ASEAN Journal for Science Education

Journal homepage: https://ejournal.bumipublikasinusantara.id/index.php/ajsed

Utilization of Visual Basic Software and its Effect on Students' Computer Programming Performances

Chuckee B. Gatchalian, Joy Edilaine R. Handig, Fioglo Baluyot*

National University Philippines, the Philippines
*Correspondence: E-mail: reyes.joyedilaine@gmail.com

ABSTRACT

The main objective of this study is to assess the utilization of visual basic software and its effect on students' performance in computer programming. This study made use of the descriptive correlational method of research that utilized a standardized questionnaire as the primary data-gathering tool. The results were analyzed and interpreted using statistical tests such as mean and regression analysis to determine the effect of the utilization of visual basic software on students' performance in computer programming. The utilization of visual basic software and computer programming performance of the students were quantified using frequency counts and weighted mean procedures. Using the aforementioned procedures, the findings of the study showed that the utilization of visual basic software was "very frequent" in terms of navigation, productivity, applicability, drill, and practices. In addition, the student's computer programming performance was "very satisfactory" as exhibited by the general weighted mean score. Lastly, it is revealed that the utilization of visual basic software significantly affects the computer programming performance of students in terms of planning, flowcharting, designing, coding, and debugging. Many implications were drawn based on the findings of the study that can be used in the utilization of visual basic software concerning computer programming performance.

ARTICLE INFO

Article History:
Submitted/Received 12 Nov 2023
First Revised 17 Dec 2023
Accepted 20 Feb 2024
First Available online 21 Feb 2024
Publication Date 01 Sep 2024

Keyword:
Computer programming performance,
Computer programming,
Visual basic.

1. INTRODUCTION

Various applications have been introduced in the education process to enhance teaching and learning activities. These applications play an important role in enriching students' learning experience on the learned subject. In programming teaching and learning, various software applications are available (Radošević *et al*., 2009). These software applications are essential since programming software and environment are closely related to and require a computer as a platform to implement and test the syntax of programming (Amoako *et al*., 2013). The programming process involves a combination of activities like planning, designing, testing, and debugging. To learn how to develop a program, students need to understand the syntax of programming language. The complexity of programming and difficulty in comprehending program logic often led to frustration and a lack of motivation to learn programming. Learning separation between theory and practical sessions is complicating the learning process in this course.

These issues contribute to the high rate of dropouts in programming courses at most universities, colleges, and also in senior high schools today. Many students find programming to be difficult and disheartening especially when they are beginners and even when they take advanced programming courses (Salleh *et al*., 2013). Today, Senior High School needs to implement the proper technique in teaching computer programming where the students can learn with a software application. This software application can facilitate the learning of the student in computer programming on their own.

Based on our previous study (Gatchalian *et al*., 2024), the objective of this study is to determine the utilization of visual basic software in computer programming performance of grade 12 TVL Computer Programming students during the School Year 2019-2020.

The visual basic software application can help and facilitate the learning of the student in computer programming on their own. Navigations, flowcharting, designing, coding, and debugging are the important elements that students can understand when they use this software application. The important things why students need to use this software application are to test the effectiveness and efficiency if they learn computer programming. Available programming software and environments such as Visual Basic intend to facilitate teaching and learning in programming and to reduce the burden on the instructor.

This study provided important information and benefits to the school, administrators, teachers, students, and researchers. The visual basic software application can teach and guide the student's attitudes toward programming and assessment activities in the process. Also, this study can help teachers to enhance the traditional ways of teaching and to keep students more engaged. The software application can teach and guide the student's attitudes toward programming and assessment activities in the process. Various software application learning strategies such as storytelling, games in learning approaches, simulation and visualization techniques as well as pair-programming approaches are implemented to enhance student engagement and to develop creative thinking as one of the preparation strategies for students to become future producers, not just consumers of technology (Hanus, 1994). This study is important to utilize the visual basic software application as a teaching tool that helps teachers and students understand computer programming. Due to all these reasons, it's really important to learn how to use programming in our daily lives.

This research is important to determine the utilization of a software application as a teaching tool that helps teachers and students understand computer programming. Due to all these reasons, it's really important to learn how to use programming in our daily lives. This study will provide important information and benefits to the following:

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

(i) School. This research will help the school find a new way of teaching computer programming to students without any references such as books or modules. It gives enhancement to the school facilities as another reference for learning about technology.
(ii) School administrators. The result of this research will help them in planning and formulating more appropriate computer software application programs that will truly address the needs of the students in the field of programming.
(iii) Teachers. The findings of this study will enable teachers to discover the effectiveness and efficiency of using software applications in teaching to improve the academic performance of the students. It encourages teachers to apply technology as a medium of instruction when they plan and prepare their lessons. It helps the teachers to enhance their creativity to design a program that can catch the interest of the student to learn programming. This research can help teachers to enhance the traditional ways of teaching and to keep students more engaged. Virtual lesson plans, grading software, and assessments can help teachers save a lot of time. This valuable time can be used for working with students who are struggling.
(iv) Students. This study will help the students achieve their potential in computer programming. Because of this study, the students are expected to be more interested in the subjects they are studying. This software application provides different opportunities to make learning more fun and enjoyable in terms of teaching the same things in new ways. Students who are engaged and interested in this study are expected to have better knowledge retention in computer programming.

