How to Cite:
Oli, M. C. (2021). Proficiency amidst COVID-19 challenges in developing computer programs: An outcomes-based approach in discrete structures. *Linguistics and Culture Review*, *5*(S3), 770-783. https://doi.org/10.21744/lingcure.v5nS3.1593

Proficiency Amidst COVID-19 Challenges in Developing Computer Programs: An Outcomes-Based Approach in Discrete Structures

Mario C. Oli
University Faculty, College of Information and Computing Sciences, Cagayan State University, Carig Campus Tuguegarao City, Cagayan Valley, Philippines

***Abstract*---**The educational system has been greatly affected by the COVID-19 pandemic which lead to the cancellation of face-to-face classes from different learning institutions in the whole world. Using the descriptive and phenomenological approach, this study determined the programming proficiency level of sophomore computer science students where laboratories were not available and online classes through learning management system was just one of the options to deliver the lessons to the students. It determined the proficiency level of the students across the four indicators (program design, program execution, specification satisfaction and coding structures). Findings revealed that the students have an "advanced skills in computer programming and application" across the different evaluative indicators. It was also found out that the proficiency levels of students based from their groupings do not vary which gave meaningful output of their project. It further manifested that their main concerns or problems in developing the programs are lack of programming skills, insufficient knowledge in programming, weak internet connection, communication barriers among members and the passive participation of members of the group.

***Keywords*---**congruence modulo relation, discrete structures, outcomes-based education, programming proficiency.

Introduction

The proficiency of students in computer programming from the degree courses of computer science, information technology, and information system is one of the measures of their learnings during their college years' preparation. Their ability in programming is one of the most essential and basic skills that they have to possess especially students in computer science (Guowu et al., 2021; Rahmat et al., 2012). Prior to developing and designing a computer system, a software or a

***Corresponding author**: Oli, M. C.; Email: mariooli299@gmail.com*
Manuscript submitted: 18 July 2021, Manuscript revised: 09 Sept 2021, Accepted for publication: 20 Oct 2021

webpage, one must be skilled in computer programming (Truman & Elliott, 2019; Nanz et al., 2013).

Programming

Programming requires a lot of thinking as it follows a particular order and sequence with the objective of executing a given instruction [3]. This sequential process of programming, called, algorithm, is usually taught at the early stage of computer degree program at any higher institution of learning, mostly as Introduction to Computer (CSC101) or Introduction to Programming (CSC102). As to the mastery of the rudiments of programming Guowu et al. (2021), students in this course must have to master the basic knowledge of program design, data structure and algorithm, so that they can to select the most appropriate programming language, precisely and proficiently complete the programming and debugging work for a given problem, and achieve the expected goal of the program (Miki, 1978; Tranberg et al., 1993).

Outcomes-based education by the commission on higher education

The outcomes-based education enables students to display or demonstrate what they know and what they can do that can be observed and measured. The Commission on Higher Education(CHED), no. 25, series of 2015 known as the "Revised Policies, Standards and Guidelines (PSG) for Bachelor of Science in Computer Science" clearly implements the shift to learning competency-based standard or outcomes-based education to the Higher Education Institutions (HEI's) in the Philippines (CMO. No 46, s.2012). This PSG prepares HEI's to integrate and adopt outcomes-based education (OBE) in their curriculum and assess how to best achieve the intended learning outcomes (ILO) of the program (Andrich, 2002; Myer, 1999).

The Commission (CMO No. 25, s. 2015) presents the minimum standards for each program which are expressed as minimum sets of desired program outcomes using a learner-based approach or outcomes-based approach. Article IV section 6.3 discusses the 10 specific attributes and outcomes with codes from CS01 up to CS10. As the students proceed to the next year level, the graduate attributes and graduate outcomes also move to higher level. CS01 is the CS10 is the highest level having a graduate attribute of "Life-long Learning" with an outcome of recognizing the need, and have the ability, to engage in independent learning for continual development as a computing professional. This is the attribute and outcome that a graduate of computer science program must possess. So as the year level progresses, the attributes and expected outcomes also progress (Möhring & Radermacher, 1984; Balogh et al., 2015).

