Promotion of Educational Effectiveness by Translation-based Programming Language Learning Using Java and Swift

Abstract

More and more programming tools have been created to help people to learn new programming languages. Although the number of tools to support beginning learners has increased, none directly compare different languages. This paper proposes a translation-based programming learning method that supports programming language learning for beginners of a new language who are familiar with a different language by comparing the same code written in the two languages. This allows learners to discover commonalities and differences between the languages, understand the grammar rules, and successfully write programs in the new language. Our method is demonstrated using a web-based programming language translator that translates Java into Swift. We also conducted an experiment to evaluate the educational effectiveness of this method.

The results show that programming language translator not only helps learners to understand the grammar, but also helps them to initialize their coding skills than common programming language learning tools.

1. Introduction

With the souring population of programming learners, the number of programming language learning tools to help learners understand and use new programming languages has rapidly increased. Common programming learning tools include guidebooks, official documents, tutorial websites and online academies.

If a learner is familiar with a general text-based programming language, he or she already has Computational Thinking [4], which "involves solving problems, designing systems, and understanding human behavior by drawing on fundamental computer science." This kind of programmers already possess knowledge about the structure, algorithm, and Computer Thinking, which allowing them to solve a problem or a task. Thus, when they learning new programming languages, they tend to focus on the basic grammar of the new programming language.

For such learners, a comparison is an effective domain-general means of learning [11]. A comparison is intended to foster "structural alignment" of two objects, which consequently allows important common properties to be discovered. Since the process of aligning two representations can extract a common higher-order relational structure, a comparison can help extract abstract knowledge for a wide range of tasks. Because comparison processes can promote conceptual learning, it may also assist in learning a new programming language.

According to Henrik [11], if the same information of code written in two different representations (programming languages) is compared, then the relevant relations between the representations can be extracted, allowing crucial structural features to be understood. In other words, when learning a new language, learners can discover the commonalities and differences between two languages by comparing the same meaning of code written in two different languages to build knowledge of the grammar rules in the new language. Comparing a new language to a familiar language should increase learning productivity.

However, most of current learning tools only include explanations and examples of one specific programming language.

One solution to this problem is using some form of translation approach. This paper proposes translation-based programming language learning method which provide the translation between two languages to support mainstream learners learning new languages. As an implementation, an educational environment is designed as a web-based programming language learning application called Jasmin that provide translation support between two languages. In this research, we explore two ways confirm whether the understanding and coding skills of learners are enhanced by this environment: reading comprehension and writing codes. Reading comprehension means the knowledge of a piece of code can be understood and appropriately applied, whereas writing code means that coding in a new programming language can be constructed to solve a given task. Reading and writing methods are commonly used at almost all universities to assess students' knowledge levels.

Therefore, the research questions in this research are:

RQ1: Comparing to the common programming learning tools, does language translation support in

Jasmin promote the learners' educational effectiveness when learning new programming languages?

RQ2: Comparing to the common programming language tools, which part of translation support in Jasmin is more useful: initializing the knowledge or start programming in new languages?

The results show that this educational environment not only helps learners to understand grammar rules, but also helps them to initialize their coding skills. What is more, the translation support is more helpful in initializing coding skills than understanding grammar rules.

As a demonstration, we translate Java and Swift in Jasmin, which are the Android application developing language and the iOS application developing language, respectively. We select these languages due to the following reasons. First, mobile applications have become very popular, and many people are interested in learning to program using these languages. "Of the 19 million software developers in the world, 8.7 million are now writing apps targeted for mobile devices, according Evans Data's recently released Developer Population and Demographics Study [6]. This represents a doubling of the mobile developer population since 2010 and an increase of 700,000 in the last year [6]." The population of mobile-minded programmers is definitely increasing.

Second, because the Android application market and the iOS application market have different platforms (Google Play and Apple Store), each application must have two versions to satisfy more users. In many companies, Android developers are asked to migrate to iOS developers (and vice versa) to more effectively manage human resources. Therefore, if Java can be easily translated to Swift, this would help Android developers migrate to iOS developers more smoothly.

The paper is organized as follows. Related works are listed in the Section 2. Educational environment design and implementation with translation support described in Section 3. Section 4 demonstrates the experiment and list results. Section 5 discusses the results and Section 6 concludes the paper with the future work.

2. Related Work

Some previous works have studied how to handle the translation approach to help the students learning new languages. For example, Matsuzawa [3] proposed an educational environment that offers improvements over current tools in order to help programming learners migrate from a visual language to a text language using Block to Java as an example. They developed a system that translate bidirectional way between Block (the block language used here) and Java [3]. The results indicate that the environment could act as a scaffolding in the migration phase. This research succeeded in promoting learner's seamless migration from Block to Java.

