Improving K-12's Logical Thinking Abilities using Educational Programming Language 'Dolittle'

SEUNGWOOK YOO¹, SEUNGEUN CHA¹, WONGYU LEE² ¹Dept. of Computer Science Education, Graduate School ²Dept. of Computer Science Education, Colledge of Education Korea University Anam-Dong, Seongbuk-Gu, Seoul, 136-701 KOREA {voosw0810,seungeun.cha,lee}@inc.korea.ac.kr

Abstract: New Korean information education curriculum effective from 2010 specially emphasizes problem solving using computer. Therefore it is necessary to conduct 'programming education' during information education curriculum. Experimental programming courses have been performed for 265 9th grade students and measured improvement of logical thinking ability that is necessary for problem solving. The language used for experiment course was 'Dolittle' and experimental courses were given for 16 weeks, I hour per week, with learning guideline previously prepared. Logical thinking test performed before and after the programming courses showed the positive changes of students logical thinking and also it showed programming education affects sub-factors of logical thinking. As a result of this study, educational programming language that lowered cognitive load for Korean 9th grade information education curriculum aided improvement of logical thinking that is necessary for problem solving using computer.

Key–Words: educational programming language, logical thinking, problem solving

1 Introduction

Generally existing information educations were recognized that they help in improving problem solving ability through logical thinking in order to properly respond to information society and equipped with computer utilizing skills.

But due to fast expansion of personal computers and broad band internet connection, computer skills are more emphasized and information education is misunderstood as training of application software. Information education curriculum is misunderstood as training of application programs or making multimedia contents and it was highly criticized[11].

Recently a new revised information education curriculum in Korea has been proposed for elementary and junior high school students. Under the objective of fostering talented resources that are appropriate for knowledge based information society and understand information science to actively utilize information skills for creative problem solving, revised information education curriculum reinforced basic principle of computer science and composed to harmonize with ICT education.

The structure is composed by application of general information processing processes that are circulation of data collection, analysis, design, processing, creation and sharing. The structure is composed in 4 sections that are 'Composition and operation of information devices', 'Expression and management of information', 'Problem solving method and process' and 'Information society and information technology'. Revised information education curriculum will be applied from 2010 toward 7th grade and higher in order.

Specially emphasized area is 'Problem solving method and process' that significantly emphasizes problem solving strategic algorithm and programming that is computer processing for the algorithm in order to enhance ability to logically propose processes for problem solving.

In information education curriculum for k-12, caution for programming education is not only technical education to foster programmer but 'Management of information flow and processing method for problem solving' for programming education and programming language must be selected that is suitable to the level of K-12 students. The reason is because numbers of limitations must be resolved in order to perform programming educations for K-12 students and especially programming usage method, program writing level, limited education time, inducing interests of students as well as maintenance must be conducted that are suitable for the knowledge development level of the students.

This study is verification of enhancing logical thinking for problem solving of junior high school students who had learned the educational programming language. With detail logical thinking verification for before and after programming language education, the level of logical thinking improvement can be measured and relationship between education of programming language and logical thinking is researched and verify effectiveness of logical thinking that is foundation of problem solving for education of programming toward K-12.

2 Knowledge Background

2.1 New curriculum and problem solving ability

Generally problem solving is defined by 'The act of finding a solution to a perplexing, distressing, vexing, or unsettled question'[2][13]. George Polya[9], in his writing 'How to Solve It' in 1945, he proposed general process of problem solving. First step is to understand problem, second step is establish plan, third plan is to execute the plan and fourth step is to confirm obtained solution. These resolution steps of Polya are basic process to solve general problems to be useful as reference but they are not suitable for computer based problem solving process.

Ministry of Education Human Resource Development that has realized the importance of computer based problem solving method and process emphasized computer scientific factor in new curriculum and composed to enhance problem solving ability through procedure thinking. Especially 'Expression and management of information' and 'Problem solving method and process' required computerized problem solving ability by actual composition of programming through information expression method in computer and learning concept of data processing algorithm.

