

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/265991384>

# Evaluating the Effectiveness of the CPP-Tutor, an Intelligent Tutoring System for Students Learning to Program in C++

Article · January 2009

CITATIONS

38

READS

373

1 author:



[Samy S. Abu-Naser](#)

Al-Azhar University - Gaza

260 PUBLICATIONS 4,754 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



IT Governance [View project](#)



Deep Learning [View project](#)

## Evaluating the Effectiveness of the CPP-Tutor, an Intelligent Tutoring System for Students Learning to Program in C++

Samy S. Abu Naser

Faculty of Engineering & Information Technology, Al-Azhar University Gaza, Palestine.

**Abstract:** In an attempt to support the growing development of the C++ programming language and to press forward web-based tailored teaching, the C++ Intelligent Tutoring System (CPP-Tutor) was designed and developed. CPP-Tutor expertly checks the student's submitted solution and determines the appropriate feedback. In this research, we describe an experiment in which we try to measure the effectiveness of the CPP-Tutor. This was accomplished by comparing the traditional method of teaching (instructor and textbook) and CPP-Tutor of an introductory course in C++ programming to freshman students in the faculty of Engineering and Information Technology of Al-Azhar University in Gaza. A group of students were taught C++ programming concepts using CPP-Tutor and a second group was taught the same concepts in parallel by traditional methods of teaching. Both groups were coordinated for similar background knowledge of the topics being taught. Post testing revealed that the CPP-Tutor group achieved significantly higher scores than the group taught using the traditional method. Furthermore, The CPP-Tutor group showed that the retention of specific topic of knowledge was better than the traditional method group.

**Key words:** Experiment, C++, Tutor, Intelligent Tutoring System, Programming, Learning

### INTRODUCTION

This paper presents an experiment for evaluating the effectiveness of the C++ Intelligent Tutoring System (CPP-Tutor) used by students in the Faculty of Engineering and Information Technology at Al-Azhar University in Gaza. A brief overview of the CPP-Tutor, the experiment methods used, and the results are presented.

**CPP-Tutor User Interface:** The user interface implemented in CPP-Tutor is representative of fundamental features of a professional programming Integrated Development Environment (IDE)<sup>[5,6]</sup>. Students are presented with a problem, the problem specification, the code editor, and a number of push buttons with which to interact with the CPP-Tutor. For example, once the student is ready to submit the solution of the problem to CPP-Tutor, the 'Check' button may be pressed. The student can request to see the hints by pressing "Hint". The hints are dynamically generated based on the problem details and the student's submitted code of the problem. At any time, the student can see his performance by pressing the 'Stats' button then the student is presented with a summary of his or her performance based on statistics including problems attempted, problems solved, number of attempts on a problem and problem difficulty. The

student, at any time, may explicitly request from CPP-Tutor to view the solution, Exit from the current problem and select a new one. The CPP-Tutor user interface is shown in fig 1.

**CPP-Tutor Algorithmic Design:** The C++ Intelligent Tutoring System's intelligence is accomplished by an embedded logic module that performs a number of tasks behind the scenes. The logic module implements a sophisticated scanner and parser that autocorrects the student's code when appropriate as well as generates a number of parse trees that have a little variation. This module then attempts to compile the best trees to ascertain the most likely path the student 'intended' to take. With this knowledge, CPP-Tutor can effectively and efficiently tutor the student<sup>[1]</sup>.

The Goals of the Logical Module Are:

- Wisely identifying the 'intent' of the student;
- Analyzing the submitted code of the student's
- 'Auto correcting' simple syntactic errors where appropriate; for example, replacing "float" by the key word "float", "Int" by "int", etc.
- Studying individual student's misunderstanding, and classifying the types of errors he or she makes;
- Constructing a 'modified student code' that will

**Corresponding Author:** Samy S. Abu Naser, Faculty of Engineering & Information Technology, Al-Azhar University Gaza, Palestine.

compile by the C++ Compiler engine or make student code nearer to a state for successful compilation,

- Creating a 'modified student code' that will satisfy the specifications of the program or make the code nearer to fulfilling program specifications,
- Providing the student with the proper hints as necessary.
- Providing the students with proper feedback.

The CPP-Tutor logical module, combined with a well-defined student modeling mechanism and dynamic hints generation capabilities, enables CPP-Tutor to significantly improve the performance of students in C++ programming.

**CPP-Tutor Authoring Tool:** An authoring tool was developed which provides the instructor a convenient means to add problems to the Knowledge base for CPP-Tutor to use. This is a very easy process because the instructor only needs to provide the problem statement, the problem description, and the problem solution.

