INNOVARE JOURNAL OF EDUCATION
Vol 12, Issue 4, 2024, 10-15
ISSN: 2347-5528
Research Article

The Effect of Improvised Computer-Based Software Package on Secondary School Students' Achievement and Attitude Towards Mathematics

Etukudo Udobia Elijah
Department of Mathematics Education,
Federal College of Education (Technical),
Omoku, Rivers State, Nigeria

Egba Anwaitu Freser
Department of Computer and Robotics,
Federal College of Education (Technical),
Omoku, Rivers State, Nigeria

Abstract

The study was conducted to determine the effect of improvised computer-based software on the achievement and attitude of senior secondary school students towards mathematics. Fifteen secondary school mathematics teachers were trained to produce a computer-based software package and use the package to teach 'graphical solution to quadratic equations' to senior secondary class one (SSCI) students. A total sample of 1487 SSCI students was used in the study: 745 students in the 15 secondary schools purposively selected on the rationale of their closeness to a computer center, used as the experimental group, and 742 students from the 15 schools randomly selected for use as the control group. Pre-test, post-test, and quasi-experimental research designs were used to conduct the study. Fifteen intact classes were simultaneously used for treatment in the experimental and control groups. The experimental groups were taught with computers using the computer-based software package on teaching graphical solutions to quadratics equations produced by their teacher, while the control groups were taught using the conventional strategy. The result reveals a significant difference in academic achievement between students taught with improvised computer-based software packages and those taught using the conventional strategy. The mean achievement score of 73% and standard deviation of 9% was obtained from the experimental group, while the control group had a mean achievement of 57% and standard deviation of 5%.
Similarly, the experimental group had an attitude test score of 82% and a standard deviation of 6%, while the control group had a mean attitude test score of 59% and a standard deviation of 11%. This gave the calculated value of the *t*-test as $t = 42.3$ for achievement and $t = 50.1$ for attitude, showing a significant difference in achievement and attitude between the experimental and control groups. Thus, an improvised computer-based software package is more effective in teaching mathematics than the conventional strategy. Hence, improvising computer-based software for teaching and learning is highly recommended.

Keywords: Mathematics, improvisation, computer-based software package, achievement, attitude

Introduction

It has become obvious that using computers in teaching and learning has contributed immensely to alleviating the problem of understanding and mastery. Computers have played a huge role in making learning easy. Teachers who have been able to lay their hand on adequate and suitable software for teaching and learning always find it easy to communicate the instructional content to learners so they can easily master it. The joy of a good teacher is in having desirable results. This is made possible with the use of the computer. Rouse (2022) emphasized that computer-based learning has many benefits, such as helping users to learn at their own pace, being very interactive, learning at one desired time, having that which is globally accepted, and having the ability to accommodate traditional methods. Rouse (2022) stated that knowledge-based training and assessments, simulation, and drills are done prudently using computers. This makes using computers relevant to teaching and learning (Ingram, 1985). The computer software package was developed and used in teaching the English Language, and it was discovered that the product was easy to duplicate, contained elaborate learning material, pictures, and displace design, which made the students more enthusiastic and interested, while the exercises were more fascinating (Rohmah, 2019). Using computers in teaching has helped to remove the complexity in instructional delivery (Garba et al., 2010; van Merriënboer & Martens, 2002).

The most severe obstacles to implementing computer-assisted instruction in school systems include the non-availability of facilities and the lack of affordable and relevant software packages. The relevant packages to use and carry out instructional delivery are scarce. Several models have been presented to be used to develop computer-based instructional packages. Rosenthal (1976) presented a model containing segments and phases of instruction, which included administration and coordination, development of a project team, definition of project scope and objectives, development of a program for system design, and conducting orientation meetings. Evaluating this model before implementing computer-based instruction makes it look herculean. Chen and Shen (1989) noted that waterfall models are the key to making software adaptable to

DOI: https://dx.doi.org/10.22159/ijoe.2024v12i4.51685. Journal homepage: https://journals.innovareacademics.in/index.php/ijoe.