Among these 'Problem Solving Method and Process', the first step is analysis of problem through various method and learn to express in detail form through conversation, writing or symbol etc and understand problem solving process to design effective resolution method for problems occurred in our life. The basics of programming is to understand concept of variable and learn the usage to declare and utilize variable for handling various types of data, understand types and method of data input and output for programming and write input/output program. Also understand conditional sentence and repetitive sentence and write structural program by utilizing conditional sentence and repetitive sentence.

The second step for problem solving method is summary of algorithm to understand definition and importance of algorithm by researching processes to solve various problems occurred in everyday life, learn various algorithm expression methods such as natural language, flow chart and express algorithm in various form. Also design algorithm for problems of actual life and understand algorithm analysis method to classify the most effective algorithm when various algorithm exist to solve problem and realize algorithm in various methods.

The detail last third step for problem solving method and process is arrangement of data. Understand various methods to arrange data as well as understand features, advantages and disadvantages of each search algorithm and based on the understand of various search algorithm, try to solve search problems in actual life.

Programming education to learn these problem solving method and process gives various cognitive burden to K-12 and especially requires logical thinking.

2.2 Logical thinking and programming education

Piaget based thinking as foundation for human intelligence and defined the thinking as intellectual management to derive answers during problem solving process [5]. Logical thinking or reasoning is a certain phenomena within the mind of human and it means realizing logical relationship.

Logical thinking is performing accurate thinking and determination as well as evaluation process for certain regulation such as relationship or contradiction among objects or ideas.

Similar to this process, programming education strongly require logical relationship recognition for programming objects and logical thinking to lead certain procedures. As a result of study performed by Gorman and Bourne in influence in logical thinking by the level of programming education, they proved programming education is effective in enhancing logical ability[3]. Clement and Gullo proved that programming education is effective in enhancing rumination and logical thinking.

As a result of above relativity between programming education and logical thinking, as programming languages are based on basic logical concept, they require logical thinking and abstract reasoning ability to influence in recognition development of learners. Therefore as program itself is logic for any programming language, it contains logical factor. Consequently, programming education shall influence in forming logical thinking based on problem solving in any ways through the process of recognition of given problem, planning and executing algorithm to solve problem and modification. Therefore it is meaningful task how conservation, proportional reasoning, probabilistic reasoning, controlling variables, correlational reasoning and combinatorial logic etc that are sub-factors of logical thinking. Also verified examination tool is necessary.

2.3 Logical thinking verification tool

GALT(Group Assessment of Logical Thinking) is logical thinking verification questionnaire that is jointly developed by Vantipa Roadranka, Russell H. Yeany and Michael J. Padilla in 1983[1][10]. There are total 21 questions and each questions multiple choices to choose answer and reason that is composed to measure 6 logical types. Sub-factors of logical type are conservation, proportional reasoning, controlling variables, probabilistic reasoning, correlational reasoning, and combinatorial logic and they stand for following abilities.

conservation

It is ability to understand that original quantity does not change even the shape changes.

• proportional reasoning

By realizing equal proportions of two quantity and logic to understand and solve quantitative relations.

• controlling variables

Realize all the variables in given condition, suppose hypothesis for the role of variables and systematically control variables to verify the hypothesis to derive conclusion.

• probabilistic reasoning

It is defined as ratio of expected probability for all the possible probability.

• correlational reasoning

It is ability to realize relationship between variables even the changes of object and phenomena are irregular.

• combinatorial logic

It is logic to count all the possible cases for solving problem without duplication. There are 4 Conservation questions, 6 Proportional reasoning questions, 4 Controlling variables questions, 2 Probabilistic reasoning questions, 2 Correlational reasoning questions and 3 Combinatorial logic questions. Most of the questions include short texts to explain the questions as well as pictures.

The feasibility of this questionnaire is shown in correlation with Semi-Clinic Interview by Piaget and Combinatorial logic shows the highest as 0.88, Correlational reasoning shows the lowest as 0.45. The overall feasibility coefficient is 0.71.

Reliability for 628 candidates from junior high school students to graduate school students showed 0.85 for Cronbach's Alpha coefficient and reliability of sub-reasoning is 0.37-0.83. The difficulty of questions is 0.02-0.78 (Average 0.40). Correlational reasoning (Average 0.11) and Proportional reasoning (Average 0.16) were the most and Conservation (Average 0.63) showed easiest. [10].

