Research Article

Design and Development of an Integrated Online System Support for C/C++ Programming

Urachart Kokaew* and Somjit Arch-Int
Department of Computer Science, Faculty of Science, Khon Kaen University
123 Moo 16, Mittraphap Road,Tambon Muang, Khon Kaen 40002 Thailand.

Poonphon Suesawaluk
Graduate School of eLearning Assumption University,
88 Moo 8 Bang Na-Trad Km. 26 Bangsaothong, Samuthprakarn 10540 Thailand.

Abstract

One of the main problems for a lecturer of a computer programming course is the difficulty in providing quick feedback to every student's coding solutions, especially in larger classes. To solve this problem, this study proposed an Integrated Online System Support for C/C++ programming language instruction. The designed system was based on the problem-solving model within the context of C/C++ programming and Gagné's nine concepts. The system was developed as an online environment. The target group was the undergraduate students who were enrolled in the Introduction to Computer Programming course in the second semester for the academic year 2013 in the Department of Computer Science, Khon Kaen University, Khon Kaen, Thailand. We investigated the impact of the system on the students' achievement and their satisfaction with this system. This area of the research was carried out using a learning achievement test and a questionnaire using the online environment. The data were analyzed using mean, standard deviation, an independent-samples T-test and a paired-samples T-test. The results of this research showed that the students' post-test scores were significantly higher than their pre-test scores at .05 significance level. The questionnaire given at the very end of the experiment showed the satisfaction of participating students towards the system. Each question in the questionnaire showed students to be satisfied.

Keywords: Design and development model; Web-based learning; Online learning.

1. Introduction

Computer programming is an important and necessary skill for students in many areas inside the world of computer technology. Unfortunately, programming courses have acquired a reputation for being too difficult [1]. There are large numbers of students enrolling in programming courses such as computer science, computer engineering, information technology, and geographic information science, all of which will require some form of programming ability. However, due to large class sizes, students may not receive the attention needed for such a demanding subject.

Jenkins (2002) pointed out that, "If computing educators are ever to truly develop a learning environment where all the students learn to program quickly and well, it is vital that an understanding of the difficulties and complexities faced by the students are developed" [1]. Higher education will increasingly use electronic meetings where eLearning will be seen as a way for colleges

***Correspondence** : urachart@kku.ac.th

and universities to reach more students without losing traditional classroom learning.

A review of previous studies regarding the use of eLearning in computer programming courses show that there is still an inability to give near real time responses and feedback on a learner's coding. It is impossible to give quick feedback for every student's solution in large classes. Checking thousands of pages of code is not an efficient use of the instructor's time. Furthermore, students frequently copy source code from others for submission. A major challenge in computer science education is the ability to provide an effective learning environment for students while contending with ever increasing class sizes [2]. There is a lack of tools that can provide pedagogical support for the learning and teaching of programming. As stated in [3], "Fully featured integrated development environments (IDEs) overwhelm novice programmers and do not have all the features that support teaching".

The conclusions drawn from these previous studies show two main problems. First, there is no quick response or feedback system for the learners within real-time and integrated environments. Second, the issue of class sizes makes it virtually impossible to manage computer science classes, thus preventing most of the students from acquiring adequate programming skills.

A major challenge in computer science education is lack of the ability to provide an effective learning environment for students. This challenge has led to various research approaches in technology-enhanced learning environments such as an Environment Learning to Program (ELP), which provides an interactive web-based environment for teaching programming [2]. Groupware enabled Integrated Learning and Development (GILD) is also an integrated learning and development environment for programming [3] as is the Virtual Education System for the C Programming Language [4]. These systems have advantages and disadvantages, however, none of the research reports applied their designs to instructional models in education [5]. Our research designed a model that follows the general guidelines for the design and development of instructions in nine concepts by Robert Gagné [6]. His nine concepts are suited for designing lessons in computer assisted instruction, or web-based learning systems and problem-solving model within the context of C/C++ programming [7].

