Introduction

Course Curricula

In all partner institutions the LOs were used one way or the other in the basic programming course where the programming language was C++. Some partners used LOs also in more advanced courses. The contents of the basic courses were quite the same in the beginning. Some of the courses held were bigger collections and some were consisted only the very basic structures of programming. These structures or topics in each basic course were variables, operators, selection, loops and arrays.

No differences in cultural or in other aspects were found in the course curricula between the partner institutions. The differences in curricula were mainly due to the different organisation of the programming studies. In this project all partner institution were having own courses from their regular study program. No common course was used.

The following comparative summary will concentrate mainly on the course of basics, because other courses didn't have similar courses to be compared.

Study Setup

The most of the partner institutions made the survey by using the same questionnaire as web form. One partner institution had made the survey for two groups of students for comparative reasons. For the first group of students LOs were not used and for the second group LOs were used. Two partner institutions made the survey by using both the questionnaire and by personal interviews for the students.

According the reports by partner institutions there cannot be seen any cultural reasons for selecting different ways to handle the study setup. The reasons fro the selected methods should be seen as personal differences not as cultural differences.

Results

The amount of students participating in the basic courses was between 30 and 60 with five partner institutions. One partner was exception when reporting 800 – 1000 students for all four courses where LOs were reported as used material. This institution is assumed to have around 500 – 600 students in the basic course.

Because one of the partner institutions had so different situation in the basic course it is very hard to compare the results between other partner universities. As there are two partner universities in Finland and one institution is the one with a big amount of students and the other one is having about the same as the others the difference cannot again be seen as a cultural difference between countries. The difference can only be seen as the difference in the size and in the institution's course organisation. These aspects are usually depending on the teaching culture formed or selected by a single university.

Almost all partner institutions reported some of their students had had problems with the LOs made by shockwave technology. The problem had been either the complete lack of player (for Linux) or the difficulties in getting or installing the player. Only difficulty differing from other partner institutions' results was the comment by one partner institution's students. As one of the most common difficulty it was mentioned the English language used in LOs. This could be seen as the first difference in cultural aspect, because other partner institutions were not mentioning language in their reports.

The partner institution with the largest amount of students in the basic course had been able to measure also the activity of the students during the course. The result they found was the students who were using the most of time to study were using the largest amount of time on studying on LOs as well. Thus the results in this survey do not give the whole picture of the usefulness of the LOs.

Analysis

One common conclusion from all reports is the visualisations are good for students when they are learning the basics of programming. Also the common conclusion by all reports is the way how the LOs are used or presented during the course should be improved. Currently the students are not using the LOs as much as they could. The improvement could be made by changing the way how the LOs are as part of the learning material. This is how the LOs are integrated into learning material and teaching. Another way to improve the usage of the LOs would be to decrease the amount of technical limitations (like players needed etc.) or by improving the usability of the LOs.

Suggestions for Finalising the LOs

This is the list of collected different suggestions for improvement:

- To increase the number of LOs
- To develop such template, where it is easier to modify LOs
- To develop LOs in native language.
- To develop an interpreter-based tool for creating LOs
- To use platform independent technologies to support a wider acceptance of the LOs.

Tampere University of Technology (TUT)
Institute of Software Systems

Course Curricula

In TUT the visualization tool VIP was used on the introductory programming courses at the following times:

- OHJ-1100 Programming I (4ECTS): periods 2-3 and 4-5 of academic year 2005-2006
- OHJ-1101 Programming I e (4ECTS): periods 2-3 of academic year 2005-2006
- OHJ-1151 Programming II e (5ECTS): periods 4-5 of academic year 2005-2006
- 8100200 Introductory course on programming (9ECTS): periods 2-4 of academic year 2004-2005

The yearly amount of students on the mentioned courses is approximately 800-1000. Two teachers and ten teaching assistants have been using the visualization tool in their teaching. The prerequisites for the courses are limited to only basic knowledge of computer literacy and they are the first programming course for the students. The programming language used on the courses is C++.

Study Setup

The usage of VIP has been studied by an empirical test setting and a quantitative questionnaire during the academic year 2005-2006.

The Empirical Study Setup

The experiment took place on the course Programming I e. The use of the visualizations was tested in the weekly exercise sessions.