As a result, the CPP-Tutor authoring tool is intended to be very user friendly and easy to add problems of different levels of difficulties. Once the instructor has submitted a C++ problem it is immediately available to CPP-Tutor and students of the system. Fig2 shows the authoring tool user interface.

The authoring tool provides a means to view all the problems in the session set and edit selected problems<sup>[4]</sup>. CPP-Tutor carefully examines the student's submitted code based on the problem description, specification, and problem solution code to determine the appropriate feedback to the student. This ensures the greatest degree independent knowledge creation for each student<sup>[2]</sup>.

**Investigation:** This study involved freshman students of an introductory programming language C++ in the faculty of Engineering and Information technology at Al-Azhar University of Gaza. The aim of this investigation was an attempt to measure the effectiveness of CPP-Tutor which was used to supplement the traditional method of learning environment.

This means that the lecturer is no longer the only focus of the learning environment. For the purpose of this study, effectiveness was assessed by student performance in the pre and post tests.

Using CPP-Tutor system means that the student has far more control over his learning experience. The potential advantages held by the CPP-Tutor are: students can learn to program C++ any day, any time and any where.

## **MATERIALS AND METHODS**

A sample of 62 freshman students taking the course C++ programming was selected to participate in this study. The student sample was divided into two groups coordinated for similar background knowledge of the C++ material presented.

**Group 1(the Control Group):** This group was taught using the conventional teaching method (lecture and textbook) and it consists of 31 students.

**Group 2(the Test Group):** This group was taught using the conventional teaching method supplemented with the Intelligent Tutoring System CPP-Tutor and it consists of 31 students.

The independent variable is the teaching treatment.

**Hypothesis:** An important consideration of this investigation was a comparison of the effectiveness of the CPP-Tutor method of learning against the traditional method of learning environment. For the purposes of this study, effectiveness was defined by student performance during the testing procedures. Three null hypotheses were tested:

- There are no significant differences in performance between students in both groups in the pre-testing.
- There are no significant differences in performance between students in both groups after first post test.
- There are no significant differences in performance between students in both groups after second post test (retention test).

**Testing Procedure:** The testing procedure consists of three stages:

- An initial test was given to assess the background knowledge of the subject matter to be presented. The purpose of this test was to enable us to divide the sample into two groups, a control group and a test group, both synchronized for background knowledge of the C++ subject material.
- A post test was administered at the end of the one month training period in order to measure the student performance.
- Another post test was given to both groups, one month later from the first post test, in order to measure their retention of the subject taught. The control group was taught in a traditional lecture situation. The test group was also given this standard teaching but supplemented with the CPP-Tutor.