Dann [2] also proposed the transfer Alice to Java approach to encourage students continue learning computing in computer science. They transferred the exact same Alice program example directly into Java program to mediate a transfer of concepts. As a result, transfer improved student achievement in learning Java [19].

Google company also put effort on programming language learning using translation. Google PencilCode [23] provide the translation between two visual based programming languages in bidirectional way. Another tool Google Blockly [12] could translate the block language to multiple languages (JavaScript, Python, PHP, Lua, Dart) in a unidirectional way to supporting the beginning learners migrate from visual based languages to text based languages.

What is more, programming language translator between text languages have also been grown fast. According to Garnelis [20], Transifex offered the rough translation between the two languages to localize the applications and access new markets [20].

Qiu [21] also developed programming language translator from C to C++, and Java to C++ for reusing the existing code to improve the quality of the software and productivity of software engineers.

Those previous researches and works are all indicate that the demand of translation support are exist and need to be satisfied. Also in the education, the translation between two languages truly help learners to migrate smoothly from one language to another language.

However, much of the current research for education is mainly focused on translate from visual based languages to text based languages. The main area of these research is introductory education for non-CS students. Researches on translation between text-based languages are more focusing on the business usage rather than education. Thus, the educational effectiveness of translation support between two text based languages have not verified.

The trend of programming language learning is learning text based languages. The mainstream learners are computer science students and professional programmers and developers.

Hence, in this research, we proposed translation-based programming language learning method which translate from a text language to another text language to support mainstream learners learning new languages. As an implementation, an educational environment is designed as a web-based programming language learning application. In this research, we explore two ways to confirm the educational effectiveness using translation support: reading comprehension and writing codes. The results show that the translation support between two text based programming languages, not only helps understanding new grammar rules, but also helps learners start coding in new languages. What is more, the promotion in initializing coding skills is higher than the promotion in understanding grammar rules.

3. Educational Environment Design and Implementation

The experiential learning [17] method was used as one of the concepts in Jasmin. Experiential learning [17] is the process of learning through experience, and is more specifically defined as "learning through reflection on doing. Knowledge and skills were stored in leaners experiences by doing the learning activity and reviewing that they have done [18].

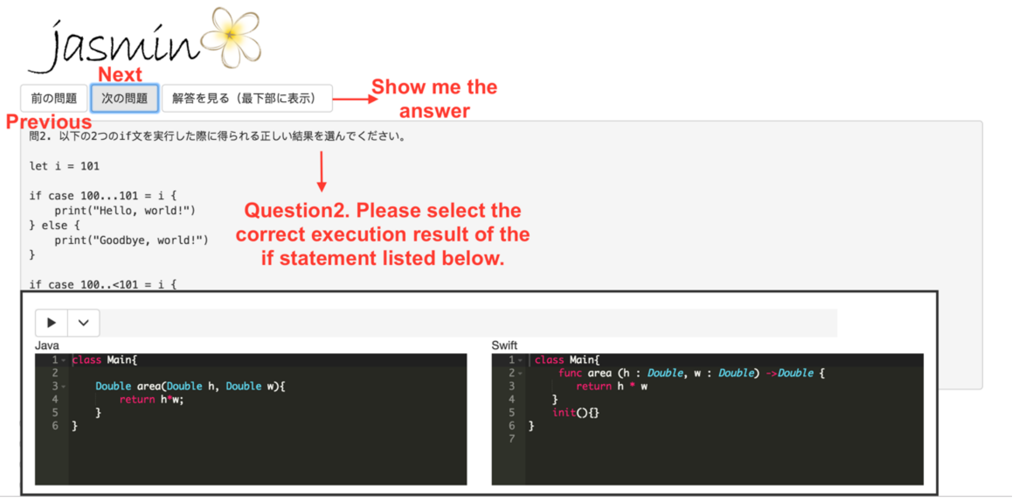
Provide the experience and let the children enjoy what they are doing is better than present the theories that have to be memorized and then the examples given, pointed out by Torin [22] from an interview about the Association of Waldorf Schools of North America. He identified that children could come to the ideas of some of the theories themselves by living into the experience. For example, children were allowed to go out into the yard and construct buildings and go field trips at the third grade. Gradually over time, the comparisons were occurred in the students' mind between different well in places, different people around the world. Then the ideas about how people live and thereby also how they think and how they related to on another were came up to children's mind.

Jasmin was designed to apply this kind of experimental learning method and the comparison to lead the learners to the commonalities and differences between two languages. Furthermore, help the learners to migrate smoothly from one language to another language.