Distinction of the questions is 0.29-0.52 and as only 3 questions showed 0.29, most questions are well distinguished for high level recognition group and low recognition group.

According to final score of GALT, developers classified recognition level. Total score 0-8 is Concrete Operational Stage, score 9-15 is transition period and score 16-21 is formal operational stage.

3 Educational programming language 'Dolittle'

The educational programming language used in experimental class for this study is 'Dolittle'. 'Dolittle' is a language based on object-oriented way of thinking developed by Kanemune in 2000[7]. It is developed for school educational purpose for K-12 and currently it is available in Korean, Japanese and English programming.

3.1 Design Philosophy

- Simple Language Layered language is avoided and simple easy language is used. Instead of using input structure more than 2 levels; inner actions are distinguished as separate method to limit input level.
- Incremental Programming

Actions are accomplished when program in input with rational length sentence unit and it maintains interest of learners during the observation. It is contrary concept from existing class type object oriented language. • Description based on text

The basic principle of programming language 'As a artificial language, interpret and execute program by previously defined rules' is well maintained to write source codes based on text expression rather than figure operation.

• Algorithm and structure

Structured algorithm description is available by basic data of numerical value, string and arrangement and equipped with control structure such as repeat and conditional divergence to define methods that is the procedure enabling experience of basic computer concept.

• Object oriented

By using previously prepared devices, it provides completion of highly functional software in short term to be suitable in limited class of elementary, junior high, or high school students for their accomplishment.

• Prototype pattern

Most of object oriented languages are class type but it is hard for elementary, junior high, or high school students to understand class, instance or factorial. It is suitable for educational purpose with the common sense paradigm that if it is prototype pattern, copy of certain object shall take over property of original object.

3.2 Example and explanation of program

As an object oriented programming language 'Dolittle' has basic function of message transfer to object. Left of '!' is receiver and right is name of message.

BigTurtle = turtle ! create.

'turtle' is name of global variable that is property of top level object and previously prepared turtle object is applied. 'create' method creates new object and the prototype is turtle object. 'BigTurtle' is newly made global variable name and newly made object is inserted. The variable in 'Dolittle' is made when initial value is applied and it is not necessary to declare in advance. When this program is executed copied turtle object that is facing direction of X axis is indicated on the center of the screen.

Quadrangle = Bigturtle ! 50 forward 90 rightturn 100 forward 90 reftturn 50 forward closepath makefigure.

As message is returned in object value, message of return object can be constantly defined by writing name of factor and method. This is called 'Gasket(Serial)' transfer.

Quadrangle ! (blue) paint.

Numeric starting Token is value literal.

Bigturtle : DrawQuadrangle = [|x y| ! (x) forward 90 leftturn (2 * x) forward 90 leftturn (x) forward closepath (y) paint].

Method is defined as inserting property block of method. Block is indicated with [...] or [...] and inserted code row is executed when block is evaluated afterward.

[x > y] ! then [...] else [...] execute.

Logic operator withdraws logic value object. On above example, send then to block and execute block. Lastly return true or false for examination of evaluation value.

3.3 Experimental classes example by using 'Dolittle'

In order to verify effectiveness of educational programming language for K-12, our researchers have performed various experimental classes using 'Dolittle'. Followings are representing research cases and simple research results.

For 5 academically underachieving students, 5 hours of programming classes were performed[4]. As a result of experimental classes programming classes were successful toward academically underachieving students and they were able to accomplish reuse of object or overriding.

For 20 male and female 8th grade students 14 times of programming classes were performed[6]. Students who were interested in programming classes, they improved to intensive course in robot education.

For 19 10th grade students, 23 times of algorithm design and 15 times of program composition classes were performed[15]. On the process of completing object program to escape maze, it is determined that schema for logical thinking and problem solving was created.

For 70 high school students, 2 times of digital logic circuit composing programming classes were performed[12]. During the programming composing process of digital logic circuit, students were able to remember for a long time.