This study is important because it will contribute to the development of an integrated online system support for students to learn the C and C++ programming languages. The developed system was implemented to help instructors and encourage students to study computer programming more effectively. The use of this system should lead to an improvement in the academic performance of students.

The problems mentioned in the introduction relating to teaching computer programming courses are faced by the authors of this paper on a daily basis. We have dealt with this problem for years and believe that it needs be addressed sooner rather than later. The aim of this research is to develop a system that can be rolled out in institutions facing the same problems.

2. Materials and methods

The purpose of this study was to design and develop an Integrated Online System Support for C/C++ programming language. This research design are based on the five phases---Analysis, Design, Development, Implementation, and Evaluation (ADDIE Model)[8]---which represent a dynamic, flexible guideline for building effective training and performance support tools. The ADDIE model is a generic process traditionally used by instructional designers and training developers. This model supports a step-by-step process that helps specialists in planning and creating their training programs.

The five phases of this research design as follows:

2.1 Analysis Phase

The first phase is to study the theoretical framework of the related theories including theories of the problem-solving model within the context of C/C++ programming and Gagné's nine concepts. So far, several web-based learning systems in many computer science programs have implemented some parts of this theory in their systems. Gagné's nine concepts for effective learning are as follows[5]:

1) Gain attention

There are methods to grab the learner's attention to present a problem or a new situation. For example, storytelling, demonstrations, presenting a problem to be solved, doing something the wrong way (the instruction would then show how to do it the right way), telling why it is important, etc.

2) Specify Objective

Specifying the objective can help the learners to organize their thoughts. For example, describing the goal of a lesson, explaining what the learners will be able to accomplish and how they will be able to use the knowledge. By utilizing this idea in the system design, after the learner is motivated to learn, the system informs the specific objectives of each lesson to the learner

3) Activate prior knowledge

Activating prior knowledge can help learners to build on their previous knowledge or skills. For example, reminding the learners of prior knowledge relevant to the current lesson or providing learners with a framework that helps learning and remembering. To use this concept in the system design, the learner has to do an overall pre-test (for all lessons) before starting any lesson and before each pre-test prior to learning each lesson.

4) Present the content

Sequencing the content or learning events by using Learning Strategies could help order the lesson.

5) Guide learning

These are instructions on how to learn with the subject matter, such as providing examples. To use this concept, we present the examples which are consistent with the contents while the learner is doing the exercises. This could help the learner understand the contents.

6) Elicit performance

Practice by letting the learner do something with the newly acquired behavior, skills, or knowledge. For example, asking questions. To utilize this idea, the system asks the learner questions while learning the lesson

7) Provide feedback

This is providing specific guidance to show the veracity of the learner's response from a test, quiz, or verbal comments. To employ this idea, during the exercise performance, the learner will be given feedback

8) Assess performance

The post-test is utilized for determining if the lesson has been learned. This should be followed by general progress information to the learner. To utilize this notion in the system design, the learner has to do an overall post-test (for all lessons) after finishing any lesson.

9) Review and transfer

This is the information supplied to the learner about similar problem situations such as providing additional practice, taking the learner in a transfer situation, reviewing the lesson, etc.

2.2 Design Phase

The second phase is to design the model according to Gagné's nine concepts, which are used to solve the learning problems of current educational practices. The model supports the designer's work, forming the foundation for evaluation and research as illustrated in Fig. 1. The model consists of four components as follows:

2.2.1 Information module

The information module is the domain knowledge of the problem-solving model within the context of C/C++ programming. The problem-solving model is a part of the instructor's information in teaching.

2.2.2 Pedagogical module

The pedagogical module contains the rules for controlling the process in which a topic is presented to a learner under Gagnѐ's nine concepts.

2.2.3 Learner module

The learner module stores the information relevant to each individual learner. This module is responsible for tracking the learner's programming performance in the system in order to record the learner's feedback to the system.