Learning process using Jasmin was followed. Some reading comprehension questions and coding tasks were given to the learners to solve. Translation between Java and Swift was provided by Jasmin. Learners could use both Jasmin and other references and current tools to search for more detailed explanation of certain grammar. By solving those problems and tasks, learners could experience what does a Swift program look like and how should they coding in Swift. After all problems were solved, learners would have the basic knowledge of Swift language.

In order to provide experience about both understanding grammar rules and coding in Swift, Jasmin was divided into two parts.

The first part was shown in Figure 1. This part was consisted of three sections. The question section that address the reading comprehension questions about calculating the output of a piece of Swift code was displayed on the left. The translator section which translates Java program to Swift program was placed on the bottom. After learner press "show me the answer" button, the correct answer was shown in the answer section which shown in Figure 2. The Java program which translated from the piece of Swift code on the question section was also displayed with the answer. Java program was treated as an explanation of


jasmin
Next
前の問題
次の問題
解答を見る（最下部に表示）
Show me the answer
Previous
Question2. Please select the correct execution result of the if statement listed below.
Java
Swift

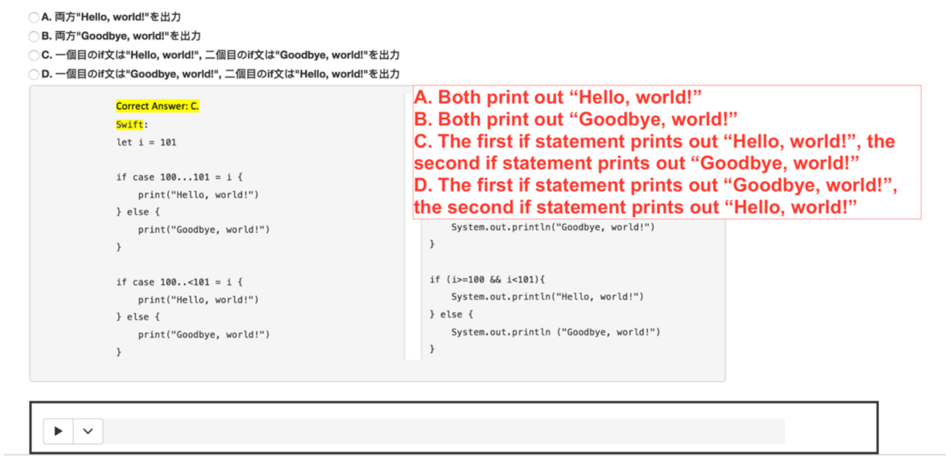
問2. 以下の2つのif文を実行した際に得られる正しい結果を選んでください。

```
let i = 101

if case 100...101 = i {
    print("Hello, world!")
} else {
    print("Goodbye, world!")
}

if case 100..<101 = i {
```

```
class Main{

    Double area(Double h, Double w){
        return h*w;
    }
}
```

```
class Main{
    func area (h : Double, w : Double) ->Double {
        return h * w
    }
    init(){}
}
```

Figure 1. Reading comprehension part of Jasmin


A. 両方"Hello, world!"を出力
B. 両方"Goodbye, world!"を出力
C. 一個目のif文は"Hello, world!", 二個目のif文は"Goodbye, world!"を出力
D. 一個目のif文は"Goodbye, world!", 二個目のif文は"Hello, world!"を出力

Correct Answer: C.

Swift:

```
let i = 101

if case 100...101 = i {
    print("Hello, world!")
} else {
    print("Goodbye, world!")
}

if case 100..<101 = i {
    print("Hello, world!")
} else {
    print("Goodbye, world!")
}
```

```
    System.out.println("Goodbye, world!")
}

if (i>=100 && i<101){
    System.out.println("Hello, world!")
} else {
    System.out.println ("Goodbye, world!")
}
```

A. Both print out “Hello, world!”
B. Both print out “Goodbye, world!”
C. The first if statement prints out “Hello, world!”, the second if statement prints out “Goodbye, world!”
D. The first if statement prints out “Goodbye, world!”, the second if statement prints out “Hello, world!”

Figure 2. Translation provided in the answer

Swift code. This could help the learners compare two languages and figure out the commonalities and differences between two languages. Time cost also could be saved by Java code. In the example shown in Figure 2, we expect learners could be aware of what does range operator (…) in Swift mean by only looking at the Java code instead of searching for the “Range operator in Swift” among the references.

Two editors were placed equally in the translator section. Java programs were written in the left editor for translating for the same example of Swift code. Target Swift code could be generated immediately by pressing the start button on the left top. This could help learners to save time for searching through the documents or tools for certain Swift grammar rules.

