LEARNING OBJECT TO ENHANCE INTRODUCTORY PROGRAMMING UNDERSTANDING: DOES THE SIZE REALLY MATTER?

Reginamary Matthews
Faculty of Engineering
The University of Nottingham, Malaysia Campus, Semenyih
Selangor, Malaysia
reginamary.matthews@nottingham.edu.my

Hew Soon Hin and Koo Ah Choo
Faculty of Creative Multimedia
Multimedia University, Cyberjaya
Selangor, Malaysia
shhew@mmu.edu.my ackoo@mmu.edu.my

ABSTRACT

Learning Object (LO) is the breakdown of larger content into smaller pieces of information that accomplishes a single learning outcome. The smaller piece of content is incorporated with multimedia elements to promote meaningful learning. The prevailing focus on learning objects for introductory programming promising in terms of enhancing the programming concepts and syntax learning. However, the size of learning object that determines the amount of information to be placed on a single learning object still debatable. There are no prominent guidelines found in literature to assist the design the size of the learning object for introductory programming modules. This study aimed to investigate the effects of different sized learning objects on programming learning. An experiment was carried out and one hundred and one novice programming students participated in this study. Two different sizes of LO was developed as a learning support tool in a lab setting. The results of this study found that smaller learning objects (named as Micro Learning Object) are useful in delivering programming knowledge and actively engaged students in learning the programming concepts and syntax.

Keywords: Learning objects, granularity, programming ability

INTRODUCTION

The design and development of educational material has changed dramatically from being a course to smaller pieces of content. The decomposition of contents into meaningful smaller pieces of content is known as learning object (Wiley, 2000; Allen and Mugisa, 2010**)**. Literally, the term Learning Object (LO) was coined by Wayne Hodgins when he named one of his working groups and eventually it was used widely in the computer mediated learning field and content creation (Polsani, 2003). Hodgins conceived the idea of learning objects when he watched his children playing LEGO ^{TM}, apparently, both of them, who had different learning preferences, met their needs equally when playing with the blocks (Hodgins, 2004). This term also has been popular in the E-Learning field (Seung, 2007).

Higher educational institutions benefit from learning objects because it is able to: (a) develop and deploy learning content efficiently and quickly (b) deliver content between LMS (Learning Management System) and LCMS (Learning Content and Management System) or other E-learning platforms and (c) reduce content development, maintenance time and delivery costs (Learning Circuits, 2005). Learning objects for programming have been widely used in several universities and in high schools across the world. Studies have reported learning objects have been designed for programming languages in higher education institutions, such as Java and C++, which helped students to learn better and understand the abstract concepts of programming (London Metropolitan, 2004; Tempere University of Technology, 2012).

Nokelainen (2006) and Waston (2010) pointed out studies on LO related to pedagogical aspects are more scant than technical ones. Attention on how LO would be meaningful for novices in the learning context seems limited in existing literature. The granularity of learning objects is seen as a pedagogical issue. The size of learning objects is unclear and there is uncertainty in determining how much amount of content is just right to facilitate the process of learning. Yau (2004) studied learning objects for Java programming and reported the size of learning objects does not make it clear in determining how much content a single learning object should contain. The other issue with learning object is the reusability that is associated with the size of learning object. Wiley (2000) highlighted that the degree of reusability is high when the size of learning object is small. When LO is highly reusable it is expected to be reused in different learning context. However, when several types of learning objects are aggregated, the degree of reusability is relatively low. The purpose of this study is to explore the effect of different sizes of learning objects used as a support tool in a programming lab. The following questions are addressed in this study:

1. Is there any difference between sizes of LO in improving the programming knowledge?
2. How does the size of LO facilitate or hinder programming learning?
3. Would it be useful to integrate different sizes of LO for programming learning?

REVIEW OF LITERATURE

The idea of breaking content into smaller chunks of information is a new approach to content creation, which is just right and uses the lowest possible size to accomplish a single learning outcome. The chunk of content attributed to be highly reusable in various learning contexts. Some literature refer to learning objects as reusable learning object (Jacobsen, 2002; Sicilia and Garcia, 2003; Seung, 2007). Although, it has emerged with different name, the concept of reusing the chunks is the key concern. Balatsoukas, Morris, and O'Brien (2008) reviewed several learning objects' frameworks that define the structure and aggregation of learning objects. The study outlined the existence of ambiguity in terms of granularity. Even though learning object appears with different nomenclature, the ultimate goal is relatively similar. Substantial amount of literatures seem to point out the same underlying characteristics for learning objects. The main characteristics are reusability, granularity (size/level) and self-contained (independent) and aggregation (assembled into larger collection) (Wiley, 2000; Bergtrom, 2006; Beck 2010).