2.2.4 Knowledge Management module

This module consists of a management module for the contents, pre-tests and post-tests for each learner's profile. This is a module for the instructor and learners while they are using the system.

2.3 Development Phase

The third phase is to develop the model into appropriate tools. This phase is concerned with the application of the proposed model in order to build computer software. The online compiler Codepad (developed by Steven Hazel) was used as a compiler system before the source code is submitted to the server as shown in Fig. 2. DevFlowcharter, a program developed by Michal Domagala, was adopted to help students draw flowcharts and generate code. DevFlowcharter was provided only in the lesson on flowcharts in UNIT 1.

2.4 Implementation Phase

The fourth phase is to implement the system in the classroom according to the model shown in Fig. 1. A use case diagram for the system is shown in Fig. 3. The actual implementation was in a computer lab containing Internet-connected PCs running Windows 7. The minimum specification of the PC was 1.0 GHz CPUs with 1 GB of RAM. There was a server with 2GB of RAM running the UNIX operating system with Apache 2.2.15, PHP 5.3.27 language, and MySQL 5.1.72.

The implementation phase began when the students registered to the system according to Fig. 4 and then logged on the system through the window screen shown in Fig. 5 during the laboratory period. The contents, pre-tests and post-tests consisted of ten units. The tests were designed to collect data according to students' skill in C/C++ programming. There were ten units. Each unit was covered in a 2-hour period of laboratory work once a week. Each laboratory period was divided into 3 parts: 40 minutes for a pre-test, 40 minutes for learning through VDO watching as shown in Fig. 6, and 40 minutes for a post-test. Afterwards, the instructor would provide the feedback to students through the system. The students could view their scores and the instructor's comments as shown in Fig. 7.

2.5 Evaluation Phase

The fifth phase is an assessment of the potential of the application software model. The final part is a discussion of the results. This phase is concerned with assessing the scores from pre-tests and post-tests and also processing the data from the responses to a questionnaire. The sample group consisted of the undergraduate students who had enrolled in the Introduction to Computer Programming course in the second semester of the academic year 2013 in the Department of Computer Science, Khon Kaen University, Khon Kaen, Thailand. T-test could be applied for a sample size smaller than 30 according to the number of enrolled students for each section. The data were collected from two groups of students who were enrolled in the

Introduction to Computer Programming course, namely the experimental group (IPS_Online) of 27 students and the control group (Dev C++) of 27 students.

In order to compare the computer programming background of the two groups of students, we applied independent-samples T-test. Both groups of students were provided the pre-test before the first programming activity began in each lesson in the laboratory. To determine the impact of the system on students' achievement from this system, we analyzed the data using a paired-samples T-test for the IPS_Online group only. Lastly, to investigate the impact of the system on students' satisfaction, we delivered an online questionnaire to the IPS_Online group one week before the end of the course. The feedback was obtained from the students' overall satisfaction toward the application, and the information was presented in terms of means and percentages.

The participants were given one unit per week for a total of ten units. The laboratory period for each week lasted two hours. The control group was provided 40 minutes for a pre-test, 40 minutes for studying from the handout, and 40 minutes for the post-test with Dev C++ compiler. Then, their files were saved into portable drives and submitted to the instructor during the laboratory period. The experimental group was given 40 minutes for a pre-test, 40 minutes for learning through VDO watching as shown in Fig. 6, and 40 minutes for a post-test. After that, the teacher provided the feedback to the students through the system. The students received the feedback by viewing their scores and the instructor's comments as shown in Fig. 7.