The major problem of the study is to assess the effects of visual basic software utilization on students' performance in computer programming. Specifically, the study sought answers to the following questions:

(i) To what extent do teachers utilize visual basic software applications in terms of Navigation; Productivity; Applicability; and Drill and practices?
(ii) How may the students' performance in computer programming be described in terms of Planning; Flowcharting; Designing; Coding; and Debugging;
(iii) Does the extent of visual basic software utilization significantly affect students' performance in computer programming?
(iv) What implications may be derived based on the findings of the study?

The hypotheses of the Study are in the following:

(i) "The following null hypothesis was tested at 0.05 level of significance."
(ii) "The utilization of visual basic software does not affect the student's performance in computer programming."

The study was focused on the utilization of visual basic software and its effect on the performance in computer programming among grade 12 TVL computer programmers. The respondents of this study are 160 grade 12 TVL computer programmers who enrolled from the school year 2019 to 2020 and 3 senior high school teachers in grade 12 for the utilization of visual basic software applications teaching computer programming. Students' performance in computer programming was described in terms of the different techniques and processes in programming. We provided visual basic software as a teaching tool while a questionnaire was given to 3 computer programming teachers for the utilization of visual basic software.

2. LITERATURE REVIEW

2.1. Theoretical and Conceptual Framework

The study is anchored on the computer-based instruction theory (CBI), which is the discipline and model of learning programs throughout the world. Any instructional program

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

that includes the use of a computer is based on this concept, which can also be implemented in conjunction with traditional teaching methods to enhance the overall educational experience of the learners. This theory can be also used in the workplace, to educate students about new work practices or regulations that must be followed within their professional environment (Gelernter & Carriero, 1992). More complex lessons can also be delivered via computers, allowing instructors to educate their learners more effectively and profoundly. Even students who are unable to attend school or individual courses are given the chance to learn computer-based instruction.

The study postulated that the utilization of visual basic software affects the students' performance in computer programming. **Figure 1** presents the conceptual model of the study that was utilized in assessing the effects of visual basic software on the programming performance of grade 12 computer programming students. The utilization of visual basic software (IV) was determined in terms of navigation, productivity, applicability, drill, and practices. Navigation refers to the effective use of the working environment and tools of an application in creating a program. The comprehensiveness of interactive and context-sensitive online help systems is for productivity. The applicability means of compatibility of software applications in any online flat forms or devices and lastly, drills and practices for the enhancement of logical thinking and decision making.

Students' performance in computer programming (DV) was determined in terms of planning, flowcharting, designing, coding, and debugging. To learn how to draw sketches of screens that present forms and objects in their proper position in a program is called planning. Flowcharting is to understand the graphical representation of the sequence and instructions used in the program. Designing to apply suitable color combinations and appropriate themes for the programs' interface based on its purpose. While, the ability of the users to understand the codes by the use of IF statements, looping, and decision statements in creating a program is called coding. Lastly, debugging employs diagnosing the program's bugs, errors, and abnormalities.

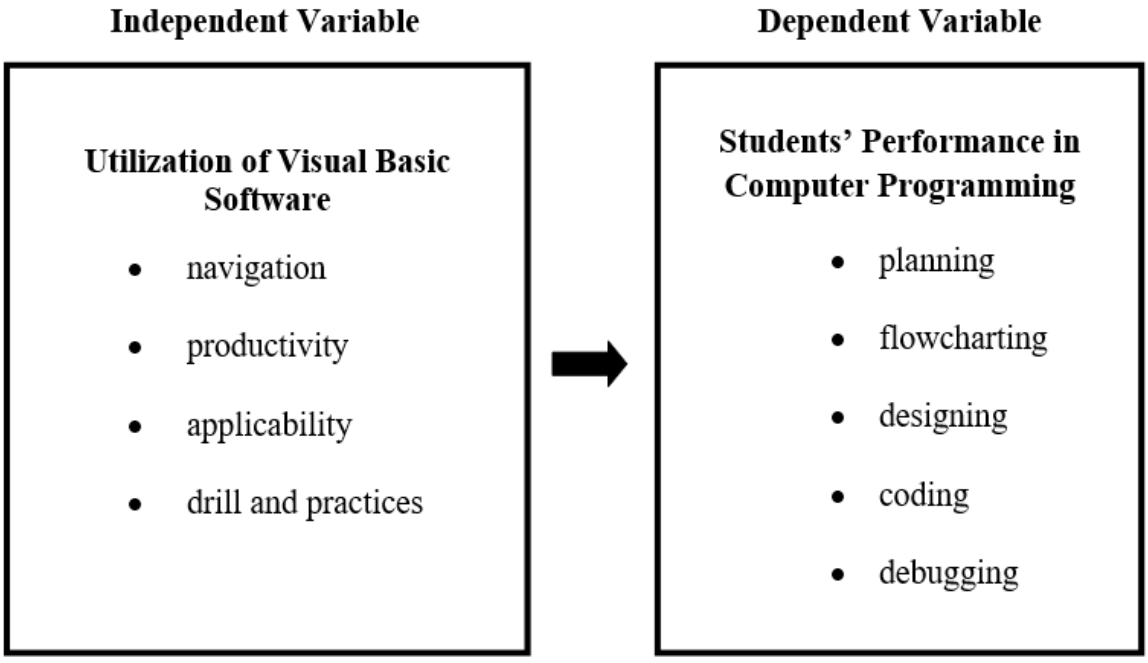

Figure 1. Conceptual model of the study.