We had a group of students who used the visualization tool for preparing for the exercise session and a reference group who did not use the visualizations. In the beginning of the exercise session there was an in-class test for all the students. We analysed the results of the tests statistically. The whole experiment is introduced in detail in the publication [1].

The Questionnaire

The quantitative survey was done for the students of the courses Programming I and Programming I e on the academic year 2005-2006 by a web questionnaire and a printed questionnaire. We analysed different kinds of student groups and their usage of VIP and problems with it. We explored the background and the motivation of the students and compare this information to their experiences about the use of the visualizations. We collected 180 responses from the students and analysed them statistically. The whole experiment and its' analysis is described in detail in the publication [2].

Results

Only the main results of both the studies are presented here. More about the results can be read in the publications [1, 2].

The Empirical Study

We found out that the use of visualizations has a strong impact on the learning of the students who do not have earlier programming experience and the students who find the course challenging. In this subgroup of students the ones who used the visualizations to prepare for the exercise sessions performed statistically better than the students who did not use the visualizations.

We also questioned the way the students had been preparing and figured out that the students using the visualizations had spent more time on studying. Thus, we can not say whether the better results in the test result directly of the usage of the visualizations. It can also be that when using the visualizations, the studying itself is more interesting and the students use more time on it and thus learn better. However, it is not important, if the visualizations improve the learning results directly. The most important result is that they do improve them.

The Questionnaire

About one third (32%) of the respondents had not used VIP. This is surprisingly little since using the tool was completely voluntary. Half of the students (51%) had used VIP on 1-5 different times. Only very few students (18%) had used it more often than that. The amounts of VIP usage are shown in Illustration 1. The motivation for using the tool varied too. About half (48%) of the students who used VIP had tried it just because they were curious about what it was. 41% of the VIP users had better motivation to try using the tool, because they wanted to check something about programming. Minority (12%) told they had been using it more than just trying it.

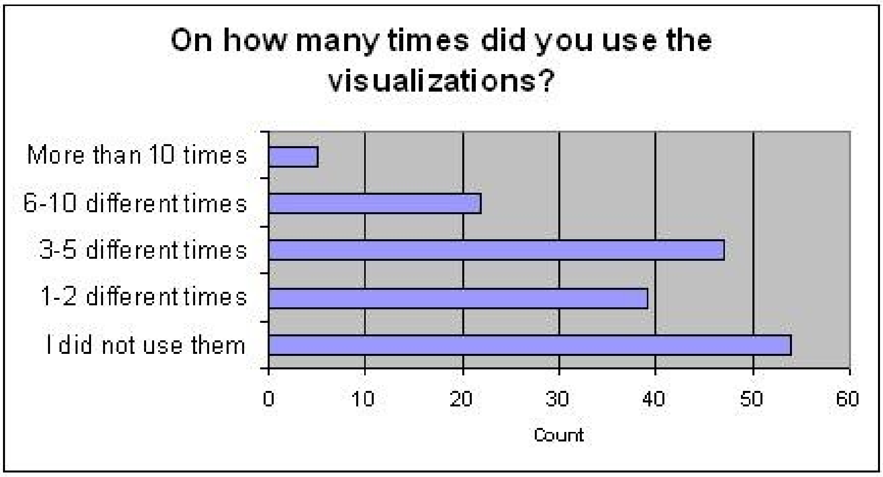


Illustration 1: Usage of VIP.

Almost one third of the students (28%) had had some difficulties with the visualization tool. There was not one certain type of difficulties that would be bigger than others.

The questionnaire asked if the students had used the visualizations. The options for answers included *No, I tried them because I was curious, I tried them because I wanted to check something about programming* and *I've been using them more in learning*. The strongest finding of this survey was that the motivation of the student affects strongly on how much he will use the visualization tool. The students who had tried the visualization tool for finding out something had used it statistically more than the students who tried it just because they were curious. This can be seen in Illustration 2.