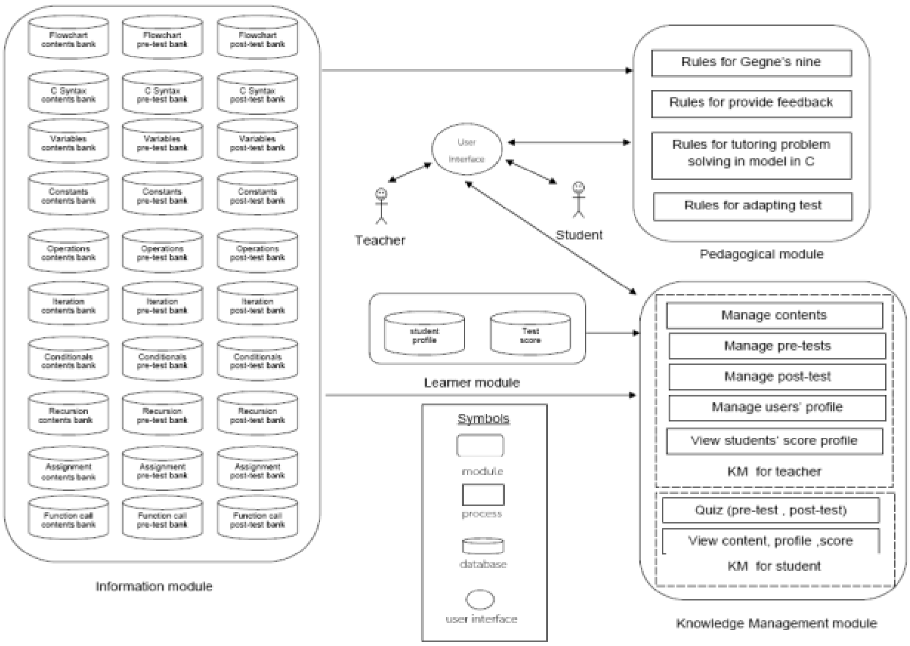


Fig.1. The software architecture for the integrated online system support for C and C++ programming language instruction.

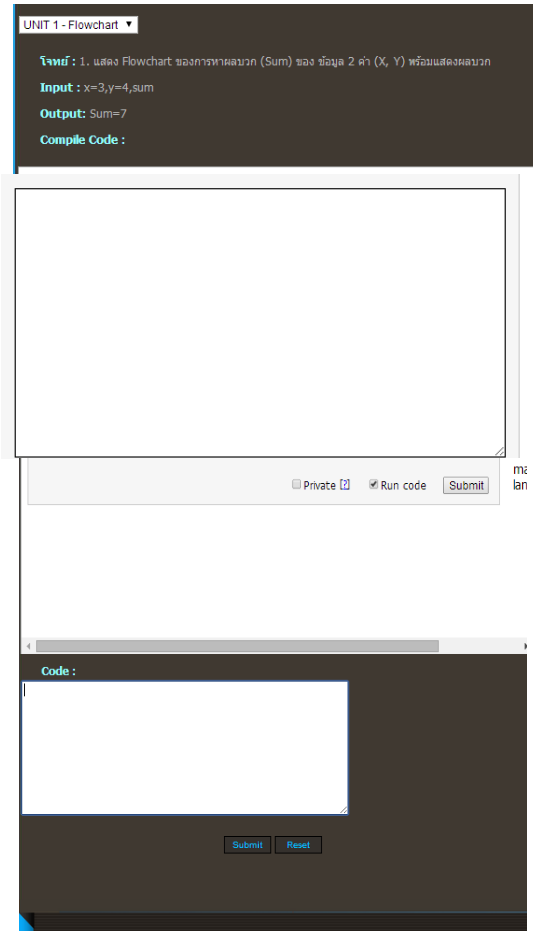


Fig.2. The page for source code experimentation and submission.