Literally, the smallest unit of information known as raw media, element or asset and includes images, video clips, audio clips, animation, photographs, java applets, tables, guidelines, and examples, summaries and so forth. The raw data is aggregated to form information units that represent various types of concepts, facts, procedures, processes, or principles. Several units of information are then aggregated to form learning objects that are built to carry a single learning outcome. At this point, the objects are highly reusable in different leaning context. A learning object can be integrated for a single lesson unit or several independent learning objects can be integrated for a single lesson which might carry various skills or content. The degree of reusability drops when several learning objects are sequenced to form a learning component such as a course. The highest level combines several learning components for a collection of courses (Advanced Distributed Learning, 2012; IEEE, 2002). The learning objects' level of aggregation as illustrated by Krull and Mallison (2004) based on Hodgins's preliminary ideas, depicted in Figure 1.

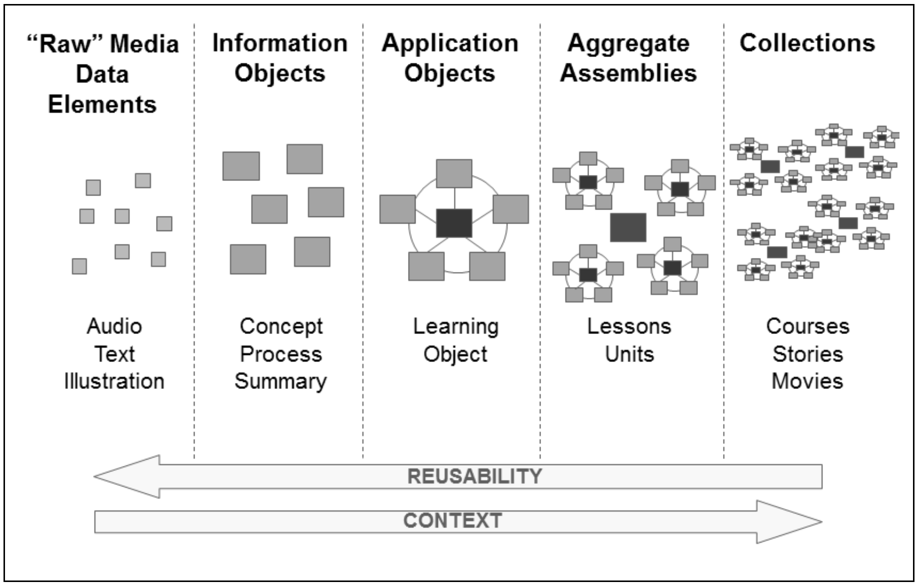


Figure 1: Modular content hierarchy

METHODS

PARTICIPANTS

One hundred and one engineering foundation students enrolled in Introduction to C programming module took part in this study. Students are randomly selected and assigned into two groups. The first group consisted of fifty students and the latter with fifty one. Most of the students are with little or without prior programming knowledge.

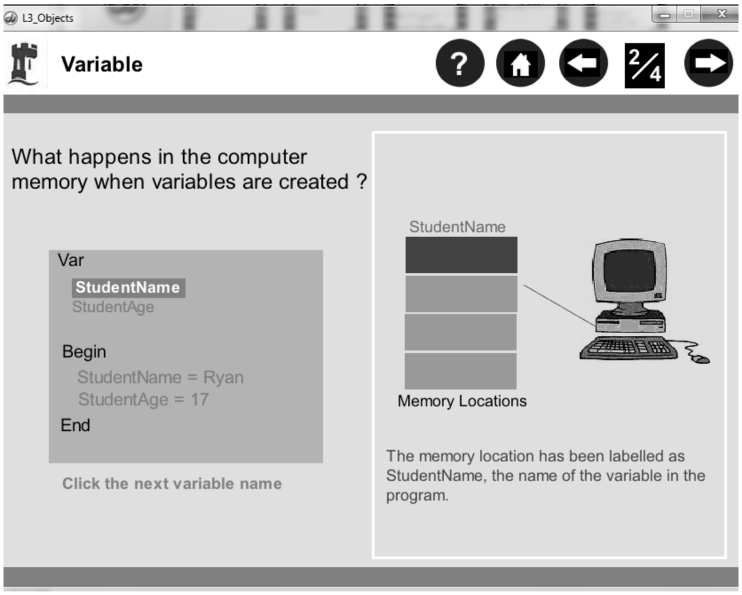
INSTRUMENTS

Two types of learning objects are designed and developed for C programming. LO is not merely presentation of information, it engages students and they must interact in order to learn. Fetaji, M, Loskovska, Fetaji, B, and Ebibi (2007) pointed out that programming knowledge cannot be transferred from instructor to learner, therefore, the responsibility of learning needs to be shifted to students. These LO intended to support the cognitive and learning process. Each LO designed to accomplish a single learning outcome. The size of learning object is determined by the number of pages, access time and logical content. Smaller Learning Objects (Micro LO) vary from 5 to 15 minutes whereas the larger LO that is aggregated with several LO (Macro LO) varies from 20 to 30 minutes. To ease reporting, the micro learning objects are named as Content Object (CO) and Self-assessment Object (SO) (see Table 1). Two cognitive learning approaches (learn and practice) identified to be used in the programming lab with the use of CO and SO respectively.