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

2.2. Definition of Terms

The following terms were defined operationally for common understanding:

(i) **Computer Programming Performance.** This study refers to the ability and skill developed by the students to learn computer programming in terms of planning, flowcharting, designing, coding, and debugging. (Mazlack 2013) It refers to the evaluation of the teacher if the students' achievement will increase or decrease after the practice.

(ii) **Utilization of Visual Basic Software.** This refers to the practical and effective use of a software application in programming that the students and teachers are actively involved in the learning process in terms of navigation, productivity, applicability, drill, and practices using the visual basic software as a medium of instruction (Kay *et al*., 2000). Visual basic software allows learners and teachers to create software interfaces and codes in an easy-to-use graphical environment.

3. METHOD

This section presents the research methods and procedures used in the conduct of the study. It includes the respondents of the study, the data gathering, and the instruments used to gather the data. It also describes the statistical treatments applied to analyze the research problem.

3.1. Methods and Techniques Used

The descriptive-correlational method of research was utilized in the study to assess the effect of the utilization of visual basic software on the student's performance in computer programming. Descriptive correlational research is a systematic investigation of the relationship present between two or more variables.

A descriptive-correlational method is characterized as a survey or normative approach to the study of conditions, an essential guide to one's thinking. It is concerned with the condition of relationships that exist; practices that prevail, beliefs and processes that are going on; effects that are being felt, or trends that are developing.

The study used a quantitative research approach to analyze and understand the predictor and criterion variables. A standardized questionnaire on the frequency of the utilization of visual basic software was used as the primary data-gathering tool.

3.2. Respondents of the Study

The participants in this study were grade 12 TVL computer programming students during the school year 2019-2020 with a total number of 160 and 3 teachers from senior high school grade 12 of computer programming with a total of 163 participants to answer the survey questions provided.

All sections under TVL computer programming from grade 12 were utilized through universal sampling with a total of 160 students. The 3 senior high school teachers in computer programming in grade 12 were also the respondents in the study (see **Table 1**).

Table 1. Respondents of the study.

Respondents	F	%
12 - TVL Computer Programming Students	160	98.16
Teachers	3	1.84
Total	**163**	**100**

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

3.3. Instrument of the Study

This study used a standardized instrument to quantify the utilization of visual basic software in teaching computer programming. The standardized instrument is designed in such a way that the questions, scoring procedures, and interpretations are validated. We contextualized it to attain the specific objective of the study. The questionnaire consists of items that explore the utilization of visual basic software applications in the terms of following categories: navigation, productivity, applicability, drill, and practices. Also, the items that explore the students' performance in computer programming are in the terms of following categories: planning, flowcharting, designing, coding, and debugging. The instrument also used Cronbach's alpha to measure if the multiple-question Likert scale survey is reliable.

On the other hand, the student's performance in computer programming was measured using the respondents' midterm grades on their major subject; computer programming during the first semester of school year 2019-2020. Based on the University's grading system for the specialized subjects, the midterm grade for the first semester was measured using the following components: hands-on activities, quizzes, drills or practices, recitations, projects, and examinations. Academic performance can be assessed based on the outcome that captures the quality of students' academic work such as course grades or GPA. Student learning is included to capture outcomes related to specific institution or program learning outcomes, including cognitive and affective skills.

3.4. Data Gathering Procedure

The questionnaire method was the mode of data gathering. Each of the respondents was given a structured set of questions. In gathering the data, we carried out the following:

(i) We asked for approval from the adviser before the distribution of the questionnaire.
(ii) A permission letter was given to the school heads /principal of the basic education department.
(iii) Distributed the questionnaires to grade 12 computer programming teachers with the permission of school heads/principals.
(iv) The questionnaires were presented and answered individually by the respondents.
(v) We collected the questionnaires personally from the respondents and checked if all the items were answered.
(vi) When the questionnaires were retrieved already, we tallied, analyzed, and interpreted the data.

3.5. Data Processing and Statistical Treatment

The data collected were tabulated and processed using Statistical Packages for Social Sciences (SPSS) and Minitab. To analyze and interpret the data gathered, the following statistical measures were used: In the survey questionnaire for utilization of visual basic software, the indicators were interpreted by using the Likert scale with a range of 1 (never) to 5 (very frequently) (see **Table 2**).

Table 2. Rank in this test.

Rank	Value	Grade	Description
5	4.50 – 5.00	90-100	Very Frequently
4	3.50 – 4.49	85-89	Somewhat Frequently
3	2.50 – 3.49	80-84	Sometimes
2	1.50 – 2.49	75-79	Rarely
1	1.00 – 1.49	Below 75	Never

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

Table 2 shows the scales used in interpreting the utilization of visual basic software.

(i) Rank 1 with the corresponding description "never" as evidenced by the range value of 1.0-1.49 with a grade of below 75,
(ii) Rank 2 for "rarely" with a range value of 1.50-2.49 with a grade of 75-79,
(iii) Rank 3 with the corresponding description "sometimes" and 2.50-3.49 with a grade of 80-84,
(iv) Rank 4 for "somewhat frequently" shows a range value of 3.50-4.49 with a grade of 85-89 and lastly,
(v) Rank 5 with mean "very frequently" with corresponding range value of 4.50-5.00 with a grade of 90-100.