Several features were required to implement the translation.

F1. Programs written in different languages were provided at the same time.

F2. Statement sequences in both programs were the same.

F3. Programs in a new language were given immediately.

Obviously, all of those features were satisfied in the translation. For example, two programs were provided in two editors by the translator with the same sequences of statements. The exact same examples were shown in the answers section.

The inner process of implementing translator was shown in Figure 3. Translation was divided into three main steps. The first step is to scan the inputted Java program to build up the Java abstract syntax tree (AST). Then Java mapper was used to map Java AST to Unified code model. Finally, the Swift generator was used to traverses and modify the Unified code

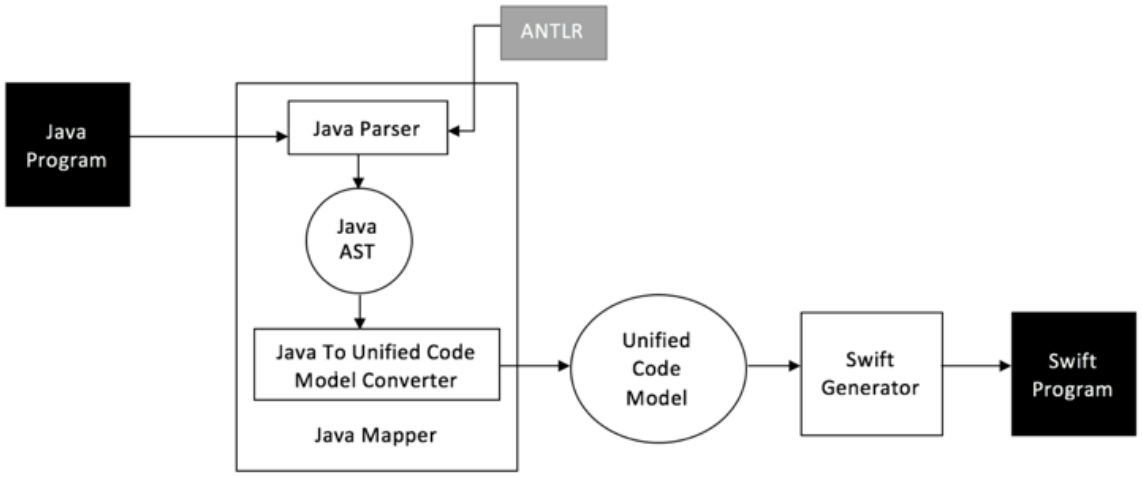


Figure 3. Inner process of translator

model to adjust to the Swift grammar and print out the target Swift code on the screen.

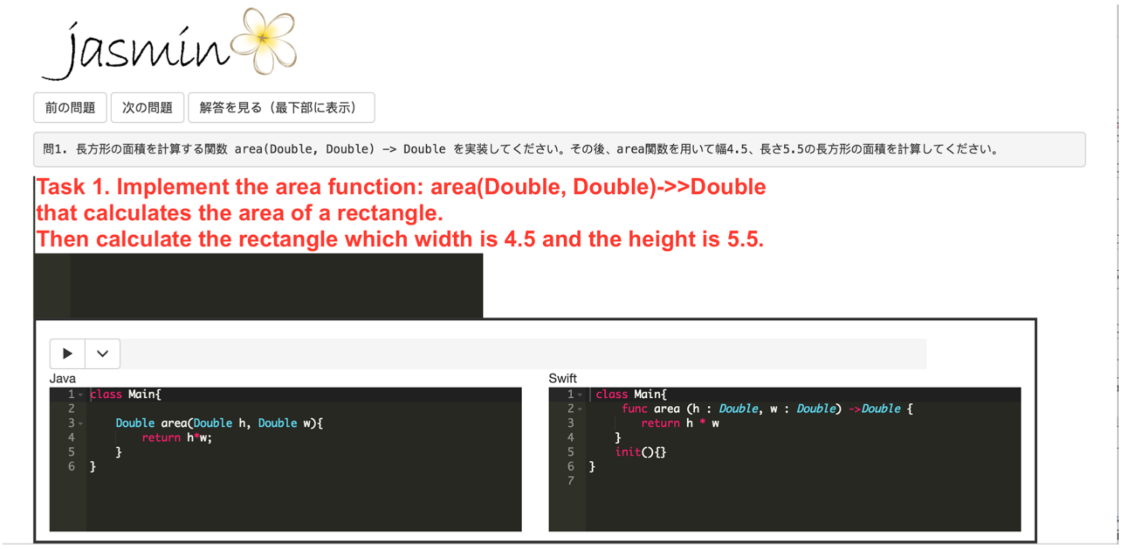
The Java Mapper and JUNICOEN were employed to generate a Unified code model from the Java programs [5]. Java Mapper has two parts: A Java parser and a Java-to-Unified-code-model converter. The Java parser is automatically generated by ANTLR (Another Tool for Language Recognition) [2]. The ANTLR as a parser generator not only recognizes languages but also interacts with a lexical analyzer (scanner), reports parsing errors, constructs abstract syntax trees, and calls user actions [2]. Using a few simple grammar annotations, ANTLR parsers can automatically construct abstract syntax trees (AST), preventing the user from explicitly calling tree constructor routines [2]. Thus, the inputted Java programs were scanned to build up Java abstract syntax tree(AST) by ANTLR [2].