Table 1: Size of learning objects

Micro LO	Size	Macro LO	Size
Content object (CO)	5 - 10 minutes	Main Page + Help Page + Content object (CO) + Self-assessment Object (SO)	20 - 30 minutes
Self-assessment object (SO)	5 - 10 minutes		

CO are designed to aid the understanding of abstract programming concepts in C programming (i.e., what is a compiler? What happens in the computer's memory when a variable is created? How a nested selection structure works? and so forth). Visuals and animation are used to explain the concept in order to ease the understanding. Figure 2 shows an example of CO created for the concept of computer variable. The pseudocode shows as assignment of string values in a variable called StudentName and an integer value in another variable is named as StudentAge. The animation on how the memory locations created for the given variables are played as the student clicks pseudocode line by line. When the last line is reached, the student will not be able to click the line of codes randomly. This is to show that the complier executes the program codes in a linear form and also to give the novice learner an understanding on how the program is executed. Animated text for each line of codes is used to enhance the understanding.



What happens in the computer memory when variables are created ?

```
Var
   StudentName
   StudentAge

Begin
   StudentName = Ryan
   StudentAge = 17
End
```

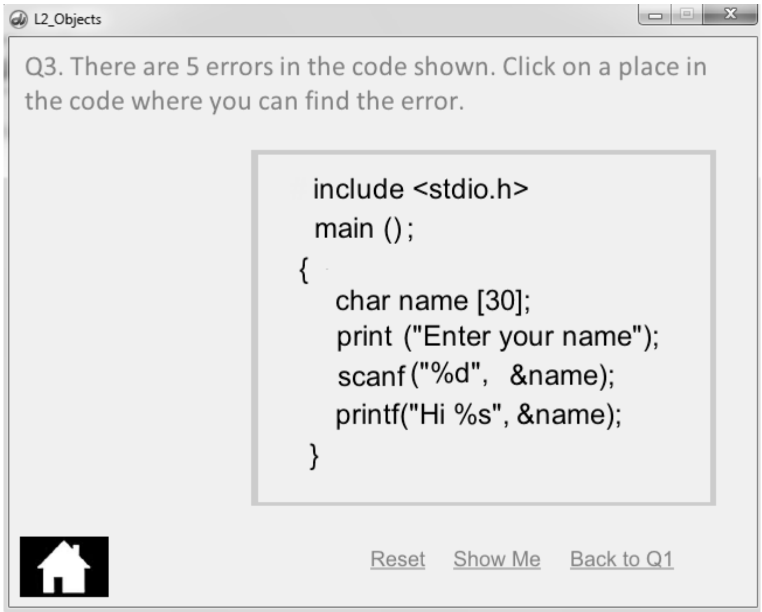
Click the next variable name

The memory location has been labelled as StudentName, the name of the variable in the program.

Figure 2: An example of Content Object for computer variable

SO is designed to help students to understand the programming syntax and codes. Students are exposed to SO after they have used the CO. The aim is to match the concept learnt in CO. As the CO enhances the understanding of the concept of programming, the SO helps students to learn programming syntax and codes.

Figure 3 shows an example of SO which is coherent with the programming concepts learnt in CO. Novice learners have difficulty in remembering the codes (Matthews, Hew, and Harprith, 2008) and this SO is aimed at helping them to recall the syntax required to write a program. An immediate feedback is displayed when students make mistakes. The feedback is the most important aspect in the SO that helps students avoid misconception. Knowing the common mistakes made when writing a program is one of the learning approaches used to aid the understanding. It is also helps students figure out the syntax error and the meaning of the error. Tracing the output and matching the correct syntax are the other type of questions used in SO.