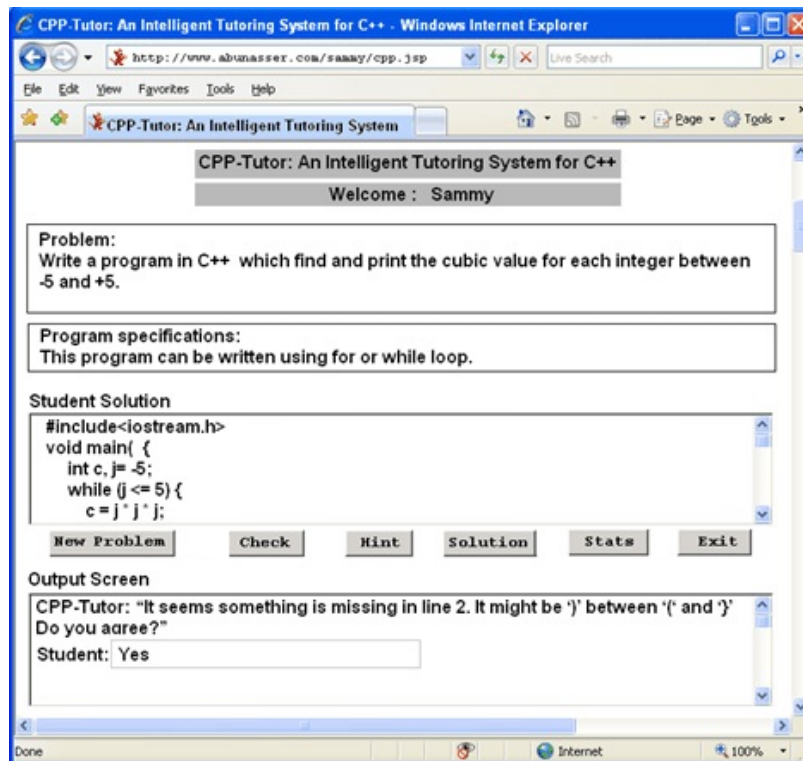


Fig. 1: A snapshot of the CPP-Tutor user interface

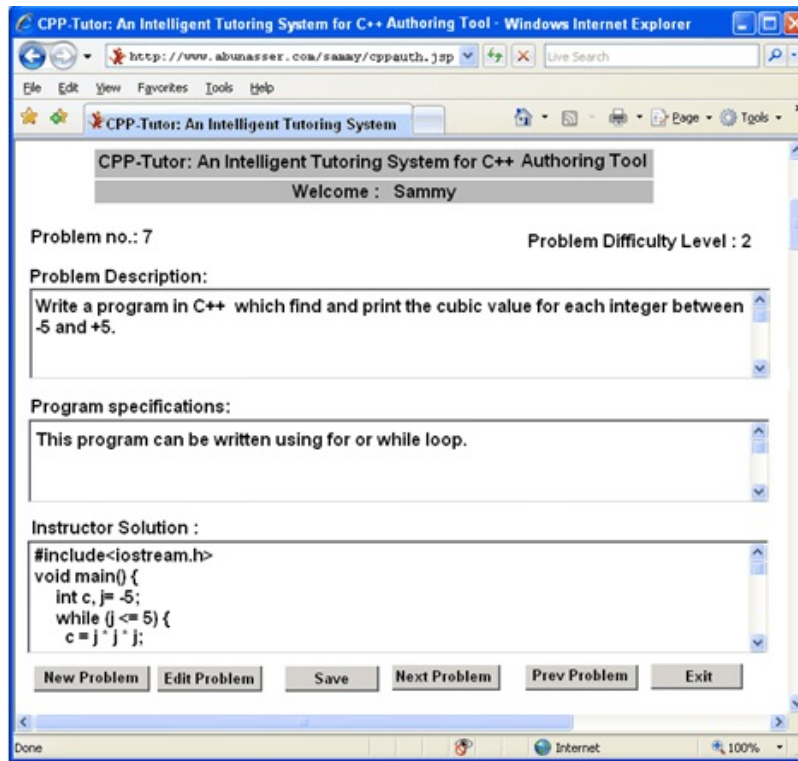
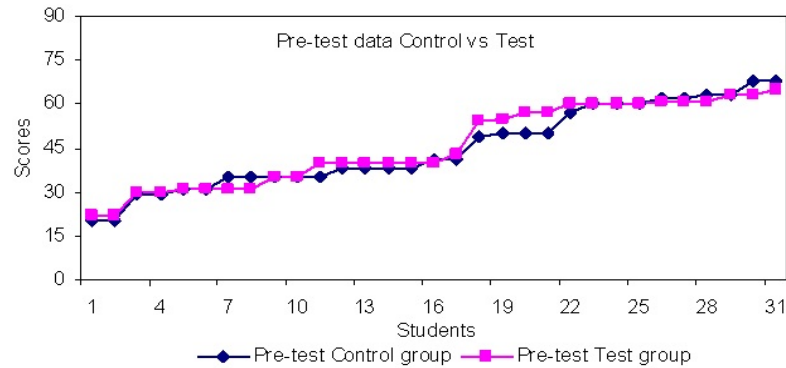
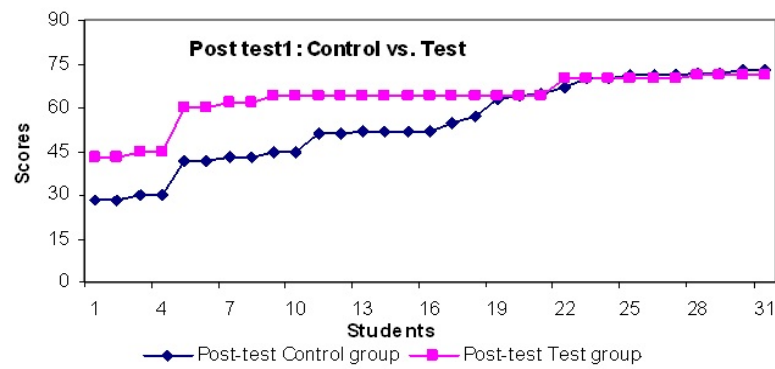


