The Need for a Competency Model of Programming Teachers: a Need Analysis Survey

Nor Masharah Husain[*], Muhammad Modi Lakulu[**], SulaimanSarkawi[**]

[*] Department of Computing, Sultan Idris Education University, Malaysia
[**]Department of Computing, Sultan Idris Education University, Malaysia

Abstract- programming teachers should have appropriate skills or competency to help them impart the knowledge of programming to their students. Further compounding the problem, the subject matter, being a computing course, is subject to continual changes that pose serious challenges to teachers in keeping abreast with new elements of learning contents. Put simply, teachers have to continually enhance their knowledge of the subject to ensure students will receive up-to-date information or knowledge pertaining to programming. Such a case puts a heavy strain on programming teachers, especially the less experienced ones. Against this backdrop, a needs analysis study was carried out to ascertain the requirements needed for the development of a competency model of teachers that can help provide the guidelines and knowledge pertaining to programming teachers. The study was based on a quantitative approach using a semi-structured survey, in which 45 programming teachers were randomly selected from various schools throughout Malaysia. A field survey was performed to yield the frequencies and percentages of the competency items of the research instrument. Overall, a reasonable number of elements of competency construct was determined that can help improve the development of such competency model.

Index Terms- model competency, programming, need analysis, programming teachers

I. INTRODUCTION

Arguably, teaching programming is very challenging, as it demands strong ability and tenacity of computer science educators. More profoundly, programming teachers' competency will have a huge impact on the current pedagogy used by teachers in their classroom. Furthermore, in programming assessment, such competency is vital to ensure effective teaching. In this respect, good assessment can only be carried out by teachers who have strong content knowledge of programming [1].

As highlighted, teachers' competency in teaching programming is important and essential to ensure effective teaching. Clearly, programming teachers' knowledge is different from other subject teachers' knowledge, because the nature of the former is totally different from the latter [2]. Undeniably, programming teachers need to have sound knowledge of the subject matter, of teaching and learning principles, and of pedagogical content knowledge, as characterized by Shulman [30].

Given the needs to have these characteristics, many scholars have expressed their concerns about the quality of teaching programming among such teachers [4].To dispel such concerns, programming teachers must strive to find the optimal teaching approach by which they can overcome many of the problems in teaching programming such that student learning is improved. Therefore, it becomes imperative to identify and to determine the essential components or element of competency that can guide the development of competent programming teachers.

Without doubt, developing such teachers is challenging, given the myriad of complex factors, such as teacher's knowledge, the intricate relations between theoretical and practical aspects of teaching process, and potential problems that may surface during the teaching of a particular programming topic. Together, the problems and challenges in teaching and learning of the subject matter will take a heavy toll on students' interest and confidence. In such situation, programming teachers must intensify their efforts and seek better strategies to help their students who may be overwhelmed in learning programing.

II. RELATED WORK

The literate is replete with previous studies pertaining to the examination and development of a number of learning tools and innovative methods to improve the teaching and learning of programming for both secondary and tertiary level. Nonetheless, the findings of such studies showed that most teachers often failed to use appropriate teaching methodsand teaching strategies [5], which could help improve student learning. Apparently, such failure seemed to stem from the lack of knowledge and skills of the teachers in coping with diverse students' learning styles.

Additionally, the literature discusses aspects relating to the preparation of competent teachers in the field of science computer. The aspects discussed include teachers preparation [5, 6, 7,8] and teachers' pedagogical contents knowledge [9, 10]. Likewise, the discussions also delve into the pedagogical contents knowledge of teachers involved in the teaching of programming [11, 12, 13, 14].

Overall, the findings of these studies help explain and highlight the importance of knowledge that each teacher should have in the context of teaching programming. In short, the findings suggest that programming teachers must be equipped with appropriate knowledge and skills to ensure successful teaching and learning of programming.