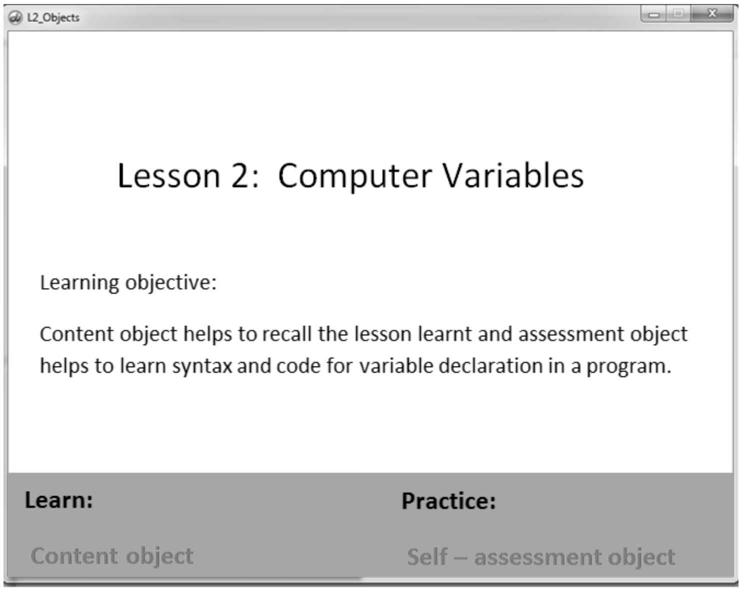


Q3. There are 5 errors in the code shown. Click on a place in the code where you can find the error.

```
 include <stdio.h>
 main ();
{
   char name [30];
   print ("Enter your name");
   scanf ("%d",  &name);
   printf("Hi %s", &name);
 }
```

Figure 3: An example of Self-assessment Object for computer variable

Macro LO is packed with learning objects designed in a sequence following learning activities. It consists of a main page (see Figure 4) and a help page and navigations icons to switch back and forth when accessing the learning objects. Each Macro LO is accompanied with a learning objective on the home page to ease the understanding of when to use the LO and its aim. Students have the freedom of choice to select the type of LO they prefer to use after they have learnt the programming concept and syntax.



Lesson 2: Computer Variables

Learning objective:

Content object helps to recall the lesson learnt and assessment object helps to learn syntax and code for variable declaration in a program.

Figure 4: An example of Macro Learning Object

A formative evaluation method (quiz) is used to explore the progress of students' understanding at the end of each lesson. The quiz consisted of five multiple choice questions, two or three which traces the output of a program questions while the last questions requires students to write programs. Data collected from quizzes explores how different sizes of LO helps or hinders the programming learning. In order to address research question 3, a survey form designed with 12 items. Table 2 shows how the survey items are divided into three subscales and codes used to ease reporting. Item codes ranging from C1 to C5 is to understand students' learning experience in using content object and S1 to S5 is for self-assessment object. A 5-point Likert scale is used for each item as follows: 1. Strongly agree; 2. Agree; 3. Uncertain; 4. Disagree; and 5. Strongly disagree. Codes ranging from R1 to R2 require students to rank the learning objects based on the following 5-point Likert scale. 1. Most useful; 2. Moderately useful; 3. Useful; 4. Least useful; and 5. Not useful at all.

Table 2: Coding for survey items

Code	Items
C1	Content object is useful to recall the programming concepts before I learn a new lesson
C2	Content object helps to relate the programming concepts that I learn in every lesson
C3	I always make sure I use the content object before the lecture is started.
C4	The time allotted to use the content object during the class time is just right.
C5	I would like to use content objects to review every lesson.
S1	The self-assessment object helps me to understand the lesson learnt.
S2	I always access the self-assessment object before I start the lab activities.
S3	I find the self-assessment object as an important learning activity.
S4	The self-assessment object helps me to reflect the programming concepts and syntax.
S5	The solution for the self-assessment object is useful.
R1	Rank the content object in terms of its usefulness in learning introductory programming concepts and C programming
R2	Rank the self-assessment object in terms of its usefulness in learning introductory programming concepts and C programming

PROCEDURE

This study carried out with a two group pre-test and post-test design. One hundred and one engineering foundation students in the final semester were required to take Introduction to C programming as a core module. Students were randomly assigned into two groups. One group was randomly selected to expose them to Micro LO and named as Micro LO group (n=50), and the latter is Macro LO group (n=51). The C programming class was conducted weekly for three hours, in a lab over eight teaching weeks. Table 3 shows the experimental procedure on how the programming classes were conducted. A pre-test was administered in the first week to investigate the difference in the level of prior knowledge between the groups. The Micro LO group used CO after the first hour of the lecture to recall the lesson learnt and SO as part of the practical programming activities, whereas Macro LO group had the freedom of choice to use the desired LO. However, as part of the class instruction students were asked to use the LO after the lecture and SO during the practical session. Similar procedures were used in both groups and the same instructor conducted programming lessons for both groups. At the end of every class, students were required to take a 20-minutes quiz related to the lesson learnt. The quizzes are numbered as Q1 to Q6 tailing the lesson L1 to L6. At the end of the teaching week, a post-test was conducted. The post-test consisted of two parts, first part with 30 multiple choice questions, and the latter with two programming questions.