Educational system during COVID-19 pandemic

In this time of COVID-19 pandemic, everything has been greatly affected and educational systems is no excused. Learning institutions cancelled face-to-face classes Hassan (2021), as one of the measures to prevent the spread of the virus among individuals in the classroom. Different learning systems have been used as synchronous or asynchronous. Higher education institutions embraced the new way of teaching the course just to deliver with quality the intended lessons for the subject. Likewise, students have to adjust on the way lessons are delivered and on how to learn by themselves through a printed learning material or online readings and discussions (McIntosh Jr & Ashley, 1978; MacRae, 1982).

Learning theories in developing computer programs

The constructivism, connectivism and experiential are the learning theories relevant in developing any computer programs. These learning theories help students create or develop the computer program because they have to create their learning as a group Hein (1991), based from their previous experiences using the concepts of programming and various activities which were provided in their first year in the course. Applying the knowledge and skills gained in learning concepts on congruence modulo relation and their knowledge and skills Siemens (2005), obtained during this year help them conceptualize the computer programs which they have to develop. The actual development of computer programs Zou & Brown (2015); Lam & Chan (2013), through the application of their skills in programming and in solving problems involving congruence modulo relations manifest the level of their learnings in the different fields (Villa et al., 2021; Oli, 2021).

Computer science program curriculum

As part of the curriculum of the Computer Science program, Discrete Structures is one of the courses that CHED prescribed for the program. There are three samples of minimum learning outcomes (LO) set for the course: LO1-Solve real-world computing problems that require mapping to permutations, combinations of a set, and modular arithmetic, LO2: Compute the event probabilities using counting and Bayes Theorem, and LO3: Solve equations involving recurrence and relate them to recursive algorithms (CMO No. 25, s. 2015). With these two sets of outcomes in outcomes-based approach as mandated by the Commission (CMO No. 25, s. 2015; CMO No. 46, s.2012), a major requirement on the development of a simple program using the concepts of modular arithmetic specifically congruence modulo relations has been conceptualized. The project aims to show to students the significance of the course to the development of a certain computer program (Zulvany, 2020; Larantika et al., 2017).

In the case of sophomore computer science college students in the Philippines, programming 1 and 2 and data structures and algorithms were their major subjects during their first year and second year (1st semester), respectively. From these course, they were trained to design, implement, test, and debug programs intended to solve computing problems using basic programming constructs (CMO No. 25, s.2015). Additionally, they were introduced to the data representation and algorithms to solve computing problems efficiently taking into consideration space requirements and time complexity of algorithm. Further, about two or three different computer programming languages were also tackled in their major subjects. Moreover, concepts of basic mathematics such as even and odd numbers, composite and prime numbers, divisibility rules, together with the congruence modulo relations were introduced to them in their class on Discrete Structures (DS) 2. So that through the congruence modulo relations, varied applications of the concept can be seen in the field of computer science such as hash functions, cryptography, pseudorandom numbers, barcoding and checking digits on International Standard Book Number (ISBN). This is the reason of having computer programs as one of their major requirements course requirements in DS 2- development of computer program using congruence modulo relation and agrees with the findings of a study Babas (2020), that younger students tend to have a higher competency level in computer programming. Through their experiences and output, the activity only manifests its adherence and support to the program of the Commission on Higher Education (CHED) on their mandate for outcomes-based education (Citrawan et al., 2018; Nyandra et al., 2018).

Problems/issues in programming

Designing and developing a computer program by college students without the actual face-to-face classroom interactions with the instructors have been difficult. Students may encounter various problems and challenges such as the internet connection Hassan (2021), communications with their peers and instructors, gadgets to use and sufficiency of basic knowledge in computer programming, thus, their proficiency is affected especially so that they major subjects in computer science must be conducted with laboratories (Hassan, 2021). From a study conducted Rahmat et al. (2012), problems such as the reference materials, lectures and lab session approaches, problem solving ability, time management and self-confidence were found as major problems in basic programming that influences student performance. In order to face the challenges brought by the COVID-19 pandemic and on rapid development and advancement of technology, which greatly affects our lives, the Commission recognizes the need to be responsive to the current needs of our nation. Hence, the country's computing capability must be continually developed and strengthened to be at par globally.