In a study by researcher [15], it was found that teachers involved in teaching computing lacked not only knowledge but

also motivation. Furthermore, [16] found that teachers did not focus on appropriate students' learning styles in the teaching and learning process such as to help the latter to think creatively and critically. Based on this study, the researchers assert that teaching styles of teachers will have a huge impact on learning process through which students will be able to improve their creativity and interest in programming. Similarly, past research [16] found that the teaching methods or approaches used by most teachers were quite rigid, which might run counter to the various learning styles of their students. Such teaching approaches may disadvantage certain students, whose learning styles may not be suitable with the type of presentation and delivery of learning contents used by their teachers. Obviously, the lack of pedagogical knowledge or skills among programming teachers will adversely affect students' motivation to learn the subject matter [17].

III. METHODOLOGY

Naturally, the imperative to develop a particular model is driven by the research gap in a particular area of study. As discussed in the previous section, there is an urgent need to develop a competency model of programming teachers that can help measure such teachers' competency and ultimately help provide a set of guidelines with which teachers can be guided in carrying out their teaching activities. In this regard, a needs analysis should be first carried out to ascertain the need for the development of such guidelines. Hence, this study was undertaken using a semi-structured survey, in which 45 programming teachers were randomly selected from various schools throughout Malaysia. Essentially, the main purpose of this survey was to explore and identify relevant elements of teaching competency based on the respondents' feedback and opinions.

A. Participant

The selection of respondents in the first phase of this study was performed using purposeful sampling method. Specifically, the researcher selected such respondents consisting of teachers who are teaching, in general, computer science and, in particular, programming. For this study, the term "programming teachers" refers to both school teachers and lecturers of various universities in Malaysia, who are teaching programming. Forty-five respondents were recruited comprising teachers of public secondary schools and lecturers of several public and private institutions of higher learning.

B. Research Instrument

The development of the research instrument used in this study was guided by the preliminary analysis of the related literature. Essentially, this instrument has three main sections. Section A contains items pertaining to respondents' demographics. Section B comprises items relating to the knowledge of respondents in programming. Section C consists of items regarding the skills of respondents in programming and their agreement with the needs for the development of a competency model of programming teachers. The validation of the instrument was based on expert validation carried out by an academic who is expert and experienced in both fields of education and technology. In the validation process, individual discussion was performed to obtain the expert's opinions and recommendations regarding the questionnaire items in terms of their clarity and coherence and the validity of the constructs. Essentially, each feedback and recommendation were noted and incorporated into the instrument to improve its reliability and quality.

IV. DATA ANALYSIS

The needs analysis study was carried out through a survey using a semi-structured questionnaire. This research instrument was administered online to 45 respondents, consisting of teachers and lecturers to obtain relevant data pertaining to elements of a competency model of programming teachers. The researcher used the quantitative approach to analyse the distribution of such data based on the frequency and percentage of each item of the questionnaire using SPSS. These statistical measures helped highlight and emphasize the needs for the development of the competency model of programming teachers in Malaysia.

V. RESULT AND DISCUSSION

The analysis carried out was intended to identify the basic knowledge of the surveyed teachers in programming and to ascertain the needs for the development of the competency model of programming teachers. The following subsections discuss the findings of this analysis.

A. First Findings: Conceptual Knowledge of Programming

The first set of questions of the needs analysis survey pertained to the contents or conceptual programming knowledge of the respondents. Effectively, this question helped highlight the conceptual programming knowledge of respondents that they would use in their teaching. Such highlight enabled the researcher to develop appropriate themes regarding programming concepts based on respondents' input. Table 1 summarizes the themes of conceptual programming knowledge identified in this survey.

Table 1: The themes of conceptual programming knowledge

Themes	**Percentage (frequency)**
Control Structure	95% (f =43)
Data Structure	40% (f =18)
Variables	37% (f =17)
Array	28% (f =13)
Algorithm	17% (f =8)
Function	15% (f =7)
Constant	15% (f =7)
Logical Thinking	15% (f =7)
Syntax	6.6% (f =3)

The findings of the conceptual programming knowledge showed nine components of such knowledge that were deemed important by the respondents in teaching the subject matter, namely control structure, data structure, variables, arrays, algorithms, functions, constants, logical thinking , and syntaxes, which were based on similar themes of previous studies reported in the literature of programming [16, 18, 19, 20].

