

Five C Framework: A student-centered approach for teaching programming courses to students with diverse disciplinary background

Mary Tom

School of Engineering and Technology
Central Queensland University, Australia
m.tom@cqu.edu.au

Abstract

The existing complexities of teaching and learning computer programming are increased where students are diverse in their disciplinary backgrounds, language skills, and cultures. Identifying opportunities for improvement and applying theoretical and empirical evidence found in literature, this study presents the Five C Framework – Consistency, Collaboration, Cognition, Conception, and Creativity – that integrates constructivist and collaborative learning theories in a student-centered teaching pedagogy. This framework was found to be effective in postgraduate courses in introductory programming over three consecutive terms. Analysis conducted using survey and interviews indicates that use of the Five C Framework reduced negative emotional issues, motivated students to become active learners, and improved the overall performance. The Five C Framework may thus be seen to provide a model for student-centered teaching pedagogy which helps to minimise complexities for diverse student cohorts.

Keywords

Student-centered teaching, collaborative learning, diversity, constructivist learning, peer instruction

Introduction

Australian universities have become learning organisations with very diverse student populations. This has made classroom environments interesting and challenging in terms of learning and teaching activities. Unfortunately, however, some students with diverse disciplinary backgrounds, language skills, cultures, perceptions, aptitudes and motivations experience emotional issues of anxiety, fear or boredom. The study described in this paper focusses on how a highly diverse cohort of students enrolled in the Master of Information Technology (MIT) course at Central Queensland University (CQU) has been supported through the introduction of the Five C Framework.

The MIT program at CQU was designed to provide graduates from disciplines other than IT or mature-age students with field experience an opportunity to obtain formal qualification in the field of Information and Communication

Technology (ICT). Facilitating migration from other disciplines benefits the ICT industry as it produces graduates with essential domain expertise in different areas. In this context, the varying levels of aptitude, interest, preparedness, and expectations of students add further complexity to student diversity and make the job of teaching programming more challenging. On the other hand, in order to design, develop, maintain and troubleshoot systems which are continually growing in complexity and size, highly-skilled programmers are needed. Considering the complexity and difficulties, teaching programming to students from different disciplinary backgrounds requires a careful approach and customised pedagogy. Simultaneously, these students need to go through an intensive learning process to achieve the expected postgraduate skill level in two years of study.

Learning programming requires the development of a multi-level skill that could be applied in different contexts using different programming languages to a multitude of application domains. Students need to develop knowledge in control constructs, handling input/output and error messages, data structures such as arrays, lists, and maps, many other concepts including recursion, inheritance, polymorphism, encapsulation, and machine related concepts of memory management, references/pointers, memory leaks, and garbage collection. In addition, learning programming requires skills in using correct programming language specific syntax and debugging the source code. Developing a clear understanding of new computer programming concepts is difficult and challenging due to the hidden nature of program execution and preponderance of jargon or discipline specific terms. Newly introduced programming concepts may remain vague until they are well practised.

Students from non-IT backgrounds may not be able to reflect and build the necessary cognitive structures without a mastery of early concepts (Olds, McKenna, & Pazos, 2007). While they may gain a basic understanding of concepts, they may be unable to apply their low-level understanding to real world programming problems (Tan, Ting, & Ling, 2009). Some of the characteristics negatively impacting found on students' achievements include:

- i. their lack of time management, motivation beyond completion of assessment requirements, systematic disciplinary approach in learning, and promptness in seeking help (Pullan, Drew, & Tucker, 2013);
- ii. a gap between the student's and the teacher's perspective of learning outcomes and task completion (Thompson, Hunt, & Kinshuk, 2006);
- iii. the known fact of "programming stress"(Glass, 1997);
- iv. the initial perception of students that learning programming is hard (Tan, Ting, & Ling, 2009).

Further, in a classroom of students with disciplinary diversity, repetitive explanations focused on students from other disciplines can create boredom (Kosheleva & Kreinovich, 2010) for students with an ICT background.