Acknowledgement: We acknowledge all the teachers, technical staff and students that participated in the study. **Authors' Contribution:** Both authors worked together to make the study a success. **Conflict of Interest**: There is no conflict of interest. **Funding Source:** The authors wish to thank the management and the STEP-B project team of the Federal College of Education (Technical), Omoku, Rivers State Nigeria for providing the funding for the research project.

Correspondence concerning this article should be addressed to Etukudo Udobia Elijah, Department of Mathematics Education, Federal College of Education (Technical), Omoku, Rivers State, Nigeria. **Email:** Udobiaetukudo@gmail.com

users and satisfying their needs. Similarly, Pappas (2014) instructed that the production of computer-based software packages has been the bends in computer-based instructional delivery. Ruliah et al. (2019) enunciated the advantages of computer-assisted instruction. They proposed using a conceptual model comprising the introduction, presentation of information, question and responses, feedback, question and responses, feedback and responses, judging and responses, remedial, case exercise, and closing.

The software has been developed with various languages for teaching and learning. However, the process seems complex and clumsy, making it difficult to spread and effectively use; hence, some packages have a short life. Software developed by Benitex (2016), Garba et al. (2010), and Michael and Igenewari (2022) has exciting results but was produced with peculiarity for specific instruction and cannot be generally applied or used in wider perspectives.

These lapses make it obvious that there should be a need to produce software that can be applied in all areas when the need arises. It has become pertinent that teachers should know how to produce software for their instructional usage. It cannot be anticipated that instructional software can be sourced from the market for every topic to be taught.

This study is carried out to train mathematics teachers in producing instruction software using a reality model. The reality model provides direction that the teachers can always follow to produce software as much as they choose to and for every topic they need to use. In addition, secondary school teachers were taught to improve the computer software used to teach graphical solutions to simultaneous equations and compare them with the conventional strategy.

Statement of the Problem

Considering that computer-based software has a huge positive impact on teaching and learning as well as general instructional delivery when applied, it becomes necessary that the product be available for teachers and learners at any time there is a need to apply. This cannot be the case, except that the teachers can produce the software for their use in teaching and learning mathematics. Teaching mathematics for learners to achieve maximally and induce an acceptable attitude towards the subject has always been the bend in education. Many strategies have been devised to help stimulate learners' attitudes towards the subject and boost their achievement, among which are ability grouping, question and answer method, discovery methods, and expository method, to mention a few. However, the outcome has remained the same. The reason is the inability to properly harness the strategies for effective teaching and learning of the subject. Computer-assisted instruction has always produced good results, but the clog in the wheel of utilizing the strategy for proper teaching and learning mathematics has been the availability of the desired software package. The improvisation of a computer-based software package for teaching mathematics by mathematics teachers has generated a positive attitude in learners. It can improve their achievement in the subject. Positive attitude and good achievement are induced using an improvised computer-based software package the subject teacher produces to teach mathematics.

Significance of the Study

The major problem that hampered the proper implementation of computer-based instructional delivery in mathematics is the availability of relevant software at an affordable cost. Though the use of computers in teaching mathematics at all levels has been accepted as a necessary breakthrough in the fields of educational development, it has not been possible to be widely applied because the mathematics teacher lacks the skills to use the packages and also relevant packages are not available in all the topics, therefore the study trained the mathematics teachers to produce the software packages themselves. Their product, the computer-based package, is used in teaching graphical solutions to quadratic equations in this study. This gives the mathematics teachers the confidence that the skills and adepts they acquire for producing computer-based software are relevant to their vocation and can assist them in producing better outcomes. The study will make the improvisation of computer-based instruction packages germane to the teaching profession readily available for teaching mathematics, improving students' achievement and general learning outcomes in mathematics.

Purpose of the Study

Several researchers have enunciated the relevance and essence of computers in teaching and learning, but getting the relevant packages for implementing computer-assisted instruction when needed has always been the clog in the wheel of progress of using computer-based packages in teaching and learning. This paper aims to alleviate the problems associated with the production of instructional software for teaching and learning mathematics by training mathematics teachers in producing and using software for teaching and learning mathematics. It enunciated the effect of an improvised computer-based software package on the achievement and attitude of students towards mathematics. In other words, it promotes software improvisation. The teachers are trained to write programs that can be utilized in mathematics classrooms. The achievement and attitude of the students taught with the programs in comparison with the regular strategy are reported in this paper.

