved from: <http://www.iste.org/standards/nets-forstudents.aspx>
- Paraskevopoulos, I. N. (2004). *Creative Thinking in School and Family*, Athens.
- Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE '10)*, 265–269.
- Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *Proceedings of the ACM 2007 Creativity and Cognition Conference*. Washington, DC.
- Shell, D. F., Hazley, M. P., Soh, L. K., Miller, L. D., Chiriacescu, V., & Ingraham, E. (2014). Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers. *Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE)*, 1-7.
- Torrance, E. P. (1966). *Thinking creatively with pictures: Figural booklet*, Scholastic Testing Service.
- Torrance, E. P., & Ball, O. E. (1984). *Torrance tests of creative thinking: Streamlined (revised) manual*, Scholastic Testing Services: Bensenville IL.
- Whiting, P. (2017). Design thinking: The search for innovation, creativity & change. *International Journal Of Innovation, Creativity And Change*, 3(1), 55-64.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.