There can be a question of how designed and desired learning outcomes can be achieved by cohorts of students with varying learner characteristics and/or diverse disciplinary backgrounds. This article reports the author's experience of developing and implementing a student-centered learning and teaching framework, the Five C Framework, that caters to the needs of student cohorts with diverse disciplinary background and learner characteristics. This study and the development of the framework focused particularly on the variable elements of student motivation, perceptions, and effort, overcoming the impact of the inherent and rather static elements of learners' previous experiences. The Five C Framework was developed with an understanding of the students' perspective of learning or more specifically the *act of learning* or *how do they learn* (Mota, Vaz de Carvalho, & Reis, 2011). Engaging students from the three perspectives of affective, cognitive, (Wright, 2011) and behavioural levels to build interest and deeper understanding (Mayer & Chandler, 2001) is achieved through the Five Cs, that is, *Consistency, Collaboration, Cognition, Conception, and Creativity*.

Students should be empowered with reflective lifelong learning skills to be successful in the highly dynamic field of ICT where programming languages and tools constantly evolve. Also, students need to develop soft skills such as teamwork, communication skills, critical thinking, and creativity with the latter being one of the skills most sought after by industry (Welkener, 2013). Traditional teacher-centric pedagogy is focused on the course content and transferring knowledge to the students whereas a learner-centric view is focused on assisting students to develop or build knowledge (Lister, et al., 2007; Wright, 2011). Computer science pedagogy varies from developing skills in design and implementation focusing on the understanding of the machine aspects or real world application aspects such as object-oriented design and development (Berglunda, et al., 2009; Lister, Box, Morrison, Tenenberg, & Westbrook, 2004). Therefore, a move from a teacher-centric to a learner-centered design is recommended (Smart, Witt, & Scott, 2012; Wright, 2011).

The Five C Framework is founded on constructivist (Derry, 1996; Dijkstra, 1997) and collaborative learning theories (Gholson & Craig, 2006; Harney, Hogan, & Broome, 2012; Mingfei & Jie, 2010; Xu, Han, & Zhang, 2008) where knowledge is constructed:

- *physically* by active learning;
- *symbolically* by the creation of mental representations;
- *socially* by sharing understanding; and,
- *theoretically* by explaining things having incomplete understanding.

Peer Instruction (PI) is well established by many studies especially in science, technology, engineering, and mathematics (STEM) subjects. It is based on small group discussions which positively contribute to student understanding (Crouch & Mazur, 2001; Porter, Bailey Lee, & Simon, 2013; Simon, Kohanfars, Lee, Tamayo, & Cutts, 2010; Simon, Parris, & Spacco, 2013). The PI framework was designed to assist students to progress through the seven levels of cognitive process and develop factual, conceptual, procedural, and metacognitive

knowledge as per the revised Bloom's Taxonomy (Bloom, 1956; Krathwohl, 2002). Developing ideas or solutions require cognitive restructuring and assimilation (Quevedo-Torrero, 2009). Collaborative learning facilitates:

- *cognitive constructivism* where peer discussion leads to improved conceptual understanding (Crouch & Mazur, 2001);
- *social construction of new knowledge* by reflecting on the new material, cognitive restructuring, cognitive rehearsal, assimilation, correction of misconceptions, and searching for different perspectives of the newly introduced concepts through interaction and activity with others by asking questions, listening to explanations, and answering conflicting questions; and,
- *teamwork* and *deeper understanding* promoted by interaction among team members with varying levels of prior knowledge (Sangin, Molinari, Nüssli, & Dillenbourg, 2011).

Collaboration also promotes cultivation of creativity and communication skills. A student who engages in a creative activity within the constraints of time and environmental elements and working beyond the constraints of current norms/boundaries goes through a process transforming his/her knowledge into a product that is fulfilling for the student (Kleiman, 2008). Public oral presentations constitute an excellent opportunity for students to develop their communication skills, and stimulate reflection, critical thinking, and analysis (Requena-Carrion, Alonso-Atienza, Guerrero-Curieses, & Rodriguez-Gonzalez, 2010).

Background

As part of the MIT course at CQU, students undertake two core courses in programming: (a) COIT29222 *Programming Principles*, where they learn programming fundamentals; and (b) COIT23001 *Object-Oriented Principles*, for learning advanced level programming concepts. The testing and evaluation of the Five C Framework was carried out by participating students enrolled in the COIT29222 course. The *Programming Principles* course includes topics on variables, primitive and complex data types, control constructs, algorithms, files and streams, principles of object-oriented (O-O) programming, and fundamentals of user interface design. The intended learning outcomes include an understanding of O-O principles, development of quality software, algorithm design, coding with control constructs, variables, methods, and use of language specific library. The course objectives include developing transferable skills such as critical thinking and analysis, oral communication, and teamwork.