B. Second Findings: Pedagogy Knowledge of Programming

The second set of questions of the needs analysis survey focused on the method or approach used by the respondents in teaching programing. More importantly, answers to the questions helped provide the researcher with a clear overview of the method or approach applied in the teaching process by the respondents. Essentially, the respondents' answers assisted the researcher to develop appropriate themes relating to the method of teaching programming. Table 2 summarizes the themes deemed vital in the teaching of programming by the respondents.

Table 2: The themes of pedagogical contents knowledge

Themes	Percentage (frequency)
Encouraging students to work in group	60% (*f* =27)
Providing students with training or questions	46% (*f* =21)
Providing students with simple examples	42% (*f* =19)
Planning effective learning activities for students	28% (*f* =13)
Preparing challenging learning activities for students	15% (*f* =7)
Assessing students' learning performance	13% (*f* =6)
Identifying students' learning problems	11% (*f* =5)

The second part of the survey questionnaire helped the researcher to develop seven specific themes. Nonetheless, the themes were amended slightly to ensure they were aligned with the appropriate elements of pedagogy used in the teaching of programming, as outlined by the objectives and aims of the study, which had been highlighted in the literature.

C. Third Findings: Problem Solving Knowledge

The third set of questions of the survey dealt with the method of problem solving used by the respondents in teaching programing. These questions were posed to the respondents to determine their current knowledge in solving problems associated with the teaching of programming. Table 3 summarizes the themes of this problem-solving construct.

The findings showed that the respondents solved programming problems by analyzing, solving, and interpreting such problems. This particular finding is consistent with other findings such as last research on problem solving [21, 22], indicating the method of solving programming problems begin with interpreting and analyzing such problems accordingly.

Table 3: Problem-solving knowledge

Themes	Percentage (frequency)
Defining problems	53% (*f* =24)
Analsysing problems	93% (*f* =42)
Recommending approriate solutions	84% (*f* =38)

D. Fourth Findings: Difficulties in Teaching Programming

The fourth set of questions of the survey helped the researcher to elicit information regarding the difficulties of respondents in teaching programing. The researcher shortlisted six options for the survey questions, however, one more thematic question was added to take into account other difficulties highlighted by the respondents. Table 4 summarizes the difficulties faced by respondents in the teaching of programming.

Table 4: Difficulties in the teaching of programming

Type of teaching difficulties	Percentage (frequency)
Lack of skills in programming	64% (*f* =29)
Lack of knowledge in programming	37% (*f* =17)
Lack of confidence in carrying out teaching activities of programming	68% (*f* =31)
Lack of enthusiasm in carrying out teaching activities of programming	31% (*f* =14)
Lack of support in performing teaching activities of programming	26% (*f* =12)
Lack of equipment in performing teaching activities of programming	13% (*f* =6)
Lack of interest in programming	11% (*f* =5)

This part of the survey revealed there were seven types of difficulty faced by the respondents. Such findings seem to concur with other findings [23]. In fact, the same researchers concluded that learning programming entailed strong skills and knowledge to overcome such problems. Similarly, researcher [24] found that teaching programming was challenging, but they noticed that teaching structured programming, in particular, was even more problematic. To overcome such problems, the researchers strongly advocate more training sessions to be carried out in group in the computer laboratory.

E. Fifth Findings: Self-Motivation in Teaching Programming

The fifth set of questions of the survey concerned the method or approach used by the respondents to motivate themselves in the teaching activities of programming. For this self-motivation construct, the researcher identified three themes based on the respondents' feedback. Table 5 summarizes the three themes of the self-motivation construct.