Another rank is in **Table 3** in the survey questionnaire for students' computer programming performance. the indicators were interpreted by using the Likert scale with a range of 1 (poor) to 5 (outstanding).

Table 3. Another rank.

Rank	Value	Grade	Description
5	4.50 – 5.00	90-100	Outstanding
4	3.50 – 4.49	85-89	Very Satisfactory
3	2.50 – 3.49	80-84	Satisfactory
2	1.50 – 2.49	75-79	Fair
1	1.00 – 1.49	Below 75	Poor

Table 3 shows the scales used in interpreting the student's computer programming performance.

(i) Rank 1 with the corresponding description "poor" as evidenced by the range value of 1.0-1.49 with a grade of below 75,
(ii) Rank 2 for "fair" with a range value of 1.5-2.49 with a grade of 75-79,
(iii) Rank 3 with the corresponding description "satisfactory" and 2.5-3.49 with a grade of 80-84,
(iv) Rank 4 for "very satisfactory" shows with a range value of 3.5-4.49 with a grade of 85-89
(v) Rank 5 with mean "outstanding" with corresponding range value of 4.50-5.00 with a grade of 90-100.

In addition, the frequency of the utilization of visual basic software was assessed using mean computation, and the students' computer programming performance was assessed using frequency, percentage, mean, and standard deviation; while the impact of utilization of visual basic software and students' performance in computer programming was assessed using correlation and regression analysis.

4. RESULTS AND DISCUSSION

This section presents analyses and interprets the data collected in the study. For an organized presentation and consistent discussion, the data are presented following the order or sequence of the questions raised in the introduction, to understand:

(i) the extent of the utilization of visual basic software applications,
(ii) performance in computer programming,
(iii) effect of the utilization of visual basic software on students' performance in computer programming, and
(iv) implications derived from the findings of the study.

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

4.1. The Utilization of Visual Basic Software

The utilization of visual basic software in the programming class is an important part of the teaching-learning process in computer programming subject since learning different subject matter requires a certain medium of instruction. Effective programming language use relies on the educators that help one gain productive outputs. Teachers are expected to use the target programming language in teaching a specific language; however, there are cases when they do not use the target programming language strictly in the programming class.

This paper investigated the utilization of visual basic software used by programming teachers. This extent was classified into four categories: navigation, productivity, applicability, drill, and practices.

4.1.1. Navigation.

Table 4 shows the utilization of visual basic software in terms of navigation. As can be gleaned from the analysis of data in **Table 4**, visual basic software in terms of navigation is utilized by the teacher "very frequently" as evidenced by the average of 4.76.

Table 4. Utilization of visual basic software in terms of navigation.

Indicators	Mean	Interpretation
Effective use of working environment and tools in creating a program.	4.80	Very Frequently
Following instructions to guide the user in creating a program.	5.00	Very Frequently
Efficient for input and output of an object inside the working environment.	4.60	Very Frequently
Allow users to see the errors of codes while typing.	4.60	Very Frequently
Effectively used to arrange object groups based on their purposes.	4.80	Very Frequently
Average	**4.76**	**Very Frequently**

The visual basic software is utilized "very frequently" by the teacher for the effective use of the working environment and tools in creating a program (4.80), for following instructions to guide the user in creating a program (5.00), for efficient input and output of an object inside the working environment (4.60), to allow users seeing the errors of codes while typing (4.60) and for effectively used to arrange object groups based on its purposes (4.76).

4.1.2. Productivity.

Table 5 shows the utilization of visual basic software in terms of productivity. The data suggests that visual basic software is utilized by the teacher "somewhat frequently" in conveniently as evidenced by the average score of 4.00.

Table 5. Utilization of visual basic software in terms of productivity.

Indicators	Mean	Interpretation
Availability for complete tools and codes needed in programs' creation.	4.60	Very Frequently
Good components for embedded links to the application's user interface.	4.40	Somewhat Frequently
Comprehensive for the interactive and context-sensitive online help system.	4.60	Very Frequently
Intuitively appealing views for the management of the program structure.	4.80	Very Frequently
Average	**4.00**	**Somewhat Frequently**

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

The visual Basic software is utilized by the teacher "very frequently" for the availability of complete tools and codes needed in programs' creation (4.60), "somewhat frequently" for the good components for embedded links to the application's user interface (4.40), "very frequently" for the comprehensive for interactive and context-sensitive online help system (4.60) and intuitively appealing views for the management of the program structure (4.80).

4.1.3. Applicability

Table 6 shows the utilization of visual basic software in terms of applicability. Analysis of the utilization of visual basic software in terms of applicability in **Table 6** reveals that visual basic software is utilized by the teacher "very frequently" as shown by the average of 4.65. It is utilized "very frequently" by the teacher for the component object model can be written in different languages (4.60), for easy to install and update in any version of Microsoft Windows (4.80), for flexible and useful in accessing databases (4.80), and utilized "somewhat frequently" by the teacher for the compatible software application in any online flat forms and devices (4.40).

Table 6. Utilization of visual basic software in terms of applicability.