The delivery of this course, over a semester, was through two hours of lecture, and two hours of tutorial where students used an advanced level IDE (integrated/interactive development environment) to code and debug applications. Two written assignments and an end-of-semester examination formed the course assessment. Each of the assignments required the students to design a programming solution to a given problem and to submit the algorithm and a working program following effective coding standards. The examination weighed

50% of the total score. The students could get a failing grade if the overall score was below 50%. Students scoring 85% or higher were awarded with a High Distinction (HD) and students scoring between 75% and 85% were awarded a grade of Distinction (D).

The Five C Framework

The 5Cs of the Five C Framework are:

1. *Consistency* – in teaching and learning practices;
2. *Collaboration* - in problem solving, social construction of knowledge;
3. *Cognition* – developing higher-order thinking
4. *Conception* – understand concepts through elaboration, assimilation and examples; and,
5. *Creativity* – create solutions by applying concepts learnt.

Figure 1 graphically shows how the Cs fit together and how they support the learners who engage in lectures, group discussion, collaborative problem solving, task completion and undertake regular home study.

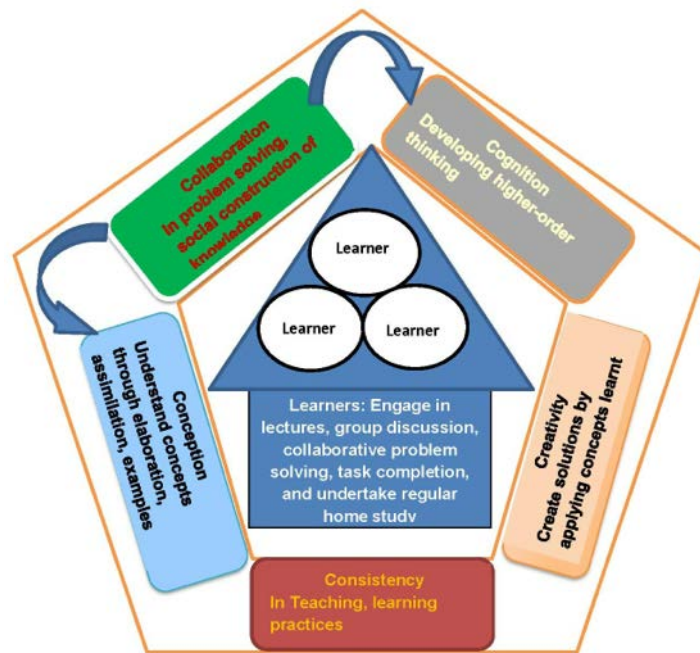


Figure 1. The 5C Framework

The implementation of the Five C Framework was founded on accepted practices such as explaining concepts using multiple methods. Students set definite goals in terms of final results and a workable time schedule for learning in the beginning of the term with guidance from the lecturer. Examples and problems were designed to give a more authentic and contextualised experience to the students. The practical implementation of each of the Five C constructs is discussed in the following sections.

The Five C Framework transforms the traditional 2-hour lecture – an audiovisual presentation with little or no interaction – into multiple “packs” of interactive sessions. The number of packs is dependent on the number of major concepts to be delivered in a lecture. As illustrated in Figure 2, each pack consists of four sessions of:

- Explanation or Elaboration
- Conceive and Communicate
- Interaction
- Collaborative Problem Solving.

Traditional Lecture	Lecture following the Five C Framework
Lecture explaining concepts and examples with very little interaction.	Concept 1: Explanation / Elaboration
	• Conceive and Communicate Session
	• Interaction Session: Demonstration of Concept Application with example
	• Collaborative problem solving / Task completion
	Concept 2: Explanation / Elaboration
	• Conceive and Communicate Session
	• Interaction Session: Demonstration of Concept Application with example
	• Collaborative problem solving / Task completion
	Concept 3: Explanation / Elaboration
	• Conceive and Communicate Session
	• Interaction Session: Demonstration of Concept Application with example
	• Collaborative problem solving / Task completion

Figure 2: Structure: Traditional vs lecture following Five Cs

The sessions within each pack are designed based on one or more of the Five C constructs of *cognition, conception, collaboration, consistency, and creativity*. These packs of four different sessions systematically guide the student from their origin, with little or no knowledge, to their destination where the students feel very comfortable with the new idea and its application. In addition to the multiple packs of four sessions during lecture, “show and tell” sessions are included as part of the tutorial sessions.