Table 3: Experimental procedure

Week	Lesson	Topics	Quizzes
W1	-	pretest	
W2	L1	C integrated environment	Quiz 1 (Q1)
W3	L2	Computer variables	Quiz 2 (Q2)
W4	L3	Types of operators for C programming	Quiz 3 (Q3)
W5	L4	Selection structures	Quiz 4 (Q4)
W6	L5	Control structures	Quiz 5 (Q5)
W7	L6	Array	Quiz 6 (Q6)
W8	-	posttest	

RESULTS

DESCRIPTIVE STATISTICS

The sample size of this study is one hundred and one. In Micro LO group (n=50), 80% of the students were male (n=40) and 20% were females (n=10), whereas 90% male (n=46) and 10% female student (n=5) were observed in Macro LO group (n=51). Most of the students were in the age range of 17 to 19 (Micro LO group n=48, 96%, Macro LO group n=48, 94%) and the rest of students were in the age range of 20 to 22 (Micro LO group n=2, 4%, Macro LO group n=3, 6%). Ninety four percent of the students in Micro LO group were Malaysian and 90% of Malaysian students were observed in the Macro LO group. Three international students were in the Micro LO group and five in the Macro LO group were from the Middle East, Indonesia, China, and Africa. Pre-test mean score between Micro LO group (n=50, m=4.66) and Macro LO group (n=51, m=4.51) suggests that the level of students' prior programming knowledge were similar.

HYPOTHESES FOR COMPARISON

The following hypotheses formulated to compare the significant difference between Micro LO and Macro LO in enhancing the programming knowledge. Statistical significant was set at P < .05.

$H_{0:}$ The post-test mean between Micro LO group and Macro LO group is not significantly different.

$H_{1:}$ The post-test mean between Micro LO group and Macro LO group shows significant difference.

Shapiro-wilk test conducted on the dependent variable revealed data was normally distributed for both groups and Levene's test for equality of variance indicated there was homogeneity of variance. The Micro LO group gained higher post-test mean score (n= 50, m=28.33) than Macro LO group (n=51, m=24.31). The result of an independent t-test, t(99) = 3.615, p=0.00, suggested there was significant mean difference between Micro LO group and Macro LO group. Thus, reject H_0 in favour of $H_{1.}$ Smaller LO works better in aiding programming learning compared to larger LO.

Data collected on weekly quiz marks was examined for further analysis on how the different sizes of LO either facilitated or hindered programming learning. Figure 6 shows the comparison of the quiz mean scores between the Micro LO and Macro LO groups.

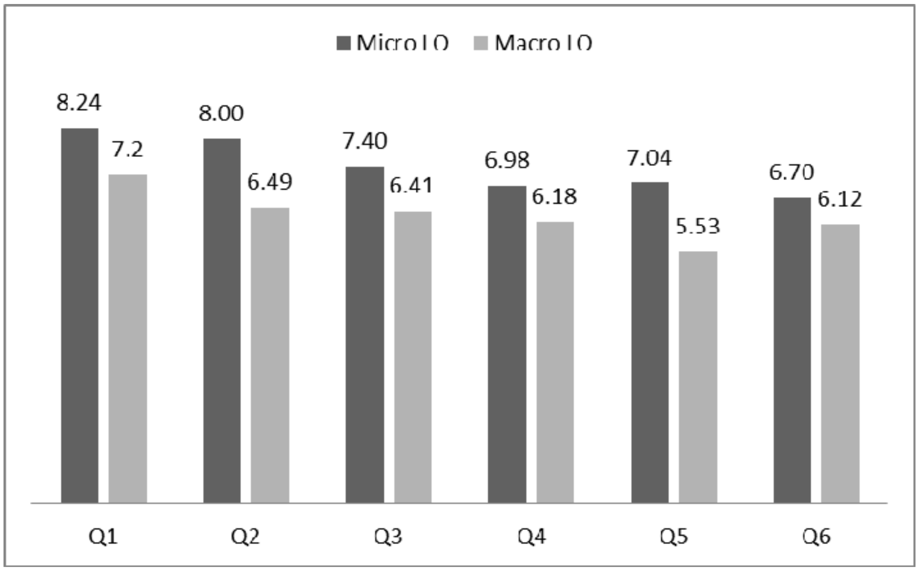


Figure 6: Quiz mean score for Micro LO and Macro LO groups

The weekly quiz mean score shows Micro LO group has been performing better than Macro LO group in all the quizzes. Micro LO seems to be useful in imparting programming knowledge as it is precisely on what the students have to understand. It is observed that the size of learning object has played a role in improving the knowledge in every lesson. Lesson 1 to Lesson 3 (see Table 3) is regarded as easier compared to the other three lessons (L4 to L6). The mean score from quiz 1 to quiz 6 shows Micro LO performed better than Macro LO and it suggests that the size of LO could be one of the reasons for this. A strong understanding on the low-level programming concepts is important to write a program which requires the integration of several low-level concepts. The post-test results showed that students in Micro LO performed better than Macro LO group. The Micro LO group inclined toward progress of being able to perform better in the post-test.