Computer Programming requires understanding of basic mathematics and one needs a thorough understanding of mathematics and arithmetic to become proficient. Undeniably, mathematics plays a vital role in computer programming for it serves as the foundation of developing and designing any computer programs. Thus, this study aimed to determine the level of proficiency in programming across the four evaluative criteria of the second year students in computer science program during the first semester of school year 2020-2021 and tested the difference on the proficiency levels of students grouped according to blocks. It also thematically explored the problems or challenges encountered by the students in developing the programs during the first year of the COVID-19 Pandemic lockdown. In order to maximize their learnings from the math course and other major subjects they have taken and to determine the extent of their knowledge and skills especially on the applications of concepts and theories they have obtained, they were greatly challenged to develop and run a basic program on the identified topics in DS 2 as mentioned earlier. The main purpose of this study.

Methodology

Research design

Descriptive and phenomenological approach were utilized to carry out this study. The descriptive part was manifested by the used of the scoring rubrics in assessing the computer programs while the phenomenological aspect was employed in analyzing the narratives of the respondents on the problems encountered in developing the programs during the time when face-to-face teaching and learning was cancelled.

Research environment

This study was conducted via online to the sophomore Computer Science Students of the College of Information and Computing Sciences of Cagayan State University, Carig Campus, Tuguegarao City, Cagayan Valley, Philippines during the first semester of Academic Year 2020-2021.

Samples and sampling technique

The data were provided by the 24 groups of students (85 individuals) from three blocks (sections) in the program. The students in each section were asked to form a group of 3 members for more effective and efficient collaborations among them (Schwarz et al., 2021). This number is ideal for a small group discussion/project because each of the members is obliged to work and is forced to contribute Lam & Chan (2013), something to the success of the project. The students were given the freedom to choose their own group mates and they were able to form such group due to geographical locations, strength of relationships and ease of communications especially so that they were not allowed to move out from their residences because of the COVID-19 pandemic protocols. This type of choosing group mates by themselves may greatly enhance their communications, confidence and self-worth because they come from the same places, same interests and openness of communication with each other.

Instruments

The researcher utilized a 6 x 4 rubrics being used in evaluating programs developed by students in the college. The rubrics has been used to evaluate or assessed the programs developed by students in the program. In the utilization of the rubrics in the study, its content was first referred to the different program chairs in the college who are experts on the fields prior. Through the comments of these experts, the instrument was then finalized and prepared for use in evaluating the proficiency of the students for this study. The 6 x 4 rubrics consists of six indicators or criteria and four numerical ratings. The indicators are program design, program execution, specification satisfaction, coding style, comments, and extra credit while the ratings composed of 4 – very good, 3-good, 2-poor and 1-very poor. Each cell in the 6 x 4 matrix rubrics has its own descriptive definition. In the interpretation, the following scales were used: advanced (3.25-4.00), intermediate (2.5-3.24), novice (1.75-2.49) and basic (1.0-1.74), respectively.

Data gathering procedure

The development of the computer program was on the onset of the final term of the first semester of the Academic Year 2020-2021. Three sections in the second year level under the program Bachelor of Science in Computer Science were involved in the study. These students were under the sole teaching supervision of the researcher in the Discrete Structures course. Every section was grouped into eight and had the freedom to choose any one from these eight different topics such as cryptography, pseudorandom numbers, hash functions, even and odd numbers, divisibility rules, ISBN/UPC, Bar coding, and prime and composite numbers.

During the inception of developing the program, students were informed of the concepts in the course to be utilized, the Congruence Modulo Relations. There are various applications of congruence modulo but they were only limited to the eight topics presented to develop and run a complete simple computer programming. However, each group provided a feedback of their works done and the extent of their accomplishments in relation to the project. Below is the process on how the students were able to successfully presented their outputs:

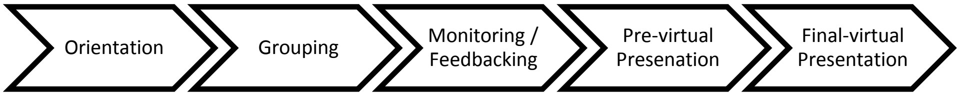

Figure 1. Process of data collection

Quantitative data were obtained through the evaluations of the invited experts in computer programming as companions of the subject professor-the researcher. These experts were program chairs in the different programs in the College. The students presented their developed program through a Learning Management System (LMS) during their scheduled classes. The qualitative information about the problems/challenges encountered form part of their research paper (documentation) and were organized through thematic analysis.