Indicators	Mean	Interpretation
The component object model can be written in different languages.	4.60	Very Frequently
Easy to install and update in any version of Microsoft Windows.	4.80	Very Frequently
Flexible and useful in accessing databases.	4.80	Very Frequently
Compatible software application in any online flat form and devices.	4.40	Somewhat Frequently
Average	**4.65**	**Very Frequently**

4.1.4. Drill and Practices

Table 7 shows the utilization of visual basic software in terms of drills and practices. The analysis of data in **Table 7**, visual basic software in terms of drills and practices is utilized by teachers "very frequently" as evidenced by the average of 4.85. Visual Basic software is utilized "very frequently" by teachers for the enhancement of logical thinking and decision-making (5.00), for the development of typing and coding systems (4.80), to increase collaborative learning activities (4.80), and to create quality and productive software application (4.80).

Table 7. Utilization of visual basic software in terms of drill and practices.

Indicators	Mean	Interpretation
Enhancement for logical thinking and decision-making.	5.00	Very Frequently
Development of typing and coding system.	4.80	Very Frequently
Increase collaborative learning activities.	4.80	Very Frequently
Create quality and productive software applications.	4.80	Very Frequently
Average	**4.85**	**Very Frequently**

Table 8 summarizes the mean score of the utilization of visual basic software. As can be observed in the summary of the utilization of visual basic software's average mean score in **Table 8**, the utilization of visual basic software in terms of drills and practices recorded the highest average of 4.85. This suggests that teachers mostly utilized visual basic software for drills and practices to enhance logical thinking, development for coding, increase collaborative learning activities, and create quality, productive software applications. The result of the study on the utilization of visual basic software in terms of drill and practices concurs with the findings of Oroma *et al*. (2012).

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

Drill and practice exercises with appropriate programming software that can enhance the daily classroom experience. Given the personalized, interactive nature of most programming software, the computer can lend itself to providing extended, programmed practice. Like visual basic software that allows students to reinforce specific skills in a certain subject area. Although not as easily integrated across the curriculum, drill and practice visual basic software can be useful. Visual basic software provides student-centered activities to apply problem-solving strategies as well as an opportunity to practice basic skills. On the other hand, productivity, although it is utilized "somewhat frequently" based on the result, is found the least utilized visual basic software by the teacher as evidenced by the average mean score of 4.00.

This connotes that the teacher is slightly less endorsed to utilize visual basic software when using the tools, codes, embedded links, interaction, and viewing the program structure. Corman *et al*. (2015) gained the same result in terms of utilizing visual basic software in teaching computer programming. Another study suggests that visual basic software as a programming language is mostly applied to solve real-life problems. There is the algorithm and logical properties discipline. Computer programming-associated forms are frequently used by several simulation models to execute logical operations in a sequential manner which is scheduled by the program. Hence, senior high school students can now apply their analytical and logical techniques alongside computation to solve programming problems. The result suggests that these categories utilized visual basic software "very frequently".

Table 8. Mean scores of utilizations of visual basic software.

Utilization of Visual Basic Software	Average	Interpretation	Rank
Productivity	4.00	Somewhat Frequently	4th
Applicability	4.65	Very Frequently	3rd
Navigation	4.76	Very Frequently	2nd
Drill and Practices	4.85	Very Frequently	1st
General Average	**4.57**	**Very Frequently**	

4.2. Students' Performance in Computer Programming

Computer programming performance is one of the nuances that need attention. It includes the background of the students, their field of study, and learning approaches applied to the study of programming courses. It is worth considering as a major factor and necessary cause of failure of students in programming courses from the learner's perspective (Burstall & Darlington, 1977). Planning, flowcharting, designing, coding, and debugging are the five fundamentals of programming that students need to achieve in their academic performance in programming.

The responses from the three sections of grade 12 computer programming were compared to the usefulness of their background, field of study, and learning approaches toward programming courses. A three-stage model of learning: presage, process, and product. Presence factors exist before the student enters the learning situation, and include such factors as prior knowledge, intelligence quotient, and home background. The process factors describe the learning context, which includes student's perceptions. The product can be objective or subjective.

However, at the same time, there are experiences, pedagogies, and contexts that can, and do, have measurable effects on students' academic success. In addition, since the respondents of the study were senior high school students, these factors should be given consideration.

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

4.2.1. Planning

Table 9 shows the students' computer programming performance in terms of planning. The data analysis reveals that in terms of planning, the computer programming performance of students is "outstanding" as evidenced by the average score of 4.56.

Table 9. Students' computer programming performance in terms of planning.

Indicators	Mean	Interpretation
Learning to sketch screens that present forms and objects in their proper position in a program.	4.55	Outstanding
Providing a complete list and quantity of objects used in a program.	4.69	Outstanding
Changing the name of objects that indicates their properties in the program.	4.34	Very Satisfactory
Writing the codes required for each object used to build the program.	4.64	Outstanding
Average	**4.56**	**Outstanding**

Computer programming performance of students is "outstanding" for learning to sketch screens that present forms and objects on their proper position in a program (4.55), providing a complete list and quantity of objects used in a program (4.69), "very satisfactory" for changing the name of objects that indicates their properties in the program (4.34), and "outstanding" for writing the codes required for each object used to build the program (4.64).

4.2.2. Flowcharting

Table 10 shows the students' computer programming performance in terms of flowcharting. The data presents that students' computer programming performance is "very satisfactory" in flowcharting as evidenced by the average score of 4.49.

Table 10. Students' computer programming performance in terms of flowcharting.