PERCEIVED USEFULNESS OF LEARNING OBJECTS

A survey data collected on students' learning experience in using the LO's in the lab setting and was aimed to find out how useful the learning object would be in delivering the programming knowledge to novices. Table 4 shows the Cronbach's alpha for the survey items and the results indicates a high reliability.

The survey results for items ranging from C1 to C5 (see Table 5) shows that the Micro LO group did not respond negatively at all compared to the Macro LO group. However, the median score in Macro LO groups (item C3, median = 3) suggests few students do have lack of interest in using the CO. Overall, students in both groups agreed that CO have helped them in learning programming. A similar result was observed for SO (see Table 6). Overall, students in both groups responded positively towards the use of SO.

Table 4: Internal reliability values for Content Objects

Items code	subscales	α
C1 – C5	Learning experience in using Content Object	0.871
S1 – S5	Learning experience in using Self-assessment object	0.874
R1 – R2	Perceived usefulness of Content object and Self-assessment object	0.823

Table 5: Survey results for Content Object

		1	2	3	4	5	
Item code	Group	Strongly Agree	Agree	Uncertain	Disagree	Strongly Disagree	Median
C1	Micro LO	17 34%	30 60%	3 6%	-	-	2
	Macro LO	18 35.3%	19 37.3%	14 27.5%			2
C2	Micro LO	14 28%	31 62%	5 10%	-	-	2
	Macro LO	13 25.5%	24 47.1%	13 25.5%	1 2.1%		2
C3	Micro LO	11 22%	34 68%	22 68%	-	-	2
	Macro LO	9 17.6%	16 31.4%	24 47.1%	2 3.9%		3
C4	Micro LO	10 20 %	28 56%	12 24%	-	-	2
	Macro LO	10 19.6%	25 49%	15 29.4%	1 2%		2
C5	Micro LO	17 34%	26 52%	7 14%	-	-	2
	Macro LO	15 29.4%	18 35.3%	17 33.3%	1 2%		2

Table 6: Survey results for Self-Assessment Object

		1	2	3	4	5	
Item code	Group	Strongly Agree	Agree	Uncertain	Disagree	Strongly Disagree	Median
S1	Micro LO	16 32%	31 62%	3 6%	-	-	2

	Macro LO	15 29.4%	23 45.1%	12 23.5%	1 2%	-	2
S2	Micro LO	15 30%	24 48%	9 18%	2 4%	-	2
	Macro LO	7 13.7%	19 37.3%	24 47.1%	1 2%	-	2
S3	Micro LO	14 28%	27 54%	9 18%	-	-	2
	Macro LO	11 21.6%	26 51%	13 25.5%	1 2%	-	2
S4	Micro LO	14 28%	29 58%	7 14%	-	-	2
	Macro LO	10 19.6%	24 47.1%	16 31.4%	1 2%	-	2
S5	Micro LO	13 26%	28 56%	9 18%	-	-	2
	Macro LO	10 19.6%	24 47.1%	16 31.4%	1 2%	-	2

Table 7 shows that Micro LO group ranked content object (RI) more useful than Macro LO group. Eighty-six percent (n=43) of the students in the Micro LO group found CO useful whereas 51% of students in the Macro LO group (n=26) rated CO useful. Similarly, 82% (n=41) of students in the Micro LO group responded positively towards the usefulness of the SO (R2) compared to those in Macro LO group (61%, n=31). Even though the Macro LO group did not respond negatively towards the use of the LO in the lab but the post-test mean score (Micro LO group n= 50, m=28.33; Macro LO group, n=51, m=24.31) shows that the larger LO did not really aid in programming learning. The Micro LO group responded positively to all subscales compared to the Macro LO group and the result indicated that students benefitted from using the LO when they are smaller.