Explanation/Elaboration session: The purpose of this session was to provide students with an explanation of the major concept, new terms, and fundamentals related to the new concept. During this session the lecturer explains and, where necessary, elaborates on the concept to be taught using simulations and real world examples, and referring back to previously-taught concepts in use.

Conceive and Communicate session: Collaborative learning is facilitated through this session which includes 3-5 minutes of small group discussions. To assist students with their group formation, students are given the opportunity during the first lecture of the semester for self-introduction, thus sharing information on their disciplinary background. This provides an opportunity for students with non-computing backgrounds to identify peers with computing backgrounds that could be useful in team formation. Students form into groups of three and engage in a brief discussion, promoting intra-group communication, analysis, and critical thinking, thus enhancing and further refining the conception of ideas. The group discussions are guided by questions related to the main concept and are intended to enable the social construction of new knowledge. These communication sessions reduce the difficulties experienced by students in framing questions and expressing their doubts in the early stages of the course.

Interaction session: During the interaction session, students raise questions or share their insights gained during their group discussions. Based on this feedback, the new concept is further explained and linked to existing schemas in the knowledge construction process. The questions and discussions raised during this session also help to clarify misconceptions exposed during the Conceive and Communicate sessions. The interaction session also enables students to share their group learning experiences with the whole class.

Collaborative Problem-solving session: Students work together in small teams in solving a given problem or undertaking an activity where they could apply the newly-learnt concepts. This facilitates the application of concepts to specific problems supported by exploration, critical thinking and analysis. This also stimulates evaluation as group members examine different approaches within the team.

Show and Tell session: The “show and tell” session provides an opportunity for students to present their own developed programming application to the class and discuss the source code. Students generally apply two or three of the major concepts they have learnt in the applications they develop. Interestingly, this promoted creativity as evidenced by many interesting and original applications exhibited by students. This session also promoted continuous learning as students took more interest in study outside class. Students were encouraged to make one presentation per term at a scheduled time.

Method

Participants

The MIT students who participated in this study belonged to three categories, those having a Bachelor's degree in either: Computing, Engineering/Science, or Arts/Commerce. A total of 105 students participated in the testing and evaluation of this framework. These students were enrolled in the *Programming Principles* course in one of the three terms commencing from Term 1, 2012. Sixty per cent of the students ($n=63$) enrolled in COIT29222 in Term 1 2013, belonged to Categories 1 and 2 while 40 per cent ($n=42$) of students belonged to Category 3.

Data Collection Techniques

The outcomes and impact of the Five C Framework have been investigated using a multipartite strategy. In 2012, the teaching of *Programming Principles* was conducted following the Five C Framework and informal student feedback and end of term teaching evaluation were used to verify its effectiveness. In Term 1 2013, a more formal method of data collection using a survey and selective interviews was adopted to evaluate the effectiveness of the framework. Initial learning difficulties experienced by the students on entry into the course were investigated using a survey and the impact of using the Five C Framework was investigated using interviews and the analysis of assessment results. Informal feedback received through individual student emails also provided valuable information.

Survey

Custom-designed survey questions were completed by the cohort of students enrolled in *Programming Principles* in Term 1 2013 ($n=105$). The survey was completed in class towards the end of the teaching period. The questions required answers with nominal or ordinal values as well as open-ended questions. The questions were designed to confirm the emotional issues confronted by the students, to understand their motivation to master the course content and the benefits of the Five C Framework.

Interviews

A number of students ($n=10$) enrolled in *Programming Principles* during Term 1, 2013 were interviewed. Each interview lasted between 20 to 30 minutes of which the last 3-5 minutes were used to collect students' own views or experiences that could not be expressed while answering the specific questions. Three representative interviews were analysed further. Each interview consisted of six questions framed to investigate student learning experiences in each component of the Five C Framework and collect any constructive feedback or comments to refine the practices.