JUNICOEN [5] was used in this research to build a Unified code model from Java AST. The Universe code model is a language independent AST which has its own universal nodes (Uni-Nodes) to build up AST [5]. The Unified code model was produced from Java AST by Java-to-Unified-code converter. Unified code model is treated as a middle representation during translation. The universe code model was enabled to extend to handle multiple programming languages in the future.

Then the Unified code model was traversed by Swift generator according to the Swift grammar to generate the target Swift program to display on the screen. The purpose of the Swift generator is to traverse the Unified code model and generate a translated, accurate, and runnable code to display on the screen. However, the grammar concepts of Java and Swift differ slightly in many places, including keywords, syntax, etc. For example, the constructor in Java is treated as a method that has the same name with the class name and has no return type. However, in Swift, the constructor has a specific name called "init". Sometimes only the keyword switching was needed to produce the code while some times the AST transformation was needed to avoid the mistakes when producing the Swift code. Deleting unnecessary keywords, elements as well as changing the necessary keyword and elements were included in AST transformation to adjust to the Swift grammar.

The second part was shown in Figure 4. The question section, answer section and the translator section were included in this part. The translator could be used to solve the tasks and the answer would be shown by pressing "show me the answer" button.

Since the purpose of this research is helping beginning learners to learn new programming languages more smoothly, only some basic grammars of new languages were handled in this learning environment including class declaration, field declaration, function declaration and recall, constructor, if and switch statements, optional value and if-let statement, range operator, closure, dictionary declaration, type cast, array declaration and access. These basic grammar rules were becoming to the start point of learning new languages. After obtaining the knowledge of those grammar rules, learners were able to understand and write more difficult Swift programs including UI managements.



問1. 長方形の面積を計算する関数 area(Double, Double) -> Double を実装してください。その後、area関数を用いて幅4.5、長さ5.5の長方形の面積を計算してください。

Task 1. Implement the area function: area(Double, Double)->>Double
that calculates the area of a rectangle.
Then calculate the rectangle which width is 4.5 and the height is 5.5.

Java
```java
class Main{

    Double area(Double h, Double w){
        return h*w;
    }
}
```

Swift
```swift
class Main{
    func area (h : Double, w : Double) ->Double {
        return h * w
    }
    init(){}
}
```

Figure 4. Coding task part of Jasmin

4. Experiment

4.1. Experimental Design

An experiment was conducted to evaluate educational effectiveness of translation-based programming learning method in terms of reading comprehension and coding skills. 12 subjects who have certain experience in programming were invited through Twitter to this experiment. Those subjects are consisted of students, programmers and other people from different jobs. Those subjects have certain experience of learning and using Java and no experience about Swift. They were randomly divided into A and B, two groups. The group A is the group without translation support when the group B is using the translation support. 7 subjects were assigned to the group A and 5 subjects were assigned to the group B.

At first, the same reading comprehension questions and coding tasks were distributed to all subjects. Jasmin with translation support was provided to one of the groups when the other group was not. Therefore, the other group was only received the correct answers of the questions and tasks without any translation between two languages.

Then the self-study section was conducted by every subjects. Both groups were asked to solve those questions and tasks during self-study section. Swift tutorial documents [15] from the Apple company was distributed to every subject as a standard reference. What is more, all kinds of published programming learning tools and documents could be used during self-study section to learn the basic grammar rules and knowledge of Swift language.

After the self-study section, the time cost in self-study section was recorded. Then a blind test was token to verify the learning performance of every subjects. Both reading comprehension questions and coding tasks were included in this test as the same form of the problems that distributed before self-study section. The questions and tasks are mainly from the "Test your Swift" website [10] and "Programming Language Swift Definition Guide 2nd Edition" book [9]. None of the references, Xcode application, online compiler, programming language learning tools were allowed during this test. The full score of both part is all up to 10.