Data analysis

The mean, standard deviations and percent forms were used to analyze the data. These described the proficiency level of students in developing the computer programs. Analysis of Variance (ANOVA) was utilized to compare the proficiency levels of students when grouped according to blocks. Thematic analysis was applied to code the important details from the problems based from the narratives of the students until the main themes were organized.

Results and Discussion

Proficiency level of students in computer programming

The computer programs designed and developed by the students together with the statistical values under the four evaluative indicators and the corresponding proficiency levels are presented in table 1.

Table 1
Proficiency level of respondents along the four indicators

Computer Program	Statistical Tools	Program Design	Program Execution	Specification Satisfaction	Coding Style
Hash Function	Mean	3.11	3.33	3.22	3.33
	Std. Deviation	.333	.500	.441	.500
	Proficiency Level	ICA	ADA	ICA	ADA
Cryptography	Mean	3.56	3.78	3.67	3.67
	Std. Deviation	.726	.441	.500	.500
	Proficiency Level	ADA	ADA	ADA	ADA
Pseudorandom Numbers	Mean	3.11	3.44	3.44	3.56
	Std. Deviation	1.054	.527	.726	.726
	Proficiency Level	ICA	ADA	ADA	ADA
Bar Coding	Mean	3.56	3.56	3.44	3.78
	Std. Deviation	.527	.527	.527	.441
	Proficiency Level	ADA	ADA	ADA	ADA
ISBN	Mean	3.22	3.11	3.11	3.56
	Std. Deviation	.441	.601	.601	.527
	Proficiency Level	ICA	ICA	ICA	ADA
Even and Odd	Mean	2.78	3.33	3.33	3.44
	Std. Deviation	.667	.500	.500	.527
	Proficiency Level	ICA	ADA	ADA	ADA
Divisibility Rules	Mean	3.33	3.67	3.78	3.67
	Std. Deviation	.707	.707	.441	.707
	Proficiency Level	ADA	ADA	ADA	ADA
Prime and Composite	Mean	3.33	3.67	3.56	3.67
	Std. Deviation	.707	.500	.527	.500
	Proficiency Level	ADA	ADA	ADA	ADA
Total	Mean	3.25	3.49	3.44	3.58
	Std. Deviation	.687	.556	.554	.550
	Proficiency Level	ADA	ADA	ADA	ADA

Legend: 1-1.74 – Fundamental skills (FS); 1.75-2.49 - Basic Computing and Applications (BSA); 2.5-3.24- Intermediate Computing and Applications (ICA); 3.25-4.0- Advanced Computing and Applications (ADA)

It can be seen in the table that students who developed and designed computer programs on cryptography, bar coding, divisibility rules, and prime and composite manifested advanced computing and applications" skills with means values ranging from 3.35-4.0 in all of the four indicators: program design, program execution, specification satisfaction and coding styles. Moreover, some of them are also advanced in their computing and applications skills along program execution, specification satisfaction and coding styles in their programs developed such as pseudorandom numbers, and even and odd numbers. Meanwhile, others are intermediate in their computing and applications skills when it comes to designing their programs on hash functions, pseudorandom numbers, International Standard Book Number (ISBN), and in even and odd numbers.

Though their proficiency level under programming design is within the range of advanced computing and applications, the mean score is the lower limit in the range and this could become lower if appropriate actions and remediation would not be made. This means that the mastery of making a design of computer program is one of the challenges they have experienced in which the same was emphasized Guowu et al. (2021), in order to proficiently complete the program precisely. In the end, students were still able to manage to develop and design their projects using the congruence modulo relations as they have advanced skills in computing and applications in all the different indicators across the different programs.