Hypotheses

1. There is no significant difference in the mean achievement score of students taught mathematics using improvised computer-based software packages and those taught using conventional strategy.
2. There is no significant difference in the mean attitude test scores of students taught mathematics with improvised computer-based software packages and those taught using conventional strategy.

Methodology

Design

The study is a quasi-experiment that was carried out with thirty (30) senior secondary school mathematics teachers. Fifteen (15) were in experimental and control groups, respectively.

Population of Study

The study population includes all mathematics teachers in public and private secondary schools in Rivers State of Nigeria. A total of 2155 mathematics teachers made up the population.

Sample and Sampling Technique

A total of one thousand four hundred and eighty-seven (1487) senior secondary class one (SSC1) students were used for the study. They were seven hundred and forty-five (745) for the experimental group and seven hundred and forty-two (742) for the control group. A purposive random sampling procedure was used to select the participants in the experimental groups due to proximity to standard and well-equipped computer laboratories, while random sampling was used to select the schools for the control group.

Instrumentation

The instruments used in the study include

1. Training manual
2. Program packages
3. Participant involvement checklist
4. Training checklist
5. Participant's trial test items
6. Participant assessment checklist
7. Mathematics achievement test
8. Mathematics attitude inventory test

Training Manual

This is a set of rules for the training. It states categorically the time each aspect of the training takes place. The introduction takes about ten (10) minutes. Familiarization with the computer and mastery of the keyboards twenty (20) minutes. Introduction to programming thirty (30) minutes. Interaction with the programmers and the technicians is thirty (30) minutes, and practical exercise is thirty (30) minutes. This formed the two-hour training for the first day.

The second day of the training was an introduction to BASIC programming. The participants were taught how to write BASIC programming language. The question and answer session was on the different technicalities of writing BASIC programming and producing software with it. The practical session was also included for forty-five minutes. The session ends with the correction of trainees' programs and assignments.

The third day continued with BASIC programming. The theoretical session was one hour, while the practical session was one hour, which ended with correcting trainees' programs and assignments.

Day four was the rehearsal of BASIC programming and package production with BASIC.

The fifth day, day five, was an introduction to Python programming. It was one hour of theoretical and practical work was for one hour. The session ended with an assignment on Python programming language.

The sixth day, day six, was practical programming with Python, which took one hour and forty-five minutes, except for the fifteen minutes of introduction, which ended with assignments.

The seventh day, day seven, was animation programming. One hour was the introduction and theoretical work on animation, and another hour was the practical programming and production of two-dimensional and three-dimensional objects.

The eight days were spent on mixed program package writing. Programming packages with BASIC and Python, while the ninth day was the introduction to mathematics program packages production and sample packages.

The tenth day was practical package production using topics from the scheme of work.

The eleventh day was trial package validation, while the twelfth day was implementation of the trial package. The packages were short instructions produced by the teachers.

The thirteen and fourteen days were for revision.

The third week was the practical implementation of the packages, marking pre-and post-tests of students taught with the improvised packages, collecting results on the participant's achievement, and closing the training workshop.

Program Packages

The training instructions were packaged in BASIC and Python. They were delivered through the local area network (LAN) to the different computers the trainees were using, with screen and multimedia facilities used for explanations, illustrations, and interactions. The packages were validated correctly. Other programming packages in mathematics instructions were also made available and used as examples for discussion.

Trainees' Involvement and Commitment checklist

The checklist contained the rating of the activities of the trainees during the lesson. It was presented in the table below for each of the participants. Technicians and research assistants gave the rating.

Training Checklist

This general checklist was used to monitor the rate at which the training was progressing and serving each aspect that was required.