Indicators	Mean	Interpretation
Understanding the graphical representation of sequence and instructions used in the program.	4.49	Very Satisfactory
Connecting the graphical representation to the objects' process.	4.42	Very Satisfactory
Using complete and correct shapes that symbolized the objects used in the program.	4.68	Outstanding
Illustrating a diagrammatic representation of a solution model to a given problem.	4.35	Very Satisfactory
Average	**4.49**	**Very Satisfactory**

The computer programming performance of students was "very satisfactory" for understanding the graphical representation of sequence and instructions used in the program (4.49), for connecting the graphical representation to the objects' process (4.42), "outstanding" for using complete and correct shapes that symbolized the objects used in the program (4.68), and lastly, "very satisfactory" for illustrating a diagrammatic representation of solution model to a given problem (4.35).

4.2.3. Designing

Table 11 shows the students' computer programming performance in terms of design. Based on the analysis of data in **Table 11**, students' computer programming performance is "outstanding" in designing with an average score of 4.65. Students' computer programming performance is "outstanding" for creating an appropriate theme for the programs' interface

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

based on its purpose (4.68), following the plan sketched in creating a program (4.67), for applying suitable color combination for the objects' interface (4.66), and for maintaining the simplicity and productivity of the programs created (4.61).

Table 11. Students' computer programming performance in terms of designing.

Indicators	Mean	Interpretation
Creating an appropriate theme for the programs' interface based on its purpose.	4.68	Outstanding
Following the plan sketched in creating a program.	4.67	Outstanding
Applying suitable color combinations for the objects' interface.	4.66	Outstanding
Maintaining the simplicity and productivity of the programs created.	4.61	Outstanding
Average	**4.65**	**Outstanding**

4.2.4. Coding

Table 12 shows the students' programming performance in terms of coding. Analysis of data suggests that students' computer programming performance is "outstanding" in coding as evidenced by the average score of 4.54. Students' computer programming performance is "outstanding" for understanding the codes for each object used in the program (4.58), "very satisfactory" for using IF statements, looping, and decision statements in creating a program (4.43), "outstanding" for applying the proper typing techniques of codes in the project form (4.62), and organizing the sequence of codes based on the flowchart used (4.56).

Table 12. Students' computer programming performance in terms of coding.

Indicators	Mean	Interpretation
Understanding the codes for each object used in the program.	4.58	Outstanding
Using IF statements, looping, and decision statements in creating a program.	4.43	Very Satisfactory
Applying the proper typing techniques of codes in the project form.	4.62	Outstanding
Organizing the sequence of codes based on the flowchart used.	4.56	Outstanding
Average	**4.54**	**Outstanding**

4.2.5. Debugging

Table 13 shows the students' programming performance in terms of debugging. Analysis of data for students' computer programming performance in terms of debugging is "outstanding" as shown by the average of 4.59. "Outstanding" as interpretation for troubleshooting programs bugs, errors, and abnormalities (4.51), "very satisfactory" for replacing undefined codes according to the set of specifications (4.47), "outstanding" for providing complete software programs that are regularly compiled and executed to identity (4.64), and lastly, for saving the program to become an application without abnormalities and errors (4.73).

Table 14 summarizes the mean score of students' computer programming performance. As can be observed in the summary of computer programming performance's average mean scores in **Table 14**, students' computer programming in terms of designing recorded the highest average of 4.65. This suggests that students mostly applied designing in creating a program by utilizing visual basic software, such as creating an appropriate program's theme, sketching, applying color combinations, and maintaining the simplicity and productivity of a program.

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

Table 13. Students' computer programming performance in terms of debugging.

Indicators	Mean	Interpretation
Troubleshooting program bugs, errors, and abnormalities.	4.51	Outstanding
Replacing undefined codes according to the set of specifications.	4.47	Very Satisfactory
Providing complete software programs that are regularly compiled and executed to identity.	4.64	Outstanding
Saving the program to become an application without abnormalities and errors.	4.73	Outstanding
Average	**4.59**	**Outstanding**

Table 14. Mean scores of students' computer programming performance.

Utilization of Visual Basic Software	Average	Interpretation	Rank
Flow Charting	4.49	Very Satisfactory	5th
Coding	4.54	Outstanding	4th
Planning	4.56	Outstanding	3rd
Debugging	4.59	Outstanding	2nd
Designing	4.65	Outstanding	1st
General Average	**4.57**	**Outstanding**	

The result of the study on students' computer programming performance in terms of design concurs with the findings of Oroma *et al*. (2012). Students more focus on designing rather than coding and flowcharting, the reasons are; to attract the users to navigate the program, and maintain the organization and clarity of the program's interface. On the other hand, flowcharting, although it is utilized as "very satisfactory" based on the result, is found the lowest computer programming performance by students utilizing visual basic software as evidenced by the average mean score of 4.49. This connotes that the students are slightly less endorsed to applied flowcharting when illustrating a grammatical representation of sequence and instructions used in creating a program.

Their study suggests that flowcharts would be difficult to read and understand if they were always drawn differently and were inconsistent. Flowcharting can be an effective, efficient way to show steps in a process. But they can also be confusing and even misleading if not properly prepared, that's why students slightly less applied flowcharting in utilizing visual basic software. The results suggest that these categories' students' computer programming performance is "outstanding".

4.3. Utilization of Visual Basic Software and its Effect on Students' Computer Programming Performances

In the conduct of the study, the null hypothesis states that the frequency of the utilization of visual basic software does not significantly affect the students' computer programming performances. The data collected were subjected to regression analysis to determine the extent of impact the predictor variables cause on the criterion variable. **Table 15** shows the regression analysis of the utilization of visual basic software and its effect on students' computer programming performances. As can be observed from the results in **Table 15**, the obtained Beta coefficients of 1.349 (navigation), 0.334 (productivity), 0.37 (applicability), and 1.349 (drill and practices) suggest that the four factors contribute significant effect on the student's computer programming performances.