Table 5: Themes of self-motivation in teaching programming

Themes	Percentage (frequency)
Learning and making effort to enhance knowledge in programming field	64% (f =29)
Discussing with teachers and experts in solving problems related to programming	26% (f =12)
Empathizing with students in dealing with their learning problems such as to motivate them to learn programming	6% (f =3)

As unanimously agreed by all educators, motivation plays an important role in developing and nurturing the interest and confidence of learners in their academic pursuits. Especially in learning programming — universally accepted as one of the toughest subjects — such psychological construct exerts a strong affective influence on students' confidence to learn not only effectively but also persistently [25].

F. Sixth Findings: Knowledge Skill in Programming

The sixth questions of the needs analysis survey entailed respondents to provide either "yes" or "no" response. The main objective of these questions was to reveal the level of skills or competency of respondents in programming. The analysis of responses of such questions showed a majority of the respondents, at more than 90%, did not have any knowledge with regard to their skills in programming. Arguably, a number of constraints might have impeded their efforts to become competent programming teachers. Hence, coupled with a lack of teaching resources, low competency among programming teachers further exacerbates the difficulties in teaching and learning programming. Clearly, such a finding further reinforces the imperative to develop a competency model of programing teachers.

G. Seventh Findings: The needs for the competency model of programming teachers

The final set of questions of the needs analysis survey focused on the needs for the development of a competency model that can help measure the level of skills or competency of programing teachers. Interestingly, all the respondents unanimously registered their strong desire for such a competency model to which they could refer in measuring their level of skills in programming.Even a study by [26] in 2015 explains that there is no model of teacher competence in computer science.

VI. CONCLUSION

Undeniably, a needs analysis can help highlight current issues surrounding the teaching and learning of school subjects. As demonstrated in this study, programming teachers had to deal with a host of teaching difficulties, which would be better understood by having a suitable competency model of programming teachers. As such, the researcher carried out a needs analysis survey involving a group of programming teachers to collect important information that helped determine appropriate components or elements of the competency construct. The findings of the needs analysis suggest that programing teachers in Malaysia urgently need a competency model to help guide them in determining the important competency elements of programming that they need to possess and master. In addition, such a competency model can help determine and measure existing levels of programming competency of such teachers.

REFERENCES

[1] Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2012a). *Portray the Pedagogical Content Knowledge of Programming: The technical report*.

[2] Erdogan, Y., Aydin, E., & Kabaca, T. (2004). Exploring the Psychological Predictors of Programming Achievement. *Journal of Instructional Psychology*, *35*, 264–271.

[3] Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, *15*(2), 4–14.

[4] Kushan, S. B. (1995). Characteristics of a Good Programming Teacher. In *National Educational Computing Conference (NECC)* (pp. 42–47). Baltimore, MD. Retrieved from http://files.eric.ed.gov/fulltext/ED392415.pdf.

[5] Gal-ezer, J., & Zur, E. (2013). What (else) should CS Educators Know ? – Revisited. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education (WiPSCE)* (pp. 83–86). Aarhus, Denmark.

[6] Hazzan, O., Gal-ezer, J., & Ragonis, N. (2010). How to Establish a Computer Science Teacher Preparation Program at Your University? In *The ECSTPP Workshop* (Vol. 1, pp. 35–39). New York, USA.

[7] Margaritis, M., Magenheim, J., Hubwieser, P., Berges, M., Ohrndorf, L., & Schubert, S. (2015). Development of a Competency Model for Computer Science Teachers at Secondary School Level. In *IEEE Global Engineering Education Conference (EDUCON)* (pp. 211–220). Tallinn, Estonia: ACM New York, USA.

[8] Schubert, S., Nelles, W., Linck, B., & Ohrndorf, L. (2013). Competence model for informatics modelling and system comprehension. In *IEEE Global Engineering Education Conference (EDUCON)* (pp. 85–93). Berlin, Germany.