Comparison on the proficiency level of students across indicators

The proficiency level of respondents according to groups and the differences are compared in table 2. It includes the mean values and standard deviations with the corresponding proficiency level are presented under the extent and the F-values and probability values together with the remarks are provided under their differences. Students in blocks A and B have advanced skills in computing and applications in terms of their abilities in designing a program with mean values of 3.33 and 3.38c, respectively, while students in block C are intermediate with mean score of 3.04. Moreover, all of the students from the three different blocks/sections demonstrated advanced skills in computing and applying in executing their programs, satisfying the specifications through their styles in coding.

Table 2
Comparison on the proficiency level of students in the across indicators

Indicators	Blocks	Extent of proficiency Level			Difference on Proficiency levels		
		Mean	SD	Proficiency level	F	Sig.	Remarks
Program Design	A	3.33	.702	ADA	1.711	.188	Not Significant
	B	3.38	.576	ADA			
	C	3.04	.751	ICA			
Program Execution	A	3.46	.588	ADA	.576	.565	Not Significant
	B	3.42	.584	ADA			
	C	3.58	.504	ADA			
Specification Satisfaction	A	3.46	.509	ADA	.311	.734	Not Significant
	B	3.38	.576	ADA			
	C	3.50	.590	ADA			
Coding Style	A	3.67	.565	ADA	.406	.668	Not Significant
	B	3.54	.509	ADA			
	C	3.54	.588	ADA			

Legend: 1-1.74 – Fundamental skills (FS); 1.75-2.49 - Basic Computing and Applications (BSA); 2.5-3.24- Intermediate Computing and Applications (ICA); 3.25-4.0- Advanced Computing and Applications (ADA)

In this regard, the computed probability values of the four indicators are greater than the set level of significance which is 0.05. Hence, it be said that proficiency level of the students from the different blocks are not statistically different as per results of analysis. This only implies that their proficiency levels in developing and designing a computer program with the integration of the concepts of congruence modulo relations have been consistent across the different indicators and that they have most likely the same levels.

Proficiency level of the students according to block

Table 3 shows the overall proficiency of Computer Science second year students in developing computer programs using the congruence modulo relations. They were grouped according to block or sections given the group mean values and the qualitative description on the level of their proficiency. It revealed that despite the pandemic being experienced where students learn their lessons online, do their course works online and submit their outputs online, these students have still managed to have computer programming proficiency level of advanced computing and applications" skills as.

Table 3
Overall proficiency of computer science second year students in programming

Block	Proficiency Score (Mean Value)	Standard Deviation	Remarks
Block A	3.46	0.25	Advanced Computing and Applications
Block B	3.36	0.32	Advanced Computing and Applications
Block C	3.45	0.33	Advanced Computing and Applications

Manifested by their group mean values of 3.46, 3.36 and 3.45 respectively. Aside from having the highest proficiency score, block A is the most consistent among the three blocks in their proficiency as described by their standard deviations because it has the lowest value which is 0.25. The result only shows that the skills of the students gained in their computer programming in their first year level from the course was still operative that they were able to apply in developing such computer programs.

Challenges encountered by the students in developing the programs

Table 4 presents the common problems and challenges or difficulties that the students have encountered in developing their computer programs. It includes the frequency so that it could be determined the number of times a problem or issue has been mentioned. These identified problems or issues were themed from their narratives which is part of the documentation of their projects. With the 110 identified different and related issues in developing the programs by the students, the lack of programming skills is their most common problem which comprises 59.09% (f = 65), followed by their insufficient knowledge in programming with 15.45% (f =17) , weak internet connection of 10% (f = 11). They had also encountered problems on how to effectively communicate with each other in the group to discuss their ideas, suggestions and comments for the improvement of the project having 6.36% (f = 7), and the lack of tools or gadgets, with 5.45% (f =

6) like laptop or desktop computers where they make their projects. Moreover, there was also an issue or problems on their attendance to meeting or their availability in doing their projects but not so much of a problem because it was 2.73% (f = 3) of the responses so with their knowledge in different programming languages and on cellphone loads.