Finally, the subjects were asked to answer a questionnaire. This questionnaire was designed to analyze subjects' attitude towards this experiment, Swift language and translation support. The group without translation support, also allowed to access to the Jasmin after the test, to experience the what does translation support do in learning process and how do they think about this kind of support.

Both of the language knowledge required in self-study section to solve the questions and tasks, and the knowledge tested in the blind test were illustrated in the table 1.

4.2. Experimental Results

Table 2 shows the measurements of the learning performance of the group without translation. The scores obtained by every subjects are rather different. Almost subjects gained better score in reading comprehension section. The total score in reading comprehension also higher than coding.

Table 3 shows the learning performance of the group with translation. Total 80% of the subjects in this group reached the highest score. Interestingly, the time cost in this group is significantly different among subjects. Subject B2 obtains the total highest score with only 32 minutes of time cost when subject B4 uses over 2 hours during the self-study section.

Table 2. Measurements of learning performance without translation support

	Subject A1	Subject A2	Subject A3	Subject A4	Subject A5	Subject A6	Subject A7
Study time (min)	102	34	79	57	139	19	43
Reading score	9	7	6	7	6	8	9
Programming score	8	3	8	3	6	6	4

Table 3. Measurements of learning performance using translation support

	Subject B1	Subject B2	Subject B3	Subject B4	Subject B5
Study time (min)	34	32	112	200	39
Reading score	8	9	9	8	9
Programming score	5	7	7	8	7

Table 1. Language knowledge covered in Jasmin and the experiment

	Language knowledge required to acquire in self-study section	Language knowledge tested in reading comprehension and coding tasks
Array access	✔	✔
Array declaration	✔	✔
Basic data types(Int, Double, Float, String, Character, Boolean)	✔	✔
Class declaration	✔	✔
Closure	✔	✔
Constructor	✔	
Dictionary declaration	✔	✔
Dictionary count property	✔	
Enum declaration		✔
Enum element access		✔
Field declaration	✔	✔
For loop	✔	✔
Function declaration	✔	✔
Function recall	✔	✔
If statement	✔	✔
If-case statement	✔	
If-let statement	✔	✔
Nested functions	✔	✔
Object creation	✔	✔
Optional value	✔	✔
Pattern matching operator	✔	
Range operator	✔	✔
Return statement	✔	✔
String library functions	✔	✔
Switch statement	✔	✔
Ternary operator	✔	
Type cast	✔	✔
Variable, constant declaration	✔	✔

Figure 4, 5 and 6 shows the comparison of each part between two groups. Left side is representing group A when the other one represents the group B. The group A tend to have lower score than group B in both reading comprehension and coding tasks. The minimum score, medium score and the highest score of both part in group A are lower than group B. In contrary, group A has lower time cost than group B in self-study section.