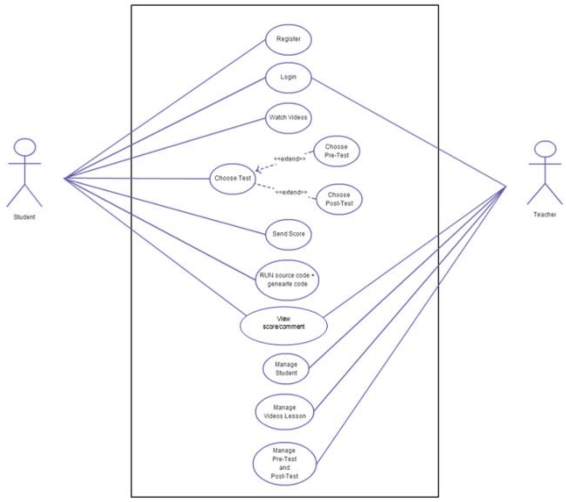


Fig.3. A use case diagram for the integrated online system support for C and C++ programming language instruction.

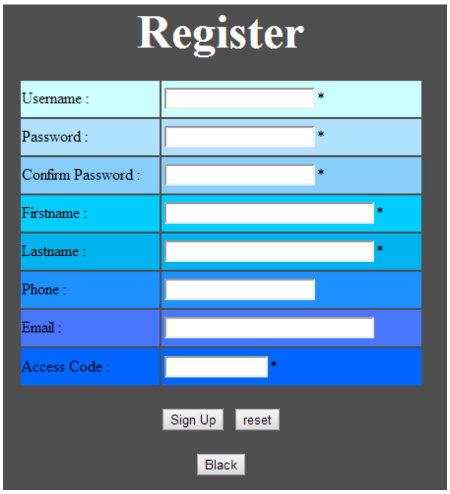


Fig.4. The registration form.

Fig.5. The main page of the system.

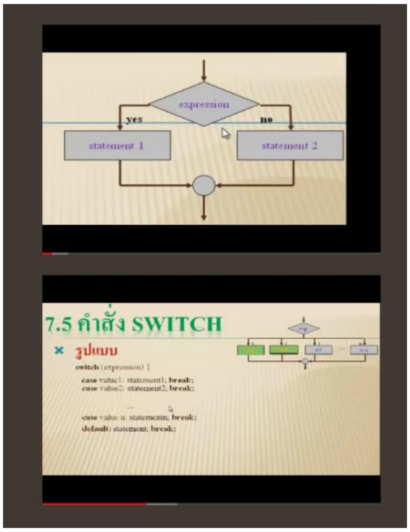
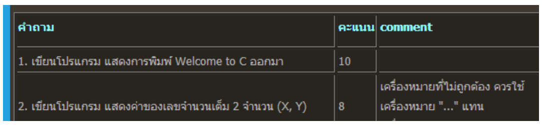


Fig.6. The window for VDO viewing.



คำถาม	คะแนน	comment
1. เขียนโปรแกรม แสดงการพิมพ์ Welcome to C ออกมา	10	
2. เขียนโปรแกรม แสดงค่าของเลขจำนวนเต็ม 2 จำนวน (X, Y)	8	เครื่องหมายที่ไม่ถูกต้อง ควรใช้เครื่องหมาย "..." แทน

Fig.7. A window for students to view their scores and the instructor's feedback.

3. Data Collection and Analysis

The quantitative data collected in this study came from the pre-test and post-test scores for all lessons and the responses to the questionnaire that were collected afterwards. The data were collected from 2 sections of students who were enrolled in the Introduction to Computer Programming course in the second semester of 2013, namely the experimental group (IPS_Online) and the control group (Dev C++). The research study started at the beginning of the semester in November 2013.

All of the data were analyzed using SPSS. The independent-samples T-test and the paired-samples T-test routines provided in SPSS were used to measure if there was any statistically significant difference in the scores of the IPS_Online group and the Dev C++ group. Specifically, the independent-samples T-test was used to assess the statistical difference between the IPS_Online and the Dev C++ groups while the paired-samples T-test was used to measure the progress and the learning gains within the IPS_Online group.