Table 4
Issues and problems encountered by the students in developing the computer programs

Problems/Issues	Freq	Percent
Loads	1	0.91
Communication	7	6.36
Internet connection	11	10.00
Knowledge in programming	17	15.45
Participation of members	3	2.73
Programming skills	65	59.09
Tools	6	5.45
Total	110	100

Programing skills

The C++ programming language is the most common language used by students in developing their projects. It was commonly used because of their familiarity of using it. In the beginning, students found difficulty on how to start making the programs especially so that they only have limited knowledge in developing a computer program. Simple as it may seem but they lack the skills in programming. Being the most common problem encountered by the students, the following are the highlighted issues or problems:

- Programming skills are concerned with the appropriateness of algorithm used which lead to correct and accurate coding structures.
- Knowledge in making the design.
- Insufficient knowledge on how to create codes.
- Incompatibilities of codes.
- Capabilities in handling and debugging.
- Errors in coding.

As one of the groups narrated in their documentation, it says:

> R5:
> "Before we develop the program, we struggled in making an algorithm. We tried many algorithm and we only choose the most accurate. During the development, we encountered several problems like getting an error on our codes, sometimes the program will crash. We spend a lot of time in debugging our codes. After we finish the program, we are now ready to test if it is accurate, but this gave us a problem that sometimes it printed the missing digit as 10, we know that 10 is not a digit anymore, we search for that problem and found that the digit 10 replaces to an X, because the roman numerals of 10 is X. We revise the codes to include that, and good thing that we finished it."

The mastery of programming skills which greatly affected by their coding abilities in order to efficiently execute an algorithm Guowu et al. (2021), has a very essential function to run a program. From the narratives of group 17 (R17), the group members mentioned about their problems on their programming skills which focused on coding and algorithm, and it states,

> R17:
> "Formulating the algorithm of the basic function of the program, coding the formulated algorithm, difficulty in coding the decrypt button, difficulty in error trapping, formulating a new algorithm for the new features of the program, coding the new algorithm, and difficulty in executing the new features."

Knowledge in computer programming

The students' knowledge (theory) in computer programming plays a vital role in developing their programming skills. Due to insufficient knowledge in coding which resulted to incompleteness of significant features in the program, the developed program was not related to the topic and lack of resources to get enough details on how to make a computer program were only few indicators that manifest the amount of knowledge in creating the project. According to members of group 13 where they chose to develop a computer program about the International Standard Book Number (ISBN), they admitted that their main concern is their lack of knowledge in programming. Below is their actual statement derived from their documentation, it says:

> R13:
> "There are several problems and issues with our research such as that is does not show what is the check number (last digit) due insufficient knowledge of programming, and it only verifies whether the ISBN is an actual ISBN and does not identify what book is that, and also due to the simplicity of our output it does not much give an impact to the readers and it only gives off a sense of plainness, which we admit for lack of knowledge in programming."

Internet connectivity and communication

Another problem that challenged the students in pursuing their programming projects is the stability of their internet connections. A number of them live outside the City proper, they claimed that they had an issue on their connection with the internet. Having a weak internet connection could impede their activities in searching for resources where they could augment their knowledge through the internet. Because during online learning, the internet is the only one who could immediately answer their queries on certain topics due to limited resources. Also, the instability of internet connectivity affected their communication with their group mates. Students could hardly express their thoughts on what to do and how to do the projects through text messaging or short message services (SMS). Aside from the weak internet connections affecting their communications, the availability of members attending a short virtual meeting is likely an issue because some of them have responsibilities to attend to their families and jobs. Based from the report of members of group 15, it says:

> R15:
> ”In conducting our program, we have encountered some issues or problems. First problem we encountered is our availability. Due to this pandemic, we cannot

> physically communicate with each other so we only have social media's to talk about our program but we also barely get the chance to talk as one because we have different responsibilities in our own home and because of the poor signal. Therefore, we came up with a decision that if one or two of us are online, we are going to leave a message in our group chat regarding our program so that we can at least share our knowledge or suggestions about it."