Table 7: Survey results for usefulness of Learning Objects

Item code	Group	1	2	3	4	5	Median
		most useful	moderately useful	useful	least useful	Not useful at all	
R1	Micro LO	19 38%	24 48%	7 14%	-	-	2.00
	Macro LO	10 19.6%	16 31.4%	43 45.1%	1 2%	1 2%	2.00
R2	Micro LO	25 50%	16 32%	9 18%	-	-	1.50
	Macro LO	17 33.3%	14 27.5%	20 39.2%	-	-	2.00

DISCUSSION

Collectively, learning objects are useful in learning programming in a lab setting. Novice learners have a great interest in using the LO as a learning support. As lecturing was regarded as one the ineffective ways of delivering programming knowledge to students, it is vital to understand how effective LO would be, if it is used to engage students actively. The size of the LO object is one of the important pedagogical aspects when it is used as a tool to aid the programming learning process. Even though students showed interest in using the LO in the lab but that could not be one of the factors in determining the success of the LO in delivering the knowledge. The Micro LO group performed better in the post-test and is aligned with the quiz mean score. Students who had a strong understanding of the low-level programming concepts such as computer variable, operators, selection and control structures have performed better in the post-test.

A student's ability in writing a complete program depends on the understanding of low-level concepts (Ala-Mukta, 2003). A complete program requires them to integrate several low-level concepts and the right choice of structures to solve the programming problem. When the students build the understanding gradually, they tend to show their ability in writing a program. The mean score on quizzes showed that the Micro LO students have built their understanding from the first lesson until the difficult lesson on the concept of an array. Smaller LO object that carries a single learning outcome is useful to improve the programming knowledge. Self-assessment object helps students to evaluate their own understanding and misconception. Sequencing the smaller LO is essential to ensure it tails the instructional programming activities (Wiley, 2000). The prevailing focus on learning objects is mainly to overcome the issues of cognitive overload. Therefore, the size of LO really matters if the concern is towards this above-mentioned issue. Integration of LO is encouraging as several types of LO could be assembled for a particular lesson. However, it has the possibility of the LO itself developing cognitive overload. As the idea behind the LO is to reduce the cognitive overload, the size has to be small, but the proper instructional plan on sequencing several LO is essential.

Table 5 shows that both the Micro LO (item C4, 76%, n=38) and Macro LO groups (item C4, 69%, n=35) rated the size of the object just right based on the time allotted to access the LO. Even though students respond positively towards the size but the effect on programming learning is different. Obviously, the post-test mean score and quiz mean score revealed the Micro LO group performed better than the Macro LO group. Smaller LO contains i.e. content object, which carries a specific low-level programming concept that requires a student to master before learning to write a complete program in order to solve programming problems. Students gradually master the interrelated concepts and then they learn to integrate them when writing a program. Shaffer, Doube, and Tuovinen (2002) pointed out that the schema formation is a dynamic process that builds more complex schema by assimilating lower level schemas into higher-level schemas. A similar scenario is possible when several types of smaller LO is integrated in the programming lesson. Students have the ability to write a program when they have strong understanding of low-level concepts with the use of smaller LOs.

Keeping the size of the LO smaller is promising because it is highly reusable. Pedagogically, the LO can be used in other learning environments such as blended E-learning or E-learning environments. It is cost effective if higher institutions invest on LO as part of teaching and learning practice. As the size of LO grows bigger, the degree of reusability and reliability in delivering the programming knowledge declines greatly. Numerous studies have pointed out the fact that the smaller and more specific the learning object, the greater its reusability will be (Silveira et al., 2005). Macro LO requires additional support pages such as help page, home page, or site map to ease the learnability and accessibility. The problem of cognitive load occurs as students need to learn and understand several pages before they can actually engage in cognitive activities in the LOs. Using Macro LO in the lab is also challenging because students have the freedom of choice in selecting the LO to use during the lesson. This can also impede the learning because programming requires both declarative and procedural knowledge (Schulte and Bennedsen, 2006). It is important for students to understand the concepts (declarative knowledge) before knowing how to use the syntax in writing the program (procedural knowledge). The Micro LO group used the CO after they have learnt the concepts and SO during the practical activities. Even though the similar procedure is used for the Macro LO group but the possibility of students mismatching the LO is not deniable. Therefore, to enhance programming understanding, the Micro LO should be integrated with proper instructional goals and activities in the lab which works better than Macro LO.

CONCLUSION

This study found that the size of LO has some effects on programming learning. Micro LO which are the smaller LO are highly reusable and useful in delivering programming knowledge to novices. Students showed interest in using the LO in a lab as a support learning tool. However, students' interest may not be a factor to determine the success in learning programming. Even though, students showed relatively positive respond toward the use of Macro LO (the larger LO), the result of performance in post-test and quizzes reflected their understanding is not strong than those used the Micro LO. The size of LO is important to ensure it is reliable in playing a role in the programming learning process. There are no guidelines on how small a LO should be. Academicians, instructional designers, and content developers may have different insights on the sizes of LO. Polsani (2003) suggested that the concept or idea should determine the size of the learning object (logical size). This study suggests a small LO should be accessed within five to ten minutes (physical size) and it should contain the key programming concept to accomplish a single learning outcome (logical size). Several types of LO is essential to promote the declarative and procedural knowledge acquisition. However, the LOs have to be self-contained to support the learning outcome, and also to ensure it is highly reusable.