Cohort marks and grades

The grades of students enrolled in COIT29222 *Programming Principles* Term 1 2013 were compared to the outcomes from two previous semesters. The three cohort results were comparable as they consisted of similar mix of students from diverse discipline areas.

Data analysis

The analysis results from the three methods of investigation are described in the following subsections.

Analysis of Survey Responses

The first question of the survey was used to verify the emotional issues experienced by the students while learning programming. As depicted in Table 1, the majority of students experienced emotional issues of stress, anxiety, or fear which negatively impacted on their study. Roughly 20% of the students ($n=21$) had not experienced such feelings.

The second question was used to ascertain the aptitude and interest of the students to follow a career in ICT. As the cohort included 40% ($n=42$) of students migrating from other disciplines, their commitment is critical for success. This is an indication of the seriousness and dedication the students can devote to their studies. Such attributes are essential to appreciate and effectively use the extra activities and support available for the students.

The third question was to collect feedback on the effectiveness, and student experiences of the collaborative learning facilitated through *conceive and communicate* sessions. The answers from the students highlights the benefits of the peer supported collaborative learning to enrich the learning experience. Students' responses to the survey were analysed and the summary of analysis is presented with the themes exposed in Table 1.

Table 1
Analysis of survey responses

#		Response Themes	% Responses (N=105)
1	Have you experienced any emotional issues while learning programming in this term?	Experienced both anxiety and stress	10
		Experienced stress while learning and practicing	50
		Affected by fear while practicing coding	20
		Absence of emotional issues	20
2	What are your career plans on completion of Master of Information Technology?	Keen to follow a career path in IT within the areas of Software development or networking	100
3	How did the <i>Conceive and Communication</i> sessions help?	Provided opportunity to clarify doubts and improve clear understanding of concepts	20
		Assisted in deeper understanding of concepts, reduced stress and fear.	80

Analysis of interviews

The analysis of the collected interview responses ($n=3$) in Figure 2 strongly demonstrate that all the Five C Framework components have positively affected the learning process and enriched the student learning experience.

Interview Questions	Interview #1	Interview #2	Interview #3
How did the “ <i>conceive and communicate</i> ” sessions help you in your learning? What are the specific skills or benefits you achieved?	<ul style="list-style-type: none"> • Provided deeper understanding, Sharing of ideas, exposure to new ideas, enriching experience, clarification of doubts, • improved confidence, helped to structure knowledge, made learning more interesting, and fun, improved intercultural communication; • transformed learning into an enjoyable experience 	Enabled knowledge sharing, exposed different perspectives, helped partners who lagged in his/her studies, assisted gaps in learning due to absence, improved intercultural communication, Helped deeper understanding, and analysis	Provided deeper understanding, assisted in clarifying misconceptions, reduced fear, improved intercultural communication, enabled sharing of ideas, made learning an enjoyable experience, promoted reflection and further study
Have you experienced any problems during the <i>conceive and communicate</i> sessions	No problems	Rarely experienced problems to explain concepts within short time	No problems
What are the consistent learning practices that helped?	Read the chapter before lecture, clarify doubts using email queries, answer review questions, make note of references to previous topics	Prepared before the lecture, regular study at home after the lecture	Engaged in regular home study before and after the lecture
Talk about the consistent teaching practices that assisted?	Review questions before lecture, repetitive reinforcement during lecture help to link previously learnt content to currently taught content.	Review of previous topics before lecture helped a lot, reference and questions on previous topics helped to remind the basics	Regular revision in class assisted to re-fresh the topics and reinforced the basics. Use of real world and simulation examples
Describe the benefit of short activities during lecture.	Improve collaborative problem solving skills	Exposed to different problem solving approaches	Improved team work, and ability to critically examine peer’s work
How did the “show and tell” assist?	Really motivated to develop [an] interesting program and be creative	Provided the awesome experience of developing a new program	Helped creativity by generation of new ideas, became more confident to apply concepts

Figure 2: Analysis results of individual interview responses

Analysis of cohort marks and grades

Student performance (grades) in Programming Principles is shown in Figure 3 and can be compared to the student performance before following Five C Framework in Year 2011, Terms 1 and 2.