Tools

A desktop computer or a personal laptop is a need when enrolling into a computer science course. It must be a necessity because it is very much needed in making a program. Having none could post a problem on skill towards program creation. This was one of the issues and challenges of the students in this study. Some groups mentioned that "not everyone is equipped with the tools needed to do the work, e.i. laptop), "sometimes the laptop they used do not function properly which made them do the program" and "no gadget to use to test the program". Contrary to the findings from the study conducted Rahmat et al. (2012), problems such as the reference materials, lectures and lab session approaches, problem solving ability, time management and self-confidence were not found to be the major problems of the students in the current study. Instead, their skills in programming play the biggest contribution that affect their output in developing a computer program. However, this problem has been resolved as they persistently look for other sources to augment their knowledge.

Conclusion

This study is based on the mandate of the Commission on Higher Education regarding the outcomes-based education. The learning experience on congruence modulo relations with a project integrating computer programming has contributed a significant learning outcomes to the students. Developing and designing a computer program is really challenging especially when these were created with numerous problems and issues encountered due to pandemic such as COVID-19 disease. But these students did not give up, instead they were challenged to overcome all of these insufficiencies and more determined to do the project despite the many problems and challenges that were encountered during this pandemic.

Because of their interests, enthusiasm, self-discipline, resourcefulness and open-mindedness, they were able to successfully develop and present their programs. The learnings they have obtained in developing these programs could be of great help in making a complex one in preparation for their capstone project in their higher years. The problems, brought by the pandemic where face-to-face classes were suspended, on lack of skills and insufficient knowledge in programming, weak internet connectivity, communication gaps, lack of tools to use and availability of members for a meeting were all surpassed as they have manifested an "advance skills in programming and applications".

Recommendations

This study was conducted during the first year of COVID-19 pandemic where face-to-face classes were suspended. This means that there was still a year where students attended classes within the four walls in the classroom. It is then recommended that a similar study may be conducted to determine the proficiency level of students in programming in which programming subjects were delivered online (no actual laboratory meetings conducted). Other studies could also be done on the emergence of mathematics to computer programming at an early years of the computer science program to expose students to programming activities aside for the activities given during class discussions. With this, programming skills of the students will definitely be enhanced and that would inspire them to do complex programming activities.

Acknowledgments

The author wishes to recognize the assistance extended by the Faculty experts for the Computer Science Department of College of Information and Computing Sciences who served as panel of evaluators during the online presentation of programs. The Faculty experts were Prof. Lara Mae Balisi (Program Chair-Computer Science), Prof. Claire Anne Bautista (Program Chair-Information System), Prof. Jayrald Naraja, Prof. Ma. Cristina Semilla, Prof. Yssandre Zinampan, and Prof. Sherwin Caacbay. And of course, the three sections (CS 2A, CS 2B, CS 2C of First Semester, AY 2020-2021 of the second year in the program Bachelor of Science in Computer Science who did not give up despite all the challenges they had encountered.

References

Andrich, D. (2002). Implications and applications of modern test theory in the context of outcomes based education. *Studies in educational evaluation*, *28*(2), 103-121. https://doi.org/10.1016/S0191-491X(02)00015-9

Babas, J. B. (2020). Programming Competencies of Filipino Information Technology Students: Inputs to Improving Instructional Processes.

Balogh, J., Das, S., Delcourt, M., Liu, H., & Sharifzadeh, M. (2015). Intersecting families of discrete structures are typically trivial. *Journal of Combinatorial Theory, Series A*, *132*, 224-245. https://doi.org/10.1016/j.jcta.2015.01.003

Citrawan, I. W., Widana, I. W., & Suarta, I. M. (2018). Education sector optimization for improving human development index. *International Journal of Social Sciences and Humanities*, *2*(1), 117-133. https://doi.org/10.29332/ijssh.v2n1.102

Guowu, X. U., Yu, L. U. O., Bingyan, C. H. E. N., Faming, G. U. A. N., Xiaoyun, L. U. O., & Mengxi, C. E. N. (2021). Analysis and Wind Tunnel Test on the Influence of Boattail Design on Aircraft Base Drag. *Chinese Quarterly of Mechanics*, *42*(1), 149.

Hassan, M. (2021). Online Teaching Challenges during COVID-19 Pandemic. *International Journal of Information and Education Technology*, *11*(1).