REFERENCES

Advanced Distributed Learning (2012). SCORM. Retrieved October 2, 2012, from http://www.adlnet.gov/capabilities/scorm

Ala-Mukta, K. (2003). *Problems in learning and teaching programming*. Codewitz Needs Analysis: Institute of Software Systems, Tampere University of Technology.

Allen, C.A., and Mugisa, E.K. (2010). Improving learning object reuse through OOD: A theory of learning objects. *Journal of Object Technology, 9*(6), 51-57.

Balatsoukas, P., Morris, A., and O'Brien, A. (2008). Learning objects update: Review and critical approach to content aggregation. *Educational Technology & Society, 11(2),* 119-130.

Beck, R.J. (2010). Learning Objects. Retrieved September 28, 2012, from http://www4.uwm.edu/cie/learning_objects.cfm?gid=56

Bergtrom, G. (2006). Clicker set as learning objects. *Interdisciplinary Journal of Knowledge and Learning. (2),* 105-110.

Fetaji, M., Loskovska, S., Fetaji, B., and Ebibi, M. (2007). Combining virtual learning environment and integrated development environment to enhance eLearning. In *Proceeding of the International Conference on Information Technology Interfaces, Croatia.*

Hodgins, H.W. (2004). The future of learning objects. Retrieved June 11, 2012, from www.reusability.org/read/chapters/hodgins.doc

IEEE Learning Technology Standards Committee (2002). Draft standard for learning object metadata. Retrieved December 14, 2012, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Jacobsen, P. (2002). History and definition of RLOs. Retrieved August 15, 2012, from http://www.mcli.dist.maricopa.edu/ocotillo/retreat02/docs/rlos.pdf

Krull, G., and Mallison, B. (2004). Modular content hierarchy diagram. Retrieved September 28, 2012, from http://edutechwiki.unige.ch/en/Learning_object#Definition

Learning Circuits (2005). A Field Guide to Learning Objects. Retrieved January 15, 2013, from http://www.learningcircuits.com/2002/jul2002/smartforce.pdf

London Metropolitan University. (2004). Learning Objects for Introductory Programming. Retrieved October 2, 2012, http://www.londonmet.ac.uk/ltri/learningobjects/

Matthews, R., Hew, S.H., and Harprith, K. 2008. Introductory programming students view on blended learning. In *International Conference on Social Sciences and Humanities*, Malaysia.

Nokelainen, P. (2006). An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students. *Educational Technology & Society*, 9 (2), 178-197.

Polsani, P.R. (2003). Use and abuse of reusable learning objects. *Journal of Digital Information, 3*(4), 164.

Schulte, C., and Bennedsen, J. (2006). What do teachers teach in introductory programming? In *Proceedings of the Second International Workshop on Computing Education Research.* p.17-28. USA

Seung, Y.Y.C. (2007, August 27). Learning objects-based e-Learning: Content design, methods, and tools. Learning Solution Magazine. Retrieved August 1, 2012, from http://www.learningsolutionsmag.com/articles/162/learning-object-based-e-learning-content-design-methods-and-tools

Shaffer, D., Doube, W., and Tuovinen, J. (2003). Applying cognitive load theory to computer science education. In M. Petre & D. Budgen (Eds), *Workshop of the Psychology of Programming Interest Group (*pp.333-346).

Sicilia, M., Garcia, E. (2003). [Review of Research in Open and Distance Learning]. *The International Review of Research in Open and Distance Learning*. 4(2), 2-11.

Silveira, I. F., Araujo, C. F., Amaral, L. H., Oliveira, I. V., Schimiguel, J., Ledon, J. S. (2005). Granularity of learning object. *Proceedings of the Informing Science and IT Education Joint Conference Abstract,* 158

Tampere University of Technology. (2012). Codewitz Learning Objects. Retrieved October 2, 2012, from http://www.codewitz.org/learningobjects.php

Watsons, J. (2010). A case study: Developing learning objects with an explicit learning design. *Electronic Journal of eLearning. 8(1),* 41-50.

Wiley, D.A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. Retrieved February 2, 2012, from http://wesrac.usc.edu/wired/bldg-7_file/wiley.pdf

Yau, J.Y.K (2004). *Learning Objects and the teaching of Java programming.* Masters' Thesis, University of Warwick. UK.