For the questionnaire assessment of students' satisfaction, a five-point Likert-type questions method was selected, with the aim of measuring how satisfied the IPS_Online group of participating students felt toward the application system. The Cronbach's Alpha Reliability Test (CART) was performed in order to test the reliability of the questionnaire. The content of the questionnaire was an assessment of the Students' satisfaction toward the system, and it was divided into three sets:
- a set based on the six strategies for web-based formative assessment [9].
- a set based on the Gagné's nine concepts [5-6].
- a set based on the problem-solving model within the context of C/C++ programming [7].

Each question was a five-point Likert item ranging from "strongly satisfied" to "strongly unsatisfied".The CART was run on a sample size of 20 students. The CART for the questions based on the six strategies for web-based formative assessment [9] yielded an alpha coefficient of 0.887 (high). The CART for the set of questions based on Gagné's nine concepts [5-6] yielded an alpha coefficient of 0.810 (high). The CART for the set of questions based on problem-solving model within the context of C/C++ programming [7] yielded an alpha coefficient of 0.838 (high). We note that Cronbach's alpha reliability coefficient normally ranges between 0 and 1 and that, the closer the coefficient is to 1, the greater the internal consistency of the items in the scale. Our results yielded alpha coefficients that were all higher than 0.8. This clearly shows that the questionnaire given to the IPS_Online student at the end of the experiment was extremely reliable. This is based on [10] who indicated that a Cronbach alpha coefficient higher than 0.8 provides a good measure of internal consistency of items in the scale.

4. Results

All participating students in both groups were administered a pre-test. The pre-test consisted of ten units, each unit being the subject of a subsequent unit. At the end of the experiment, all participants were given a post-test and a questionnaire in order to capture their attitudes towards the system environment. An independent-samples T-test was performed in order to assess whether there was a statistically significant difference between the students' performance before the experiment (pre-test) and that at the end of the experiment (post-test). The results of these tests showed a pre-test p–value of 0.867 (not significant) and a post-test p-value of 0.001 (significant) as shown in Table 1.

Table 1. Independent-samples T-test for the Pre-test and Post-test (n= 27).

Test	Group	Mean	SD	t	p
Pre-test	IPS_ Online	42.96	21.41	0.169	0.867^{ns}
	Dev C++	42.04	18.82		
Post-test	IPS_ Online	79.81	12.67	3.515	0.001*
	Dev C++	68.15	11.70		

* $p < 0.05$.

This shows that there was no statistically significant difference between the mean score of the IPS_Online group and that of the Dev C++ group at the beginning of the experiment and that there was a statistically significant difference between the mean scores of the IPS_Online group and that of the Dev C++ group at the end of the experiment. The mean score at the end of the IPS_Online group was significantly higher than the mean score of the Dev C++ group implying that the Integrated Online System Support for C/C++ Programming Language was effective. The IPS_Online group of students who benefited from the system showed significant improvement when compared with the Dev C++ group who did not benefit from the system.

The paired-samples T-test was carried out in order to assess the improvement of the IPS_Online group during the interval between the pre-test and the post-test. The result of the test showed the means of 42.96 (pre-test) and 79.81 (post-test), and a paired-samples T-test Sig. (2-tailed) value of 0.000 (significant).

Table 2. Paired-samples T-Test for the Pre-test and Post-test of IPS_Online group (n = 27).

Pair 1	Mean	SD	t	p
Post-test_Online	79.81	12.67	6.920	0.000
Pre-test_Online	42.96	21.41		

Paired T-Test Sig.(2-tailed) = 0.000 (significant)

The Paired Sample T-test results underline the conclusion that the Integrated Online System Support for C/C++ Programming Language was effective in helping the students to significantly improve their programming skill.