Hein, G. (1991). Constructivist learning theory. *Institute for Inquiry. Available at:/http://www. exploratorium. edu/ifi/resources/constructivistlearning. htmlS.*

Lam, B. H., & Chan, H. L. (2013). Experiential Learning.

Larantika, A. A. A. D., Zauhar, S., Makmur, M., & Setyowati, E. (2017). Collaboration as a strategy for poverty alleviation. *International Journal of Social Sciences and Humanities*, *1*(3), 40-48. https://doi.org/10.29332/ijssh.v1n3.51

MacRae, C. D. (1982). Urban housing with discrete structures. *Journal of Urban Economics*, *11*(2), 131-147. https://doi.org/10.1016/0094-1190(82)90024-9

McIntosh Jr, S. C., & Ashley, H. (1978). On the optimization of discrete structures with aeroelastic constraints. *Computers & Structures*, *8*(3-4), 411-419. https://doi.org/10.1016/0045-7949(78)90186-4

Miki, H. (1978). A relation between Bernoulli numbers. *Journal of Number Theory*, *10*(3), 297-302. https://doi.org/10.1016/0022-314X(78)90026-4

Möhring, R. H., & Radermacher, F. J. (1984). Substitution decomposition for discrete structures and connections with combinatorial optimization. In *North-Holland mathematics studies* (Vol. 95, pp. 257-355). North-Holland. https://doi.org/10.1016/S0304-0208(08)72966-9

Myer, S. A. (1999). Outcomes-based education in a critical care nursing course. *Critical care nursing clinics of North America*, *11*(2), 283-290. https://doi.org/10.1016/S0899-5885(18)30169-2

Nanz, S., Torshizi, F., Pedroni, M., & Meyer, B. (2013). Design of an empirical study for comparing the usability of concurrent programming languages. *Information and Software Technology*, *55*(7), 1304-1315. https://doi.org/10.1016/j.infsof.2012.08.013

Nyandra, M., Kartiko, B.H., Susanto, P.C., Supriyati, A., Suryasa, W. (2018). Education and training improve quality of life and decrease depression score in elderly population. Eurasian Journal of Analytical Chemistry, 13(2), 371-377.

Oli, M. C. (2021). Proficiency amidst COVID-19 challenges in developing computer programs: an outcomes-based approach in discrete structures. *Linguistics and Culture Review*, *5*(S3). https://doi.org/10.21744/lingcure.v5nS3.1593

Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., & Ab Rahman, R. (2012). Major problems in basic programming that influence student performance. *Procedia-Social and Behavioral Sciences*, *59*, 287-296.

Schwarz, B. B., Swidan, O., Prusak, N., & Palatnik, A. (2021). Collaborative learning in mathematics classrooms: Can teachers understand progress of concurrent collaborating groups?. *Computers & Education*, *165*, 104151.

Siemens, G. (2005). Connectivism: A Learning Theory for the Digital Age http://www. elearnspace. org. *Articles/connectivism. htm (Accessed: 01/03/2007)*.

Tranberg, M., Slane, S., & Ekeberg, S. E. (1993). The relation between interest congruence and satisfaction: A metaanalysis. *Journal of Vocational behavior*, *42*(3), 253-264. https://doi.org/10.1006/jvbe.1993.1018

Truman, E., & Elliott, C. (2019). Barriers to food literacy: A conceptual model to explore factors inhibiting proficiency. *Journal of nutrition education and behavior*, *51*(1), 107-111. https://doi.org/10.1016/j.jneb.2018.08.008

Villa, F. T., Natividad, A. N., & Tulod, R. C. (2021). Research and extension completed and on-going registered programs database system (RECORDS). *Linguistics and Culture Review*, *5*(S1), 214-222. https://doi.org/10.21744/lingcure.v5nS1.1346

Zhou, M., & Brown, D. (2015). Educational learning theories.

Zulvany, V. (2020). Macrolinguistics: texts and discourses, conversation interactions and conversation components. *Macrolinguistics and Microlinguistics*, *1*(2), 104–116. Retrieved from https://mami.nyc/index.php/journal/article/view/10