For 60 vocational high school students, 10 hours of 'Dolittle' classes and 20 hours of 'Visualbasic' classes were performed[14]. As a result of experimental classes, group who took experimental classes showed higher academic achievement and satisfaction toward educational procedures.

For 50 college freshmen and sophomore students who did not majored computer science, 6 hours of programming classes that object concept applying storymaking technique is applied were performed[8]. It is determined that 'Dolittle' is significantly effective language for students who did not majored computer science and 'storymaking approach' effectively

1790-1979

provides motivation and enhances academic achievements to students.

As results of research conducted above, short term (approximate 10 hours) programming education toward K-12 is established and students understood computer operation principle during completion of program with interest. This indicates that experimental classes using educational program language 'Dolittle' can be established without burden. Therefore 'Dolittle' is selected for experimental class programming language for enhancing logical thinking that is necessary for problem solving in this study.

4 Research Method

4.1 Objects for experimental class

Experimental objects for this study are total 265 9th grade students in 10 classes composed of 128 male students and 137 female students. For survey conducted previous to the classes, only 3 students answered 1 month of programming experience or programming learning experience.

Table 1: 'Dolittle' Programming education contents

Major academic contents and activities
Learning program operation method and function
Drawing simple figures by using basic grammer
Writing program by using repetition instruction
Understanding object and creating new object
Operation of object by using button
Figure object operation with timer method
Combination of timer and other commands
Creating new command
Operation of object by using collision method
Writing program by using parameter
Writing program by using function
Using field object and writing program
Using array object method and writing program
Writing program by using inheritance
Evaluating program writing ability
Writing simple application software

4.2 Contents of experimental classes and evaluation process

First semester of 9th grade academic curriculum is mostly 23 weeks. But actual class time excluding evaluation period and various student activities is 16 weeks. For total 16 weeks, 16 classes were performed and the class lasted 45 minutes each. 10 minutes were spent in order to deliver details of academic contents and 35 minutes were spent for students to practice examples.

Table 1 shows 'Dolittle' programming education contents for 16 weeks. The composition of academic program is restructuring of previously applied experimental classes into 16 courses that are mentioned above. It is actual contents applied to K-12, college students and adult female.

Beginning of the experimental class is focused on familiarizing program functions and understanding concept of the object. Middle part of experimental class writing program with combination of command 'timer' and 'collision' for making animation in order to maintain interest of students. Last parts of experimental class is writing advanced program by using concepts such as 'Array' or 'Inheritance' and it includes making problematic situation by the students to resolve the problems.



Figure 1: Execute Screen of Dolitte

The figure 1 shows example of program execution completed by class participant to make simple animation by using timer method after learning collision method and field object. Students were highly interested in making animation with program and despite high burden of recognition there were no surrendering students. Figure 2 shows source code screen in Korean.

Evaluation in order to study relationship between programming education and logical thinking, following steps are conducted. (1) Testing of logical thinking before experimental class

(2) Testing of logical thinking after completion of experimental class

(3) 3 time of program writing ability evaluation on the end of experimental class Academic achievement evaluation



Figure 2: Source Code of Sample Program

Testing of logical thinking before the experimental class was taken once on March of 2007, testing of logical thinking after the experimental class was taken once on July of 2007 and same testing questionnaire is used.

Evaluation of experimental class was divided into non-public logical thinking score evaluation for research purpose and public academic achievement evaluation notified to student and parents. Academic achievement evaluation score is opened as final term examination grade. Lastly, after the education, questionnaire was conducted to gather opinions of the students.

5 Research result

5.1 Change of logical thinking

In order to analyze difference between before and after 'Dolittle' programming education, first, the result of logical testing results for before and after experimental class were analyzed. Analysis tool was spss v.12. Second, correlational coefficient was used to determine correlation between 'Dolittle' programming education and logical thinking. Lastly, regression analysis was used to determine influence of 'Dolittle' programming education to logical thinking. All the verification was conducted with significance level of p<0.05.

As shown on the table 2, the average of prelogical testing is 9.90 and post logical testing is 11.29.