Table 1
Participants Checklist

Items	1-10	11-20	21-30	31-40	41-50
Punctuality					
Attentiveness					
Contributions					
Asking questions					
Answered questions					
Ability to use a computer					
Ability to write codes					
Ability to interpret codes					
Logical production of ideas					
Quality program package on a chosen topic					

Note. The above was the individual checklist for each of the participants during learning.

Table 2
Training Checklist

Items	1	2	3	4	5	6	7	8	9	10
Proper introduction										
Proper use of package										
Relevance to trainees need										
Carrying trainees along										
Helping the trainees to master concepts										
Enabling the trainees to meet objectives										
Trainees' ability to produce packages										
Trainees' package meeting the objectives										
Adequacy of trainees' packages										
Learners' learning outcome improvement with trained package										

Participant Trial Test Items

This comprised the pre-test and post-test items the trainees used to implement their packages. They were set on how their packages were produced and administered before and after their experiments.

Participant Achievement Checklist

This is where items were scored and evaluated by the participants.

Table 3
Participant Achievement Checklist

Items	1	2	3	4	5	6	7	8	9	10
Involvement in the training										
Ability to write the program										
Production of a workable package										
Relevant of the package to the object to the topics										
Acceptance of the package by learners										
Achievement of objectives through the package										
Proper usage of the package										
Evaluation of the objective via the package										
General assessment of the trainee product										
Learners assessment of trainees' product										

Mathematics Achievement Test

The Mathematics achievement test comprised fifty (50) multiple choice items given to the subjects in both the control and experimental groups as pre-test and post-test. The test had a parallel form reliability coefficient of .87. The items selected were those that satisfied the discriminatory index of 0.45 to 0.55. The items were distributed on the basis of 20 on cognitive, 15 on psychomotor, and 15 on affective domains. In totality, ten items were on memory, 10 on comprehension, 10 on analysis, ten on application, 5 on synthesis and 5 on evaluation. The subjects had to draw graphs to answer some of the items to achieve the affective and psychomotor domains. The items were selected from 80 items used for the pilot study.

Mathematics Attitude Inventory Test

This comprised a 20-item Likert scale test, which was structured to determine the effect of strategy on attitude towards mathematics. It had a split-half reliability coefficient of .78. It was given to the subjects at the end of the treatment.

Research Procedure

The research was carried out by purposively sampling fifteen secondary school teachers who were used to teach the experimental group and participate in a training program for fifty mathematics teachers for computer-based instructional package production. The training was carried out in the Federal College of Education (Technical) computer laboratory with the help of two programmers, four technicians, and five other research assistants. The trainees were taught how to produce computer-based instructional packages in mathematics.

After the teaching, they produced their packages and tested them. Their packages were in different topics in mathematics taught at the level which they were teaching. They used the packages to evaluate their learner's learning outcomes to determine the success of the training. They were evaluated using the checklist. The data were collected using the checklists.

After the general training, the fifteen secondary school teachers produced computer-based instructional packages on graphical solutions to quadratic equations and used them for this study.

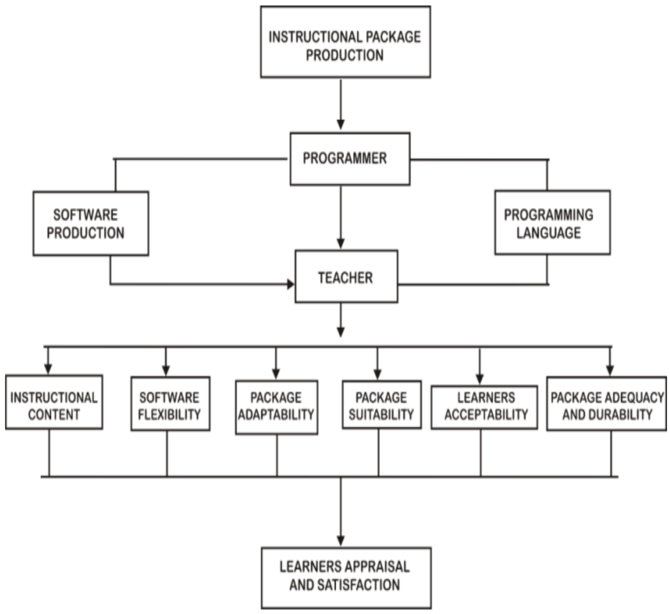
Training Model

The efficiency skewed four prompt models encompass four key agents that drive the activities: Instruction, programmer, teachers, and learners, which were used for the study.