On comparison of the student performance between the terms with and without following Five Cs framework it is evident that there is substantial improvement and the failure rate has dropped below 15%. The influence of the Five Cs framework is more visible on comparison of the percentage of students achieving higher grades of HD and D. In 2011, around 11 to 13% of students achieved higher grades. In 2012 and 2013, between 20% and 28% of students scored higher grades.

Term Year of study	Course name	Fail Rate %	HD and D %
Term 1 2011	Programming Principles (without Five Cs)	38	13
Term 2 2011	Programming Principles (without Five Cs)	21	11
Term 1 2012	Programming Principles (following Five Cs)	14	19
Term 2 2012	Programming Principles (following Five Cs)	9	21
Term 1 2013	Programming Principles (following Five Cs)	0	28

Figure 3. Comparative data showing academic performance (grades) of students in Programming Principles before and after following Five C Framework

Validity Issues

This is an in-situ study of independently run university classes and therefore, no experimental controls have been imposed. The validity of this work is reasonably established with the following facts in implementing the Five C Framework, and data collection methods used.

Central Queensland University has several geographically distributed campuses and this study is conducted at Brisbane campus. The assessment items and core course materials are prepared by the course coordinator and a formal moderation process is followed to ensure application of same standards across campuses. Therefore, it can be ensured that the assessment items are not modified or compromised as part of this study.

The lecturer who has been involved in teaching of the introductory programming course used in this study has several years of experience in teaching programming. The final results compared are from cohorts of students taught by the same lecturer. These facts avoid possible persuasive effects, or influences that could have impacted the validity of this study.

Results and Discussion

The qualitative summary of the responses indicates that aspects of the Five C Framework had a positive impact in student learning. The consensus found in the interviewees' responses is a valid and reliable indication of similar experiences as intended in the design of the framework. Use of this framework motivates and engages students with an enriching and enjoyable learning experience, assists in developing a deeper understanding with improved cognitive skills, empowers students with ability to work in team and collaboratively solve problems, and transforms the whole learning process into an interactive, knowledge sharing experience. The various benefits of the Five C Framework as evidenced from testing evaluation of the framework are discussed below.

- **Removal of negative emotional issues and stress impacting study**

The analysis conducted in previous sections shows that this framework has achieved the desired effect of alleviating or reducing study related stress, anxiety or fear experienced initially. This has in turn promoted more enthusiasm, and learning has become a more enjoyable experience.

- **Promotion of active learning and increased engagement with persistent student effort**

As students were interested, they also made persistent efforts to master the course content and became more responsible and active learners. Students become active learners by engaging in group discussion, analysis, and application of concepts to solve problems. The opportunity to design and build their own application and present it to the class also motivates students to make extra effort and develop a keen interest in learning. This can also improve research skills and development of lifelong learning skills.

- **Deeper understanding of concepts through small group discussion**

The small group discussions and analysis of concepts, and shared misconceptions and conflicts assisted in deeper and clearer understanding of concepts. Informal student feedback reveals that this goal is achieved. One example of a student response from the Term 2, 2012 cohort was: *“The methodology followed by you helps every student to gain thorough knowledge of the subject especially the basic concepts.”*

This has also led to more engaged discussions on variations and application of concepts. The involvement of knowledgeable students in peer support alleviates the boredom that could be experienced otherwise. This also promoted sharing of appropriate examples of application of concepts leading to

improved assimilation. Students developed higher order cognitive skills of analysis and evaluation by collaborative problem solving.

- **Cultivation of creativity and soft skills of communication, and team work**

Students are motivated to work beyond the constraints and produce small software applications that could be exhibited to the class thus attaining fulfilment. The engagement and enthusiasm motivated students to be persistent in their efforts and push their cognitive skills to the higher level of creativity and experience fulfillment.

- **Overall improved student performance**

Collaborative learning was promoted through small group discussion, and interaction as a whole class promoted social construction of new knowledge. As this model has alleviated many of the negatively impacting emotional issues and motivated students to be persistent in their efforts, improved student performance could be achieved.

Having found the Five Cs framework effective, from Term 1, 2013 this was also introduced in lectures of COIT23001, Object-Oriented Principles. This practice has been continued since Term1, 2013 in both the COIT29222 and COIT23001 courses. The partnership formed by students in COIT29222 course is carried over to the project team formation for the final term Software Engineering course. Students approach this course with greater confidence, and readiness to take the challenge. During the project implementation, students exhibit creativity, and independence in coding and debugging software application with multi-layered architecture and many components.