The B coefficient results indicate that in every unit increase in the extent of utilization of visual basic software in terms of navigation, productivity, applicability, drill, and practices will increase the computer programming performance of the respondents. The positive B

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

coefficient values in navigation, productivity, applicability, drill, and practices suggest that in every unit increase of teachers' utilization of visual basic software in the following terms, there is a corresponding increase in students' computer programming performance.

Table 15. Regression analysis of utilization of visual basic software and its effect on students' computer programming performances.

Variables	Unstandardized Coefficients		Standardized Coefficients		
	B	Std. Error	Beta	t	Sig.
(Constant)	4.32	0.671		6.438	0.023
Navigation	0.624	0.213	1.349	2.937	0.099
Productivity	0.15	0.147	0.334	1.021	0.415
Applicability	0.105	0.097	0.37	1.083	0.392
Drill and practices	0.591	0.222	1.349	2.663	0.117
R-squared = 0.859					
F-value = 3.047					
p-value = 0.262					
alpha = 0.05					

Further analysis of **Table 15** reveals an F-value of 3.047 with the associated p-value of 0.262. Since the associated probability exceeds 0.05 alpha, it is, therefore, safe to conclude that the combined effect of the utilization of visual basic software in navigation, productivity, applicability, drill, and practices did not form a set of significant predictors on the academic performance. Hence, the decision is to *accept the null hypothesis which states that the utilization of visual basic software does not affect the student's computer programming performance.* The results of the study corroborated the findings of Esteves *et al*. (2011) and Daiva (2011) with stated that visual basic software is not only a programming language but primarily a integrated, interactive development environment in creating a program. It has been highly optimized to develop users' logical and critical thinking skills. It is particularly easy to develop productive and effective programs and to connect them to handler navigations provided by the application. This includes visual basic software for effective drills and practices on the part of both teachers and learners. White (2012) introduces "visual basic.net" Like visual basic software, VB.net includes the concepts of "encapsulation" and "abstraction". Abstraction means to show essential features to the outside world, so that learners can understand what a programming element does, without having to necessarily understand exactly what goes on inside it. Encapsulation means storing both data and methods in one unit. This implies that visual basic software should be utilized in computer programming classes for effective practices to develop the understanding and skills of students in the field of programming.

4.4. Implication Drawn from the Findings of the Study

The following were the implications drawn based on the findings of the study:

(i) Although teachers in programming utilized visual basic software as a fundamental language in the field of programming, the actual demonstrations, discussion, and hands-on allowed and compelled teachers to utilize visual basic software for the learners understanding. Although learners could understand computer programming, visual basic software at times should be utilized as a means to complement an understanding of creating and designing a specific program.

(ii) Teachers and learners of programming subjects at senior high schools where visual basic software is used as one of the effective and productive programming languages and also

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

develop learners' discipline in flowcharting, coding, planning debugging, and designing a program.

(iii) Students might need appropriate critical and logical thinking to be actively involved in the class. Thus, the teacher's use of appropriate knowledge, methods, and techniques seems to be significant where active involvement and participation are crucial. This must be appropriately supported through proper demonstration, application, suitable activities, and tasks.

(iv) The effect of the utilization of visual basic software on student's computer programming performance underscores the attention of students and teachers to the endorsed outlets of learning in programming.

4.5. Several findings

4.5.1. Problem 1: The frequency of utilization of visual basic software

The frequency of utilization of visual basic software was "outstanding" as evidenced by the general weighted mean value of 4.57. The utilization of visual basic software was "outstanding" in terms of applicability, navigation, drill, and practices as evidenced by the weighted mean values of 4.65, 4.76, and 4.85, and "very satisfactory" in terms of productivity as evidenced by the weighted mean value of 4.00 respectively.

4.5.2. Problem 2: The respondents' computer programming performance

The students' level of computer programming performance was "outstanding" as exhibited by the general weighted mean score of 4.57 also. The student's computer programming performance was "outstanding" in terms of coding, planning, debugging, and designing as evidenced by the weighted mean values of 4.54, 4.56, 4.59, 4.65, and "very satisfactory" in terms of flowcharting as evidenced by the weighted mean value of 4.49 respectively.

4.5.3. Problem 3: Effect of utilization of visual basic software on the students' computer programming performance

Results of regression analysis revealed that the obtained Beta coefficients of 1.349 (navigation), 0.334 (productivity), 0.37 (applicability), and 1.349 (drill and practices) suggest that the four factors contribute significantly effect on the student's computer programming performances as evidenced by p-values which do not exceed the .05 alpha. The B coefficient results indicate that in every unit increase in the extent of utilization of visual basic software in terms of navigation, productivity, applicability, drill, and practices will increase the computer programming performance of the respondents. The positive B coefficient values in navigation, productivity, applicability, drill, and practices suggest that in every unit increase of teachers' utilization of visual basic software in the following terms, there is a corresponding increase in students' computer programming performance.