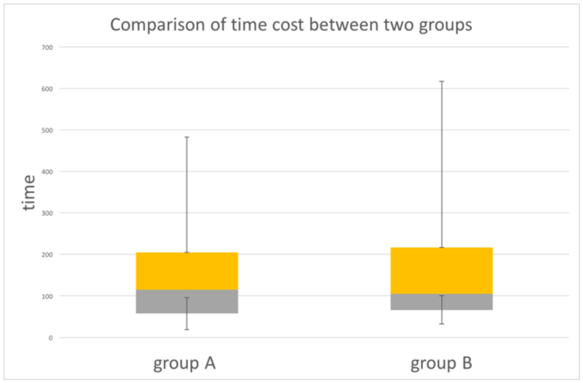


Figure 4. Comparison of time cost between two groups

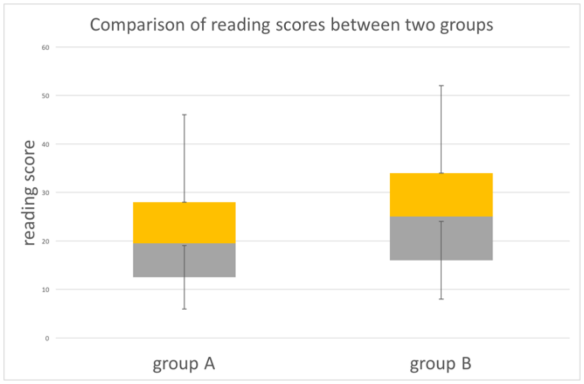


Figure 5. Comparison of reading scores between two groups

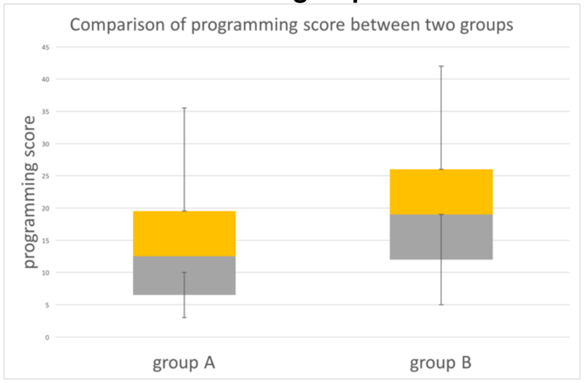


Figure 6. Comparison of programming scores between two groups

Table 4 compares the average of learning performance between two groups. Although the group B costs about 16 minutes more in self-study section than group A, the subjects in group B have significantly higher accuracy percentage in both reading comprehension questions and coding tasks. The difference in grammar understanding between two groups is 11.7%, and in coding skills, it becomes 13.7%.

Table 4. Average of the study performance between two groups.

	Group A	Group B
Average study time(min)	67.6	83.4
Accuracy percentage in reading	74.3%	86.0%
Accuracy percentage in programming	54.3%	68.0%

Table 5 shows the results of the questionnaire after test. 91.7% of the subjects answering "yes" to the question that whether they compare the new language with familiar language when they learning, or the solution in familiar language occurred to their mind when they see the tasks. About 92% of subjects answered that the translation from the new language to familiar language would be useful in both understanding new grammar and initializing coding skills. Also 83.3% of subjects have positive attitude about that the translation from familiar language to new language would help both part when learning new languages. However, subjects considered that translation between two languages would help understanding new grammar rules rather than helping coding in new languages.

Table 5. Results of the questionnaire after test

		All Subjects
Translation existence		91.7%
From a new language to a familiar language	Help understanding the grammar rules	41.7%
	Help coding in new languages	16.7%
	Help both	33.3%
From a familiar language to a new language	Help understanding the grammar rules	58.3%
	Help coding in new languages	16.7%
	Help both	8.3%

5. Discussion

Table 2 and Table 3 suggests that coding in new languages is more difficult than understanding new grammar rules when learning new programming languages. The p value of the t-test is less than 0.01 which implies that learners often have difficult in coding than understanding new grammar rules of new languages.

According to the Table 4, accuracy percentage in both reading comprehension questions and coding tasks of group B are higher than group A. The p value of t-test in reading comprehension part is less than 0.05, which suggests the statistical difference between two groups. However, the p value in coding part is larger than 0.05 to indicate that there is no statistical difference between two groups.

Therefore, the answer to the first research question is, positive. The translation support in Jasmin does promote the learners' educational effectiveness when learning new programming languages. Also the answer to the second research question is, the translation support in Jasmin promotes initializing the knowledge part more than coding parts. The differences of programming experience and levels among subjects may lead to the conclusion that translation support could not help learners initializing coding skills of new languages.

Table 5 reveals that the comparison is always exist in learning process. Learners always compares new languages with the languages they familiar with to discover the commonalities and differences. This could help them to understand and discover the characteristic of new languages better and extend their way of thinking among diverse programming languages.

6. Conclusion and Future Work

In order to support learners to learn new programming languages, this paper proposed translation-based programming language method. As a demonstration, the learning environment with translation support called Jasmin was implemented. We conducted the experiment to evaluate the learning performance of using Jasmin.

In conclusion, using programming language learning environment with translation support can aid beginning learners in gaining knowledge about a new programming language and initializing the coding skills. Also, it is more effective in promoting learning performance in initializing the coding skills than understanding new grammar rules of new programming languages.

After the experiment, some comments about this experiment and translation support were sent to us. One of the subjects think it is good for the beginners to learn new languages. With the translation support, the beginners could start programming in new languages rather than getting confused in the start point. Some of subjects think it is interesting comparison between the specific grammar which Swift have and Java do not. They expecting this kind of translation support could promote the learning performance in the other programming languages. One of the comments is about asking us to extend current tool to cover more programming languages such as C++ or C#.

Another comments point out that, since the learners could get the Swift code from the translator without bugs, they could concentrate more on learning new languages rather than fixing bugs. This helped them to keep up the motivation learning Swift.

However, few subjects admitted that they get confused when they suddenly given the questions at very first time without any grammar explanation. They confessed it would be more appreciated that if there are some literal explanations about Swift grammars. Since Jasmin only provide the translation support, learners have to search for more detailed explanation for certain grammar rules which might reduced the motivation of learning.

In this research, we use Jasmin with translation support to measure the educational effectiveness among learners.

Several directions are considered as the future work. We plan to consider the feedbacks and comments and combine tutorial documents to Jasmin to provide the full support in learning new languages. Also, Jasmin could be extended to cover more programming languages.