Conclusion

As information systems are growing in complexity and size, excellent software developers are essential. Teaching programming is complex and challenging due to the hidden nature of program execution and large amount of jargon. This is further exacerbated when teaching students from diverse disciplinary background at the postgraduate level. The Five Cs framework provides a cohesive model, well founded on constructivist and collaborative learning theories, which is an applicable teaching pedagogy, especially suitable for diverse student cohorts. The building constructs of this framework are *consistency, collaboration, cognition, conception, and creativity*. Following a student centered approach this framework transforms the authoritative role of the lecturer to a guide and empowers students with active learning and leadership skills. This framework is implemented based on the perspective that academic learning is a socially embedded cognitive process and knowledge is socially constructed through interaction and activity with others. Use of this framework transforms the learning to an enriching and enjoyable experience, developing a deeper understanding with improved cognitive skills, and development of soft skills such as team work, communication, and oral presentation. This framework has also motivated students to follow consistent

learning and extend their skills to a creative level, alleviating many negatively impacting emotional issues.

The Five C Framework is a successful model for student-centered teaching pedagogy addressing issues around learning and teaching of programming. This framework is constructed to promote higher order cognitive skills, which is one of the fundamental learning outcomes of higher education. There is scope for extending the use of Five C Framework in other science and technology courses or where the student cohorts are diverse in their background, ethnicity, or aptitude. The Five C Framework and the experiential knowledge shared here may reduce the helplessness that academics feel when assisting students from backgrounds which are diverse in study disciplines, language skills, and culture.

References

- Berglunda, A., Eckerdala, A., Pearsa, A., Eastb, P., Kinnunenc, P., Malmic, L., Thomask, L. (2009). Learning computer science: perceptions, actions and roles. *European Journal of Engineering Education*, 34(4), 327 - 338. doi: 10.1080/03043790902989168
- Bloom, B. S. (1956). *Taxonomy of educational objectives, Handbook I: The cognitive domain*. New York: David McKay.
- Crouch, C. H., & Mazur, E. (2001). Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9), 970-977. doi: 10.1119/1.1374249
- Dijkstra, S. (1997). The integration of instructional systems design models and constructivistic design principles. *Instructional Science*, 25(1), 1-13.
- Derry, S. J. (1996). Cognitive schema theory in the constructivist debate. *Educational Psychologist*, 31(3/4), 163-174.
- Glass, Robert L. (1997). Practical programmer: The ups and downs of programmer stress. *Communications of the ACM*, 40(4), 17-19.
- Gholson, B., & Craig, S. D. (2006). Promoting constructive activities that support vicarious learning during computer-based instruction. *Educational Psychology Review*, 18(2), 119-139. doi: 10.1007/s10648-006-9006-3
- Harney, O., Hogan, M., & Broome, B. (2012). Collaborative learning: The effects of trust and open and closed dynamics on consensus and efficacy. *Social Psychology of Education*, 15(4), 517-532. doi: 10.1007/s11218-012-9202-6
- Kleiman, P. (2008). Towards transformation: Conceptions of creativity in higher education. *Innovations in Education and Teaching International*, 45(3), 209-217.
- Kosheleva, O., & Kreinovich, V. (2010). What is the best way to distribute efforts among students: Towards quantitative approach to human cognition. *Applied Mathematical Sciences*, 4(9), 417- 429.
- Krathwohl, D. R. (2002). A revision of Bloom's Taxonomy: An overview. *Theory Into Practice*, 41(4), 212-218. doi: 10.1207/s15430421tip4104_2
- Lister, R., Berglund, A., Box, I., Cope, C., Pears, A., Avram, C., Bower, M., Carbone, A., Davey, B., de Raadt, M., Doyle, B., Fitzgerald, S., Grandell, L., Kutay, C., et al.