Further analysis reveals an F-value of 3.047 with the associated p-value of 0.262. Since the associated probability does not exceed .05 alpha, it is, therefore, safe to conclude that the combined effect of the utilization of visual basic software in navigation, productivity, applicability, drill, and practices form a set of significant predictors of academic performance. This gives enough reason for the study to reject the null hypothesis of no significance that the utilization of visual basic software does not significantly on the computer programming performances of students.

4.5.4. Problem 4: Implications are drawn from the findings of the study

The following implications are drawn from the findings of the study:

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

(i) Although teachers in programming utilized visual basic software as a fundamental language in the field of programming, the actual demonstrations, discussion, and hands-on allowed and compelled teachers to utilize visual basic software for the learners understanding. Although learners could understand computer programming, visual basic software at times should be utilized as a means to complement an understanding of creating and designing a specific program.
(ii) Teachers and learners of programming subjects at senior high school where visual basic software is used as one of the effective and productive programming languages and also develop learners' discipline in flowcharting, coding, planning debugging, and designing a program.
(iii) Students might need appropriate critical and logical thinking to be actively involved in the class. Thus, the teacher's use of appropriate knowledge, methods, and techniques seems to be significant where active involvement and participation are crucial. This must be appropriately supported by utilizing proper demonstration, application, suitable activities, and tasks.
(iv) The effect of the utilization of visual basic software on student's computer programming performance underscores the attention of students and teachers to the endorsed outlets of learning in programming. Training, seminars, and programs are essential avenues to address the demands of education in the present time.

5. CONCLUSION

In the light of the findings of the study, the following conclusions were drawn:

(i) Although teachers in programming utilized visual basic software as a fundamental language in the field of programming, as indicated in the senior high school, visual basic software was utilized very frequently by the teacher for the following purposes: navigation, productivity, applicability, drill, and practices.
(ii) Students' computer programming performance met their satisfactory competency requirements. It is also indicated that the student's computer programming performances are influenced effectively by utilizing visual basic software, wherein the frequency of utilization of visual basic software and students' level of computer programming performance was outstanding.
(iii) The utilization of visual basic software in terms of navigation, productivity, applicability, drill, and practices affects the computer programming performance of the students.
(iv) The null hypothesis that the utilization of visual basic software does not affect significantly students' computer programming performance has been rejected.
(v) The findings drew several implications that may help learners and teachers realize the need for a comprehensive awareness of visual basic software and computer programming performance.

Based on the findings and conclusion of the study, the following recommendations are hereby offered:

(i) That the learners be focused and understand the different methods, techniques, and practices in the field of programming to excellently meet learning objectives, especially for planning, flowcharting, designing, coding, and debugging.
(ii) The teachers maximize the learners' ability and skills to use visual basic software and other programming languages that may suit different methods in computer programming, especially for navigation, productivity, applicability, drill, and practices.

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295

(iii) The teachers should implement collaborative activities in programming, use of appropriate knowledge, methods, demonstrations, applications, suitable activities, and tasks for the student's assessment.
(iv) School administrators support and conduct seminars and training for the teachers and students to develop their skills and abilities in computer programming.

6. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

7. REFERENCES

Amoako, P. Y. O., Sarpong, K. A. M., Arthur, J. K., and Adjetey, C. (2013). Performance of students in computer programming: Background, field of study and learning approach paradigm. *International Journal of Computer Applications*, *77*(12), 17-21.

Burstall, R. M., and Darlington, J. (1977). A transformation system for developing recursive programs. *Journal of the ACM (JACM)*, *24*(1), 44-67.

Corman, L. and Guynes, C. and Vanecek, M. (2015). A study of laterality and cognitive style in information systems and liberal arts students. *Journal of Computer Information Systems*, *35*(2), 26-34.

Daiva. (2011) Problems in choosing tools and methods for teaching programming. *Informatics in Education*, *11*(2), 271–282.

Esteves, M., Fonseca, B., Morgado, L., and Martins, P. (2011). Improving teaching and learning of computer programming through the use of the second Life virtual world. *British Journal of Educational Technology*, *42*(4), 624-637.

Gatchalian, C.B., Handig, J.E.R., and Baluyot, F. (2024). Review on utilization of visual basic software for improving students' computer programming performances. *ASEAN Journal of Educational Research and Technology*, *3*(3), 185-196.

Gelernter, D., and Carriero, N. (1992). Coordination languages and their significance. *Communications of the ACM*, *35*(2), 96.

Hanus, M. (1994). The integration of functions into logic programming: From theory to practice. *The Journal of Logic Programming*, *19*, 583-628.

Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., and Crawford, K. (2000). Problem-based learning for foundation computer science courses. *Computer Science Education*, *10*(2), 109-128.

Oroma, J.O., Wanga, H., and Ngumbuke, F. (2012). Challenges of teaching and learning computer programming in developing countries: Lessons from Tumaini University. *The 6th International Technology, Education and Development Conference, Valencia*, *37*(4), 107–110.

Radošević, D., Orehovački, T., and Lovrenčić, A. (2009). Verificator: educational tool for learning programming. *Informatics in Education*, *8*(2), 261-280.

Salleh, S. M., Shukur, Z., and Judi, H. M. (2013). Analysis of research in programming teaching tools: An initial review. *Procedia-Social and Behavioral Sciences*, *103*, 127-135.

White, G. (2012). Visual basic programming impact on cognitive style of college students: need for prerequisites. *Information Systems Education Journal*, *10*(4), 74.

DOI:
p- ISSN 2828-3287 e- ISSN 2828-3295