The teacher and the programmer first examined the instructional contents, and then the nature of the software package was decided by flowcharting the components and processes. It was from here that the programming language or languages were chosen. After that, the presented instructional content and subject matter were keyed into the package. Flexibility, adaptability, suitability (relevance), acceptability (meeting the learners' needs), adequacy, and durability were considered very important elements of the functional ability of the packages. The packages were exposed to learners' appraisal in order to seek their satisfaction in trial with a smaller group, a smaller group of the contemporaries for which the package is produced. It is at this point the package can be considered efficient. This gives rise to the "efficiency skewed four prompt model" for producing computer-based packages for teaching and learning mathematics.

The training was based on the "efficiency skewed four prompt model" termed REALITY MODEL given below.

Figure 1
Efficiency Skewed Four Prompt Model



Data Analysis Procedure

The hypotheses were tested using a *t*-test to determine the difference between the experimental and control groups in achievement and attitude. All the hypotheses were tested at .05 level of significance.

Results

Hypothesis 1: There is no significant difference in the mean achievement score of students taught mathematics using improvised computer-based software packages and those taught using conventional strategy. The data analysis for testing hypothesis 1 is given in Table 4 below.

Table 4
Mean, Standard Deviation, and t-test of Achievement of Experimental and Control Groups

Groups	*n*	*M*	*SD*	*t*-test	Decision
Experimental	745	73	9	42.3	Reject
Control	742	57	5		

Note. $p = .05$, $t_{critical = 1.96.}$
$N = 1487$.

The result revealed a significant difference between experimental and control group student achievement. The experimental group, with a mean achievement score ($M = 73$, $SD = 9$), as against the control group with a mean achievement score ($M = 57$, $SD = 5$), had higher achievement, which gave rise to *t*-test of 42.3 which is higher than the critical value of 1.96. As a result, hypothesis one is rejected. There is a significant difference in the achievement of the experimental and control groups.

Hypothesis 2**:** There is no significant difference in the mean attitude test scores of students taught with improvised computer-based software packages and those taught using conventional strategy. The data analysis for testing hypothesis 2 is given in Table 5 below.

Table 5
Mean, Standard Deviation, and t-test of Attitude Scores of Experimental and Control Groups

Groups	*n*	*M*	*SD*	*t*-test	Decision
Experimental	745	82	6	50.1	Reject
Control	742	59	11		

Note. $p = .05$, $t_{critical = 1.96.}$
$N = 1487$.

The result revealed a significant difference between the attitude scores of students in experimental and control groups. The experimental groups with a mean attitude score of ($M = 82$, $SD = 6$), against the control groups with mean attitude scores ($M = 59$, $SD = 11$), had higher attitude scores, which gave rise to *t*-test of 50.1 which is higher than the critical value of 1.96. As a result, hypothesis two is rejected. There is a significant difference in the attitude scores of the experimental and control groups.

Discussion

The main purpose of this research is to reveal the possibility of making computer-based packages for the teaching and learning mathematics available everywhere and when needed, at the lowest cost or as cheap as possible, since computer-based software is seen to be adequate, essential, and efficient in promoting quality delivery of instruction and learning with ease (Michael & Igenewaru, 2022; Rohmah, 2019; Rouse, 2022). The paper reports a study in which the researcher trained mathematics teachers to produce computer-based software packages on the topics in the curriculum or scheme of work for the level they are teaching. Fifteen (15) secondary school mathematics teachers who were trained alongside thirty-five (35) other mathematics teachers from primary and tertiary levels of education were used for the study. They produced or improvised computer-based instruction software packages on graphical solutions to quadratic equations and implemented them. The result shows that there were significant differences in the achievement and attitude towards mathematics of students taught with software improvised by the teachers and those taught with the conventional strategy. The students who were taught with an improvised computer-based instructional software package had higher mean achievement and attitude test scores.