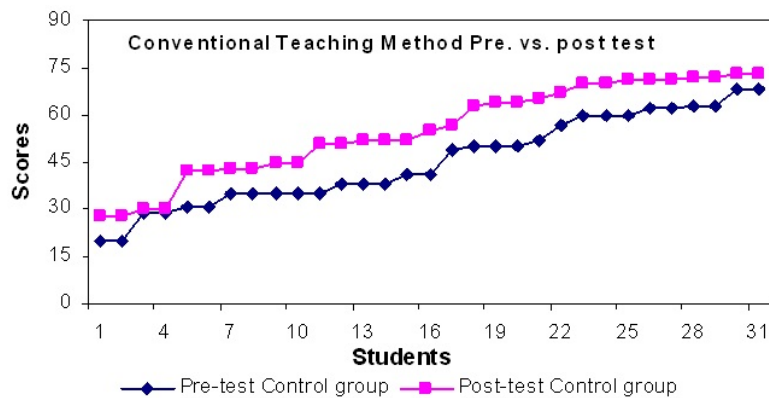
Fig. 2: A snapshot of the CPP-Tutor authoring tool



**Fig. 4:** Pre-test Control vs. Test Group



**Fig. 5:** Post-test Control vs. Test Group



**Fig. 6:** Pre-Test vs. Post-Test for Control Group

**Statistical Analysis:** This section presents the statistical results obtained from the investigation. The data collected was concerned with doing: an initial assessment of the two teaching methods and a comparison of the effectiveness of both teaching methods.

The performances of students in the pre and post tests in both groups are shown below.

**Initial Assessment:** In Table2 of the above results show that hypotheses1, can be accepted ( $t=-1382$ ,

$p=0.177$ ) showing that there is no difference in the knowledge level between the control and test groups based on initial student performance.

A student performance which was determined by pre and post test scores is exemplified by the calculated means of student scores and the relevant standard calculations in the tables below.

**Post Test 1 Assessment:** The results ( $t=-5.372$ ,  $p=0.001$ ) show that hypothesis2 can be rejected ( $p<0.05$ ) showing there is a significant difference in the

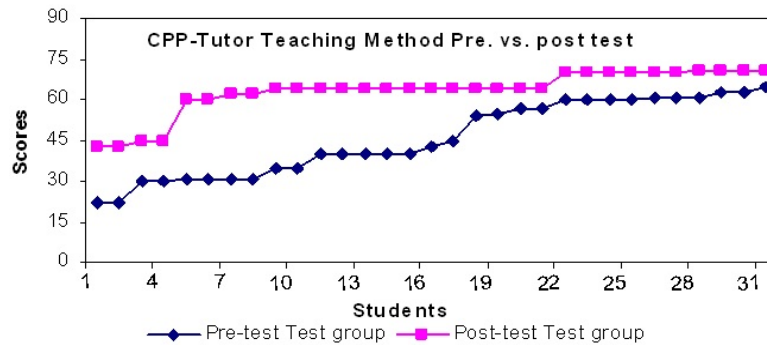


Fig. 7: Pre-Test vs. Post-Test for Test Group

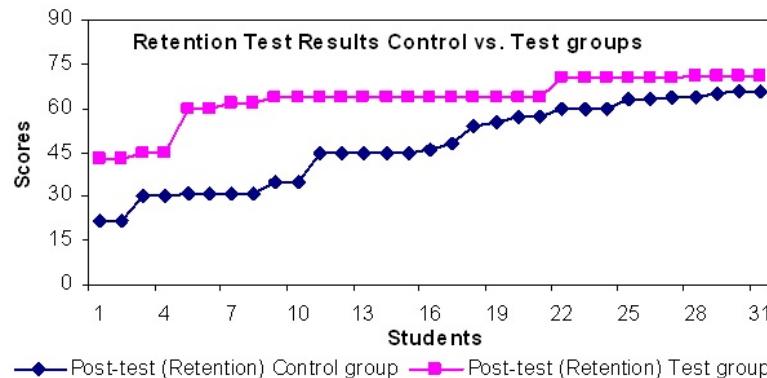


Fig. 8: Post-test2 for control & Test Group

performance of the two groups. This result can be acceptable due to the teaching method.

**A Comparison of the Control Group, Before and after Post Test 1:** Table 5, shows a 10% increase in the mean value of post-test over the pre-test of the control group.

**A Comparison of Test Group, Before and after Post Test 1:** Table 6, shows approximately 17% increase in the mean value of post-test over the pre-test of the test group.

**Retention Test (Post-Test2):** The results ( $t = -9.450$ ,  $p = 0.0001$ ) show that hypothesis3 can be rejected ( $p < 0.05$ ) showing there is a significant difference in the performance of the two groups. This result can be acceptable due to the teaching method.