- (2007). Differing ways that computing academics understand teaching. *Australian Computer Science Communications*, 29(5), 97-106.
- Lister, R., Box, I., Morrison, B., Tenenberg, J., & Westbrook, D. S. (2004, June). The dimensions of variation in the teaching of data structures. In *ACM SIGCSE Bulletin*, 36(3), 92-96. doi: 10.1145/1007996.1008023
- Mayer, R. E., & Chandler, P. (2001). When learning is just a click away: Does simple user interaction foster deeper understanding of multimedia messages? *Journal of Educational Psychology*, 93(2), 390-397. doi: 10.1037/0022-0663.93.2.390
- Mingfei, L., & Jie, Z. (2010). Study on the mechanisms of team learning upon knowledge transfer: A research based on social constructivism learning theory (pp. 196–200). In *Proceedings, 2010 3rd International Conference on Information Management, Innovation Management and Industrial Engineering*, November, Kunming, China.
- Mota, D., Vaz de Carvalho, C., & Reis, L. P. (2011, April). A conceptual model for collaborative learning activities design (pp. 1026-1031). In *Global Engineering Education Conference (EDUCON) 2011*. doi: 10.1109/educon.2011.5773274
- Olds, S. A., McKenna, A., & Pazos, P. (2007, October). Work in progress - Promoting conceptual understanding through effective peer discussions in large classes (pp. T1D-7- T1D-8). In *Proceedings, Frontiers In Education (FIE) Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007*. doi: 10.1109/FIE.2007.4417908
- Tan, P. H., Ting, C. Y., & Ling, S. W. (2009, November). Learning difficulties in programming courses: Undergraduates' perspective and perception. In *Proceedings, Computer Technology and Development (ICCTD) 2009 (Vol. 1)* (pp. 42-46). doi: 10.1109/ICCTD.2009.188
- Porter, L., Bailey Lee, C., & Simon, B. (2013, March). Halving fail rates using peer instruction: A study of four computer science courses. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (pp. 177-182). doi: 10.1145/2445196.2445250
- Pullan, W., Drew, S., & Tucker, S. (2013, September). An integrated approach to teaching introductory programming. In *Proceedings, e-Learning and e-Technologies in Education (ICEEE), 2013* (pp. 81-86). doi: 10.1109/ICeLeTE.2013.6644352
- Quevedo-Torrero, J. U. (2009, April). Learning theories in computer science education (pp. 1634-1635). In *Proceedings, Information Technology: New Generations, 2009*. doi: 10.1109/ITNG.2009.294
- Requena-Carrion, J., Alonso-Atienza, F., Guerrero-Curieses, A., & Rodriguez-Gonzalez, A. B. (2010, April). A student-centered collaborative learning environment for developing communication skills in engineering education. In *Proceedings, Education Engineering (EDUCON), 2010* (pp. 783-786). doi: 10.1109/EDUCON.2010.5492499
- Sangin, M., Molinari, G., Nüssli, M.-A., & Dillenbourg, P. (2011). Facilitating peer knowledge modeling: Effects of a knowledge awareness tool on collaborative learning outcomes and processes. *Computers in Human Behavior*, 27(3), 1059-1067. doi:10.1016/j.chb.2010.05.032
- Simon, B., Kohanfars, M., Lee, J., Tamayo, K., & Cutts, Q. (2010, March). Experience report: Peer instruction in introductory computing. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 341-345). doi: 10.1145/1734263.1734381

- Simon, B., Parris, J., & Spacco, J. (2013, March). How we teach impacts student learning: Peer instruction vs. lecture in CS0. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 41-46). doi: 10.1145/2445196.2445215
- Smart, K. L., Witt, C., & Scott, J. P. (2012). Toward learner-centered teaching: An inductive approach. *Business Communication Quarterly*, 75(4), 392-403. doi:10.1177/1080569912459752
- Thompson, E., Hunt, L., & Kinshuk, K. (2006, January). Exploring learner conceptions of programming. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52* (pp. 205-211). Australian Computer Society.
- Welkener, M. M. (2013). Understanding the complexities of cognition and creativity to reform higher education practice. *International Journal of Educational Reform*, 22(3), 221-234.
- Wright, G. B. (2011). Student-centered learning in higher education. *International Journal of Teaching & Learning in Higher Education*, 23(1), 92-97.
- Xu, Z., Han, H., Zhang, Y., & Zhang, C. (2008, December). Research and practice on new interactive teaching model based on constructivist learning theory. In *Proceedings, IEEE International Symposium on IT in Medicine and Education*, December 12-14, Xiamen, China (pp.182-186). doi: 10.1109/ITME.2008.4743848

Copyright © 2015 Mary Tom