As a result, teacher-produced computer-based software packages should be encouraged by training the teachers to write programs for their use and that of their students and pupils so that they can learn effectively with the packages.

Summary

The research was carried out to determine the effect of improved computer-based instructional software packages on achievement and attitude of students towards mathematics, using 1487 students, divided into 745 students in the control group and 742 students in the experimental group. The result showed that students taught with an improvised computer-based software package had higher mean achievement and attitude test scores than those taught using the conventional strategy. Therefore, teachers should be trained to produce computer-based software packages for teaching mathematics.

Conclusion

This research focused on the effect of improvised computer-based software packages for teaching and learning mathematics on students' learning outcomes, achievement, and attitude toward mathematics. The result showed significant differences in the achievement and attitude towards mathematics of students taught with improvised computer-based instructional software in mathematics and those taught using conventional strategy. Those taught using improvised computer-based software packages for teaching and learning mathematics had higher mean achievement and attitude test scores. Hence, the improvisation of computer-based software packages for teaching and learning mathematics should be encouraged.

Recommendations

Based on the findings of this study, it is recommended that:

1. Teachers should be trained to produce computer-based software packages for teaching and learning.
2. The training should be done by:
   2.1. Teachers themselves involve computer programmers to help them
   2.2. School authorities sponsoring the teachers for the training
   2.3. The Government
   2.4. Community
   2.5. Donor agencies
   2.6. Philanthropist
   2.7. An international organization like UNESCO
3. The use of teacher-made computer software packages should be encouraged to reduce the scarcity of software packages for effective teaching and learning.

References

Benitex, I. P (2016). Development of computer-based training to supplement lessons in fundamentals of electronics. *Asia Pacific Journal of Multidisciplinary Research*, *4*(2), 122 -129.

Chen, J. W., & Shen, C. W. (1989). Components for institutional software development. *Educational Technology*, *29*(9), 9–5.

Garba, M., Umar, Z., & Hu, S. (2010). Development of computer-based learning system for teaching and assessing mathematics. *International Journal of Natural and Applied Sciences*, *6*(2), 196–205.

Ingram, R. (1985). Simultaneous interpretation of sign languages: Semiotic and psycholinguistic perspectives. *Multilingua*, *4*(2), 91-102. https://doi.org/10.1515/mult.1985.4.2.91

Michael, J. P., & Igenewari, S. (2022). Assessment of computer-based instruction (CBI) techniques in teaching of computer studies in Rivers State senior secondary schools. *International*

Journal of Scientific and Research Publications, *12*(2), 376–385. http://dx.doi.org/10.29322/IJSRP.12.02.2022.p12251

Pappas, C. (2014). *Institutional design models and theories: Computer-Based instructional theory.* E-learning Industry. Retrieved March 12, 2024, from https://elearningindustry.com/computer-based-instruction-theory

Rohmah, F. N. (2019). Developing computer-based instructional media for English speaking skill at senior high school. *Journal of English Teaching, 5*(9), 63-76.

Rosenthal, L. (1976). Model for implementation for computer-based instructional system. *Educational Technology*, *16*(2), 13–22.

Rouse, M. (2022, June 17). *What Does Computer-Based Learning Mean*? Technopedia. Retrieved March 12, 2024, from https://www.techopedia.com/definition/11167/computer-based-learning-cbl

Ruliah, D., Syahrial, Z., & Muchtar, H. (2019). The computer assisted instruction model based on a combination of tutorial model and drill and practice model in the instructional design of database systems in information technology colleges. Universal *Journal of Educational Research*, *7*(9A), 117–124. https://doi.org/10.13189/ujer.2019.071614

van Merriënboer, J. J. G., & Martens, R. (2002). Computer-based tools for instructional design: An introduction to the special issue. *ETR&D* 50, 5–9. https://doi.org/10.1007/BF02504980

Received: 09 May 2024
Revised: 01 June 2024
Accepted: 12 June 2024