**Discussion:** The results of the analysis show that whilst there was initially no difference in the performance of the two groups at the pre testing phase of the investigation; subsequent testing revealed differences regarding the performance of the students for both phases of post testing. This illustrates that the addition of the CPP-Tutor to the learning process improved the knowledge of the students in subject

Table 1: Pre-Test Control vs. Test Group Results

Group	Mean	Standard Deviation
Control - traditional environment	44.87	14.23
Test - using CPP-Tutor system	45.74	13.92

Table 2: Paired t-test Results

Paired t-test	-1.382
Probability value	0.177

Table 3: Post-Test1 Control vs. Test Group

Group	Mean	Standard Deviation
Control traditional environment	54.84	14.65
Test using CPP-Tutor system	63.10	8.23

Table 4: Paired t-test Results

Paired t-test	-5.372
Probability value	0.001

Table 5: Traditional teaching method results

Traditional Teaching Method	Mean	Standard Deviation
Control group Pre-test	44.87	14.23
Control group Post-test	54.84	14.65

Table 6: CPP-Tutor teaching method result

CPP-Tutor Teaching Method	Mean	Standard Deviation
Test group Pre-test	45.74	13.92
Test group Post-test	63.10	8.23

Table 7: Post-Test2 Control vs. Test Group Results

Group	Mean	Standard Deviation
Control - retention test	47.45	14.27
Test - retention test	63.10	8.23

**Table 8:** Paired t-test Results

Paired t-test	-9.450
Probability value	0.0001

material presented. It can be also seen from Table 6 and 7 that when the teaching process is considered, the percentages of improvement were higher when the CPP-Tutor component was added.

The second post test was applied to determine whether the CPP-Tutor had any effect on the retention of the subject material after one month from first post test. This is where the greatest difference between the groups was observed (Table 7).

Conventional teaching method focuses on the material presented to the students, and thus usually there is no much time available for considering related issues to understand the material. Since the CPP-Tutor is not constrained to time or location, students can reach its learning environment easily and review the learning material, test their knowledge and understanding at any time.

**Conclusion:** In this paper, we have presented an evaluation of the CPP-Tutor, an Intelligent Tutoring System for student learning to program in C++. The results of this study suggest that CPP-Tutor can be used as an effective teaching method, however, that does not mean CPP-Tutor can replace human instructors. Intelligent Tutoring Systems can be effective only under the supervision of a human instructor<sup>[7,8,9,10]</sup>.

Finally, CPP-Tutor removed some of the formality of traditional learning, helped students raise their self confidence and become more effective in managing their time and to develop study habits that result in enhanced learning.

Further research is needed in the possible use of other (ITS) models, because ITS proved to be an effective tool for aiding students in their study at the university level.

## REFERENCES

1. Abu Naser, S. S., 2008a. Developing an Intelligent Tutoring System for Students Learning to Program in C++. *Inform. Technol. J.*, 7(4).
2. Graesser, A. C. and Person, N. K., 2001, Teaching tactics and dialog in auto tutor, *IJAIE*, 12: 12-23.
3. Koedinger, K. R. 2001, Cognitive tutors, in *Smart machines in Education* Forbus, K. D. and P. J. Feltovich (Eds.), Cambridge, MA: MIT Press), pp: 145- 167.
4. Rowe C. N. and P. T. Galvin, 1998, An authoring system for intelligent procedural-skill tutors., *IEEE: Intelligent Systems*, 14: 61-69.
5. Abu Naser, S. S., 2008b. Developing Visualization Tool for Teaching AI Searching Algorithms. *Inform. Technol. J.*, 7(2): 350-355.
6. Conlan, O., C. Hockemeyer, V. Wade, D. Albert, and M. Gargan, 2002. An architecture for integrating adaptive hypermedia service with open learning environments. In *proceedings of ED-MEDIA 2002*, volume 1 of *World Conference on Educational Multimedia June 24-29, 2002.*, Hypermedia and Telecommunications, pp: 344-350.
7. Abu Naser, S. S., 2006. Intelligent Tutoring System (ITS) for Teaching Database to Sophomore Students in Gaza and its Effect on their Performance. *Inform. Technol. J.*, 5(5): 916-922.
8. Anderson, J. R., A. T. Corbett, K. R. Koedinger, and R. Pelletier, 1995. Cognitive Tutors: Lessons learned. *JLS* , 4: 167-207.
9. Mark, M., and Greer, J. E, 1995. The VCR tutor: Effective instruction for device operation. *JLS*, 4(2): 209-246.
10. Shute, V. J., and Glaser, R., 1990. A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments*, 1: 51-77.