Table 3. Results based on the six strategies for web-based formative assessment [9]. Means and percentages in the satisfaction stage (n = 27)**.**

Items	Mean	%				
		Strongly satisfied	**Satisfied**	**Moderate**	**Unsatisfied**	**Strongly unsatisfied**
It gives me more opportunities to familiarize myself with learning materials.	3.96	22.22	55.56	18.52	3.70	0.00
It allows me to know what I need to study more.	4.00	25.93	51.85	18.52	3.70	0.00
It pushes me to make clear what I did not understand.	4.04	29.63	48.15	18.52	3.70	0.00
It helps me understand my grade status in class.	4.07	29.63	51.85	14.81	3.70	0.00
It helps me understand my classmate's grade status.	3.89	25.93	48.15	18.52	3.70	3.70
It gives me more opportunities to interact with my teacher.	4.04	29.63	48.15	18.52	3.70	0.00
It offers a more efficient way to clarify my incorrect concepts.	4.00	25.93	51.85	18.52	3.70	0.00
It helps me understand my answer history.	3.96	22.22	55.56	18.52	3.70	0.00
It helps me understand what I have learned.	4.04	25.93	55.56	14.81	3.70	0.00
It can motivate me to answer carefully in order to pass test.	4.07	29.63	51.81	14.81	3.70	0.00
It gives me a feeling of success if I pass test.	3.96	25.93	51.85	18.52	0.00	3.70

Note: Five-point Likert-type scale:
5- Strongly satisfied, 4- Satisfied, 3-Moderate, 2- Unsatisfied, 1- Strongly unsatisfied

Five-point Likert-type questions were used in the questionnaire to assess the students' satisfaction. This method was selected with the aim of measuring how satisfied the IPS_Online group of participating students was with the online system. The results were interpreted based on the calculations of the means as shown in Table 3, Table 5 and Table 6 and based on the interpretation from [11] replicated here in Table 4.

Table 4. Interpretation of the Students' Satisfaction [11].

Mean	Interpretation
4.51 – 5.00	Strongly satisfied
3.51 - 4.50	Satisfied
2.51 - 3.50	Moderate
1.51 - 2.50	Unsatisfied
1.00 - 1.50	Strongly unsatisfied

Table 5. Results based on the Gagné's nine concepts [5]. Means and percentages in the familiarization stage (n = 27).

Items A feeling about the system	Mean	%				
		Strongly satisfied	Satisfied	Moderate	Unsatisfied	Strongly unsatisfied
Gain attention	3.85	22.22	51.85	18.52	3.70	3.70
Specify objective	4.00	25.93	51.85	18.52	3.70	3.70
Activate prior knowledge	3.81	22.22	48.15	22.22	3.70	3.70
Present content	3.89	25.93	48.15	18.52	3.70	3.70
Guide learning	3.93	22.22	51.85	22.22	3.70	0.00
Explicit performance	4.00	29.63	48.15	18.52	0.00	3.70
Provide feedback	4.04	29.63	48.15	18.52	3.70	0.00
Assess performance	3.96	22.22	55.56	18.52	3.70	0.00
Review and transfer	3.93	22.22	55.56	18.52	0.00	3.70

Note: Five-point Likert-type scale: 5- Strongly satisfied, 4- Satisfied, 3-Moderate,
2-Unsatisfied, 1- Strongly unsatisfied

Table 6. Results based on the problem-solving model within the context of C/C++ programming [7]. Means and percentages in the familiarization stage (n = 27)**.**

Items A feeling about each unit	Mean	%				
		Strongly Satisfied	Satisfied	Moderate	Unsatisfied	Strongly Unsatisfied
Flowchart	3.81	22.22	48.15	22.22	3.70	3.70
C Syntax	3.93	25.93	51.85	14.81	3.70	3.70
Variables	4.04	29.63	51.85	14.81	0.00	3.70
Constants	3.78	18.52	55.56	14.81	7.41	3.70
Arithmetic operation	3.81	22.22	48.15	22.22	3.70	3.70
Iteration	3.96	22.22	55.56	18.52	3.70	0.00
Conditionals	3.78	18.52	51.85	22.22	3.70	3.70
Recursion	3.74	18.52	51.85	18.52	7.41	3.70
Assignment	3.93	22.22	55.56	18.52	0.00	3.70
Function call	3.81	25.93	44.44	18.52	7.41	3.70