Table 2: Analysis of Logical testing(GALT) results

	Pre-test		Post-test		t value n va	
	М	SD	М	SD	t-valu	ie p-valu
Overall	9.90	4.398	11.29	4.432	8.542	0.000
Conservation	2.79	.981	3.03	0.927	4.577	0.000
Proportional reasoning	3.01	2.008	3.55	1.981	6.483	0.000
Controlling variables	1.76	1.357	1.92	1.388	2.288	0.000
Probabilistic reasoning	0.55	0.542	0.66	0.505	3.031	0.000
Correlational reasoning	0.25	0.465	0.34	0.512	2.527	0.000
Combinatorial logic	1.56	1.184	1.80	0.940	3.051	0.000

It is statistically significant and therefore it can be determined that 'Dolittle' programming education enhanced logical thinking.

For more accurate analysis, the mean and standard deviation are studied for conservation, proportional reasoning, controlling variables, probabilistic reasoning, correlational reasoning, and combinatorial logic that are sub-variables for logical thinking.

On pre-testing and post-testing conservation has increased from 2.79 to 3.03, proportional reasoning has increased from 3.01 to 3.55, controlling variables has increased from 1.76 to 1.92, probabilistic reasoning has increased from 0.55 to 0.66, correlational reasoning has increased from 0.25 to 0.34 and combinatorial logic has increased from 1.56 to 1.80. This result is also statistically significant. Therefore subvariables of logical thinking are also enhanced after the programming classes.

5.2 Correlation analysis result for programming education and logical thinking

In order to study correlation for programming education and logical thinking, correlational analysis was performed. Table 3 shows correlation analysis result for programming education and logical thinking.

As result of the statistics, correlational coefficient is 0.532 and significant difference is 0.000. Therefore GLAT testing result after 'Dolittle' educational programming language classes shows logical thinking has certain correlations and regression analysis was performed.

Table 5: Influence of programming education toward

logical thinking

Table 3: Correlation for programming education andpost-test(GALT)

Pearson 'R	P-Value
0.532	0.000

5.3 Influence of programming education to logical thinking

An analysis was performed to determine influence of 'Dolittle' programming education to post GALT testing result. Table 4 is analysis result of ANOVA for influence of 'Dolittle' education toward GALT.

Table 4: Influence of programming education towardlogical thinking(ANOVA)

	df	SS	MS	F	p- value
Linear Regression	1	1468.255	1468.255	41.418	0.000
Residual	263	3716.786	14.132		

When influence of programming education toward logical thinking is analyzed according to table 5, as determination coefficient (R square) is 0.283, it is determined that programming education has 28% determinative in enhancing logical thinking. Analysis results of each sub-variable are shown below.

As determination coefficient (R square) of conservation is 0.136 programming education has about 14% determinative in enhancing conservation logical thinking, 18% determinative in enhancing proportional reasoning logical thinking, 19% determinative in enhancing controlling variables logical thinking, 10% determinative in enhancing probabilistic reasoning logical thinking, 2% determinative in enhancing correlational reasoning logical thinking and 14% determinative in enhancing combinatorial logic logical thinking. With these results it can be determined that programming education had the most determinative in controlling variables among sub-variables of logical thinking followed by proportional reasoning and least determinative in correlational reasoning.

5.4 Responds of students

After experimental classes of first semester, the responses of students were that even programming is

	R	R Square	Adjusted R Square
Overall	.532	.283	.280
Conservation	.369	.136	.133
Proportional reasoning	.424	.180	.177
Controlling variables	.438	.192	.188
Probabilistic reasoning	.314	.099	.095
Correlational reasoning	.128	.016	.013
Combinatorial logic	.368	.135	.132

difficult it is interesting and wanted to learn more. As a result of education attitude, interests were constantly maintained and it is surprising to find that there were no surrenders even the difficulty level raised. But one the stage of algorithm composition, there were significant amount of time consumed compared to language skill education.

6 Conclusion

This study is to determine influences of first semester programming classes in regular academic curriculum by using educational programming language 'Dolittle' toward logical thinking of 9th grade students. Also determination of which sub-variable has the most influence among logical thinking. Results of the research are as following.

First, there were differences on logical thinking for before and after the programming education. Especially logical thinking was enhanced in all the subvariables of logical thinking.