[9] Driel, J. H. Van, Verloop, N., & Vos, W. De. (1998). Developing Science Teachers ' Pedagogical Content Knowledge. *Journal of Research in Science Teaching*, *35*(2).

[10] Ioannou, I., & Angeli, C. (2013). Teaching Computer Science in Secondary Education : A Technological Pedagogical Content Knowledge Perspective. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education (WiPSE)* (pp. 1–7). Aarhus, Denmark.

[11] Rolandsson, L. (2010). Tell me about Pedagogy in programming! (p. 2010).In *International Computing Education Research (ICER)*. Aarhus, Denmark.

[12] Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2011). Teaching Programming in Secondary School : A Pedagogical Content Knowledge Perspective. *Informatics in Education*, *10*(1), 73–88.

[13] Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2012b). Programming : Teachers and Pedagogical Content Knowledge in the Netherlands. *Informatics in Education*, *11*(1), 81–114.

[14] Schulte, C. (2013). Reflections on the role of Programming in Primary and Secondary Computing Education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education (WiPSE)* (pp. 17–24). Aarhus, Denmark.

[15] Hubwieser, P., Armoni, M., Dagiene, V., Diethelm, I., Giannakos, M. N., Knobelsdorf, M., … Schubert, S. (2011). Computer Science / Informatics in Secondary Education. In *Proceedings of the 16th annual conference reports on Innovation and technology in computer science education (ITiCSE)* (pp. 19–38). Darmstadt, Germany.

[16] Gomes, A., & Mendes, A. J. (2007). An environment to improve programming education. In *International Conference on Computer Systems and Technologies - CompSysTech* (pp. 1–6). Rousse, Bulgaria.

[17] Makris, D., & Giannakos, M. N. (2013). Could you help me to change the variables ? Comparing instruction to encouragement for teaching programming. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education (WiPSCE)* (pp.79–82).). Aarhus, Denmark.

[18] Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. In *Proceedings of the 24th ascilite Conference* (pp. 99–107). Singapore.

[19] Govender, I., & Govender, D. W. (2012). A constructivist approach to a programming course: Students' responses to the use of a Learning Management System. *African Journal of Research in Mathematics, Science and Technology Education*, *16*(2), 238–252.

[20] Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and First Results of a Psychometric Test for Measuring Basic Programming Abilities. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education (WiPSCE)*.

[21] Dwiyogo, W. D. (2016). Teaching and Learning Process : Thinking and Problem Solving. *The Online Journal of New Horizons in Education*, *6*(3), 121–129.

[22] Sprankle, M., & Hubbard, J. (2012). *Problem Solving and Programming Concepts* (9th ed.). Pearson.

[23] Hofuku, Y., Cho, S., Nishida, T., & Kanemune, S. (2009). Why is programming difficult ? In *ISSEP '13: 6th International Conference on Informatics in Schools: Situation, Evolution and Perspectives* (pp. 13–24).

[24] Ala-mutka, K., & Lahtinen, E. (2005). A Study of the Difficulties of Novice Programmers. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE)* (Vol. 37, pp. 14–18).

[25] Enstr, E. (2013). Dynamic programming – structure , difficulties and teaching. *IEEE Frontiers in Education Conference (FIE)*, 1857–1863.

[26] Seitz, C., & Zendler, A. (2015). Process-related competence areas to computer science education: An empirical determination. *International Journal of Research Studies in Computing*, *4*(1), 1–12.

AUTHORS

First Author – Nor Masharah Husain, Ph.D Student, Department of Computing, Sultan Idris Education University, Malaysia. masharah.husain@gmail.com.

Second Author – Assoc. Prof. Muhammad Modi Lakulu, Associate Professor, Department of Computing, Sultan Idris Education University, Malaysia. modi@fskik.upsi.edu.my.

Third Author – Dr. SulaimanSarkawi, Senior Lecturer Department of Computing, Sultan Idris Education University, Malaysia, sulaiman@fskik.upsi.edu.my.