Note: Five-point Likert-type scale: 5- Strongly satisfied, 4- Satisfied, 3-Moderate,
2- Unsatisfied, 1- Strongly unsatisfied

5. Conclusions

In this study, we designed and developed a web-based system from the software architecture for the integrated online system support for C and C++ programming language instruction as shown in Fig. 1. The system was designed to offer Khon Kaen University C/C++ programming students a chance to improve their C/C++ programming skills. The system employed Gagnѐ's nine concepts and problem-solving model within the context of C/C++ programming. The results of this research showed that student's post-test scores were significantly higher than their pre-test scores at .05 significance level. Both the independent-samples and the paired-samples T-test results underlined our conclusion that the system was effective in helping students improve their C/C++ programming skills. The questionnaire given at the very end of the experiment showed the satisfaction of participating students. In conclusion, we believe that the system could enable students to improve their C/C++ programming skills within this learning environment.

6. Further work

This system can be developed further to teach other computer languages. There is also a possibility of using the system to teach other subjects such as Data Structures ,Theory of Computation, etc. The results of the system deployment could be an innovation to contribute to mLearning and uLearning. The results of the research can be examined and compared with others related research in related fields.

7. Acknowledgements

The authors would like to thank Steven Hazel, who developed an online compiler (Codepad), and Michal Domagala, who developed DevFlowcharter. These are the integrated tools used in our system. We also wish to express our appreciation to the students who were enrolled in the Introduction to Computer Programming course in the second semester in the academic year 2013 in the Department of Computer Science, Khon Kaen University, Khon Kaen, Thailand for their cooperation in this study.

8. References

[1] Jenkins, T. On the difficulty of learning to program. 3rd Annual LTSN-ICS Conference, LTSN Centre for Information and Computer Sciences Loughborough University, 2002.

[2] Truong, N., Bancroft P., & Roe, P. A Web Based Environment for Learning to Program. In Proceedings of the 26th Australasian Computer Science Darlinghurst, Australia: Australian Computer Society, Inc. conference - Volume 16 (ACSC '03), Michael J. Oudshoorn (Ed.), Vol. 16, pp. 255-264, 2003.

[3] Cubranic, D., Storey, M.-A., & Ryall, J. A Comparison of Communication Technologies to Support Novice Team Programming, ICSE Education Track, Shanghai, China, 2006.

[4] Moon, I., Han, S., Choi, K.,Kim, D., Jeon, C., Lee, S., & Jeon, H., Virtual Education System for the C Programming Language, ADVANCES IN WEB BASED LEARNING - ICWL 2008, Lecture Notes in Computer Science, (5145), pp. 196-207, 2008.

[5] Kochakornjarupong, D., A Web-based System Design for Enhancing Learning Problem Solving in Artificial Intelligence, The Seventh International Conference on eLearning for Knowledge-Based Society, pp. 62.1-62.1, 2010.

[6] Gagné, R. M., The Conditions of Learning and the Theory of Instruction (4^{th} ed.), New York: Holt, Rinehart, and Winston, 1985.

[7] Kordaki, M., A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative

evaluation. Computers & Education, Vol.54, No.1, pp. 69-87, 2010.

[8] Strickland, A.W., ADDIE. Idaho State University College of Education Science, Math and Technology Education, 2006.

[9] Chao, L.,Wang, T.H.,Yao, K. C.,Chen, F., & Wang, W. L., Studies on Undergraduate Achievement and Attitude of Electrical Theory by Using Web-Based Formative Assessment, 2007.

[10] George, D., & Mallery, P., SPSS for Windows step by step: A simple guide and reference. 11.0 update (4th ed.). Boston: Allyn and Bacon, 2003.

[11] Srisaard, B., Research Introduction. (7th ed.). Bangkok: Suweeriyasan Press, 2002.