Second, when relationship between programming education and logical thinking is determined, there was relationship between programming education and influence on logical thinking.

Third, it is determined that 'Dolittle' programming education help in enhancing logical thinking of students. Especially there were significant differences on sub-variables. Controlling variables was mostly influenced and correlational reasoning was least influenced.

Fourth, students maintained constant interest in programming education.

Consequently it is determined that performing programming educations for 1 hour a week for 1 semester by using educational programming language had significant differences in enhancing logical thinking. This is significant result for the new Korean education curriculum to be changed in enhancing problem solving using computer and to make logical foundation for enhancing logical thinking in order to enhance problem solving ability that can be applied to school environment education.

References:

- [1] Betty L., and Bitner-Corvin, Is the galt a reliable instrument for measuring the logical thinking abilities of students in grades six through twelve?, *Annual Meeting of the National Association for Research in Science Teaching*, April 1988.
- [2] Nell Dale and John Lewis, Computer Science Illuminated, *Jones and Bartlett Publishers* 3rd edition, 2007.
- [3] H. Gorman, and L. Bourne, Learning to think by learning logo: Rule learning in third-grade computer programmers, *Bulletin of the Psychonomic Society* 21(3):165–67, 1983.
- [4] H. Kang, Y. C. Yeum, S. W. Yoo, and W. G. Lee, An education of the underachieved students using 'dolittle', *Proceeding of The 14th KACE Winter Conference* Cheju, Korea, January 2005.
- [5] J. Piaget, and B. Inhelder, The early growth of logic in the child: Classification and seriation, *Basic Books*, New York, 1958.
- [6] J. H. Kim, D. Y. Kwon, Y. C. Yeum, S. W. Yoo, and W. G. Lee, Improving learning motivation and academic achievement using educational robots in secondary school, *Proceeding of The 14th KACE Winter Conference*, Cheju, Korea, 2005.
- [7] S. Kanemune, T. Nakatani, R. Mitarai, S. Fukui, and Y. Kuno, Dolittle-experiences in teaching programming at k12 schools, *The Second International Conference on Creating, Connecting and Collaborating through Computing, IEEE*, 2004.
- [8] H. S. Kim, H. S. Jang, H. C. Lee, D. Y. Kwon, Y. C. Yeum, S. W. Yoo, H. C. Kim, and W. G Lee, Teaching programming to non-cs major students: Experiments with storymaking approach, *IPSJ Symposium Series Vol. 2005, No. 8*, Tokyo, Japan, 2005, pages 135–141.

- [9] George Polya, How to Solve It, *Princeton University Press*, 1945.
- [10] Yeany R.H., Padilla M. J., and Roadranka, V, The construction and validation of group assessment of logical thinking, *A paper presented at the annual meeting of the National Association For Research in Science Teaching*, Dallas, Texas, April 1983.
- [11] C. Sgouropoulou, I. Chalaris, and M. Fouka, Bridging Theory and Practice for Systematic Learning Design, 2005 WSEAS International Conference on ENGINEERING EDUCATION, Athens, Greece, 2005.
- [12] S. K. Yeo, D. Y. Kwon, Y. C. Yeum, S. W. Yoo, and W. G. Lee, Instructional design and development on 'digital logic circuit' using dolittle, *Proceeding of The 14th KACE Winter Conference*, Cheju, Korea, 2005.
- [13] C. C. Wang, R. J. Fang, K. I Su, and S. J. Lu, Social Interactions of the Collaborative Problem Solving in a Mobile Learning Environment, *Proceedings of the 6th WSEAS International Conference on Applied Computer Sciecne*, Hangzhou, China, 2007.
- [14] S. W. Yoo, K. A. Kim, Y. Kim, Y. C. Yeum, S. Kanemune, and W. G. Lee, Empirical study of educational programming language for k12: Between Dolittle and visual basic, *International Journal of Computer Science and Network Security* 6(6), 2006, pages 119–124.
- [15] S. W. Yoo, Y. C Yeum, K. Susumu, and W. G. Lee, Programming Learning Pattern of Novice Programmer in Self-Guided Learning, *IPSJ Symposium Series Vol.2006, No.8*, Tokyo, Japan, 2006.