What is more, the investigation of the relationships between language similarity and educational effectiveness also could be considered as a future work. In this research, we use Java and Swift which are similar languages to measure the educational effectiveness among students. However, the learning effectiveness of using translator might change due to low percentage of similarity between two languages.

7. References

[1] Y.Matsuzawa, K.Sakamoto, T.Ohata, K.Kakehi , "Programming Language Translation System for Programming Education(In Japanese)", IPSJSIG-CE, August 2015, pp. 223-230.

[2] T.J.PARR, R.W.QUONG, "ANTLR: A Predicated-LL(k) Parser Generator", SOFTWARE-PRACTICE AND EXPERIENCE, VOL.25(7), JULY 1995, 789-810.

[3] Y.Matsuzawa, T.Ohata, M.Sugiura, S.Sakai, "Language Migration in non-CS Introductory Programming through Mutual Language Translation Environment", SIGCSE2015'Proceedings of the 46th ACM Technical Symposium on Computer Science Education, February 2015, Pages 185-190.

[4] Wing, J., Computational Thinking, Communications of the ACM, Vol. 49, No.3, 2006, pp. 33-35.

[5] K.Sakamoto, A.Ohashi, D.Ota, H.Washizaki, Y.Fukazawa, "UNICOEN : Framework that analyzes and transforms source code supporting multiple programming languages.(In Japanese)", IPSJ, Vol.54, No.2, (2013), pp. 945-960.

[6] Marthe Rana, "Mobile Developer Population Reaches 8.7M Worldwide, New Evans Data Developer Population and Demographics Study", available from <http://www.evansdata.com/press/viewRelease.php?pressID=210> (2016.05.30).

[7] Yang, T.-C., Hwang, G.-J., Yang, S. J. H., & Hwang, G.-H. "A Two-Tier Test-based Approach to Improving Students' Computer-Programming Skills in a Web-Based Learning Environment". Educational Technology & Society, 18 (1), (2015), 198–210.

[8] Batia LAUFER, Haifa, Israel, “Electronic dictionaries and incidental vocabulary acquisition: does technology make a difference?”, ELECTRONIC DICTIONARIES IN SECOND LANGUAGE COMPREHENSION.

[9] Takeshi Ogihara, “Programming Language Swift Definition Guide 2nd Edition”, 2015/12/25.

[10] Test Your Swift, available from <https://www.hackingwithswift.com/test/> (2016.05.30) .

[11] Henrik Saalbach, Lennart Schalk, Michael Schneider, Elsbeth Stern, “Learning through Comparison”, available from <http://www.ifvll.ethz.ch/research/Comparison> (2016.05.30).

[12] Google Blockly abaliable from <https://developers.google.com/blockly/ > (2016.05.30).

[13] Swift quiz, available from < http://www.ios-blog.co.uk/quiz/> (2016.09.11)

[14] Humaira, R.; Sakamoto, K.; Ohashi, A.; Washizaki, H. & Fukazawa, Y., “Towards a Unified Source Code Measurement Framework Supporting Multiple Programming Languages”, Proceedings of the 24th International Conference on Software Engineering and Knowledge Engineering, Knowledge Systems Institute Graduate School, 2012, 480-485.

[15] The Swift Programming Language (Swift 2.2) from iOS Developer Library available from <http://www.studiogalago.com/the-swift-programming-language/ >(2016.05.30).

[16] The Java™ Tutorials from ORACLE Java Documentation available from <https://docs.oracle.com/javase/tutorial/java/index.html> (2016.05.30).

[17] Felicia, Patrick (2011). Handbook of Research on Improving Learning and Motivation. p. 1003. ISBN 1609604962.

[18] Experiential learning, Dick, Bob (2002) The design of experiential learning activities. Unpublished paper (mimeo).

[19]W. Dann, D. Cosgrove, D. Slater, D. Culyba, and S. Cooper. Mediated transfer, “Alice 3 to java”. In Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, SIGCSE ’12, pages 141–146, 2012.

[20]Antonis Garnelis, “Translation between programming languages” [Online Document], cited August 2, 2016. Available HTTP: https://www.transifex.com/blog/2012/translating-between-programming-langs/ (2016.09.01).

[21] Lili Qiu, Programming Language Translation, Department of Computer Science Cornell University Ithaca, NY 14853

[22] Waldorf School of New Orleans, Experiential Learning and Waldorf Education, available from < https://www.youtube.com/watch?v=mSeyZFaGKIY> (2016.09.01)

[23] Google pencil, available from <https://pencilcode.net/ > (2016.09.01)