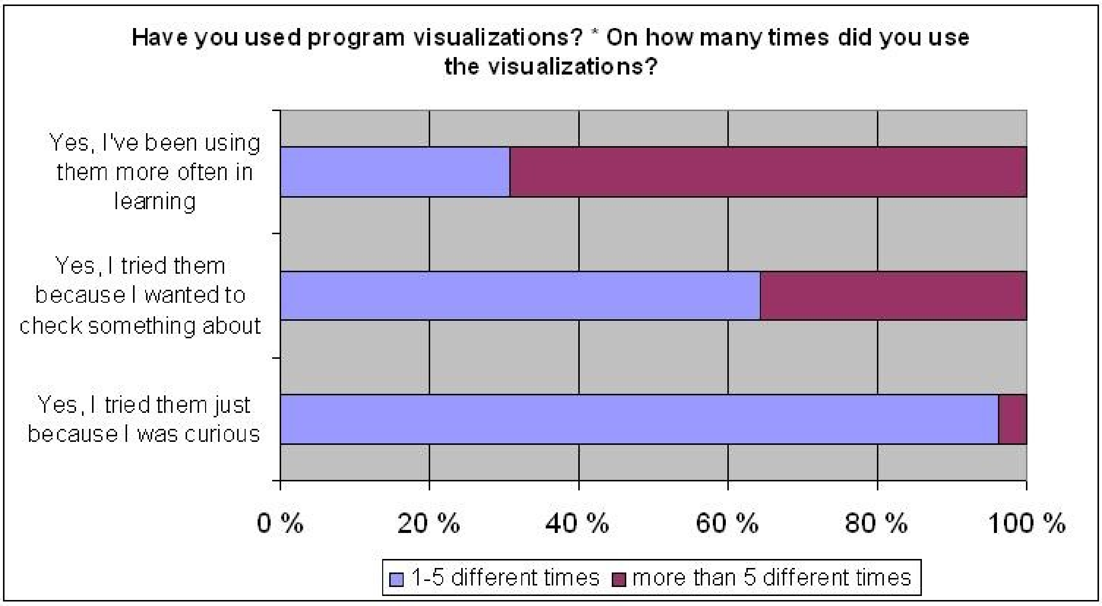


Illustration 2: The motivation affects the amount of the usage of VIP.

The students assessed their own progress on the programming course on the scale *I don't think I will pass the course, The course is challenging but I manage to learn, I don't have difficulties but I'm learning a lot* and *I didn't learn anything new in the course*. When analyzing the usage of VIP of these student groups it can be seen, that the ones using VIP the most are the ones who find the course challenging but who still keep learning. The students who do not think they will pass the course do use VIP but less than the first ones. Both the groups who do not have difficulties on the course are the ones using VIP the least. This phenomenon is presented in Illustration 3.

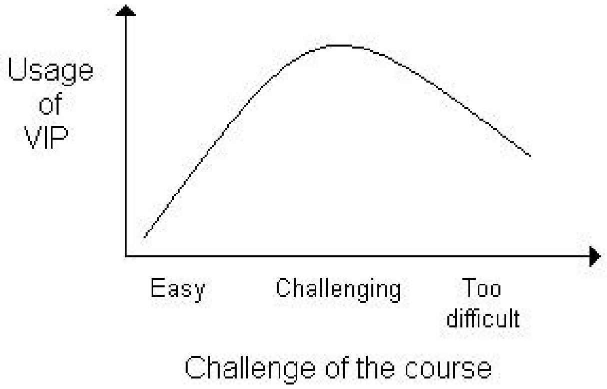


Illustration 3: The challenge of the course affects the usage of VIP.

Analysis

Using program visualizations improve the learning of students with no earlier programming experience and the students who have difficulties in programming. We cannot say whether the better learning results originate from the pedagogical impact of the visualizations or from the fact that they made the students study for a longer time. Either way, using visualizations improved the students' learning and preparation for the exercise sessions which was the purpose. Therefore, we recommend both using visualizations in teaching and using the exercise sessions to integrate the visualizations to the other parts of the course.

The students are interested in using visualization tools voluntarily even if they might sometimes have problems with them so developing the visualizations is definitely not useless. The visualizations are mostly used by the students who find the course challenging but not too difficult to be passed.

The teacher should concentrate on the way they present the visualizations for the students. For instance the teacher should remember to explain the visualizations and their figurative language for the students and to give the students some sort of tasks to work with when they use the visualizations. The teacher can also guide the students with weaker computer using skills to use the visualization tool as a visual debugger if he has problems with using the normal program development environment. Above all the teacher should remember that the students with most difficulties cannot be helped with only technological aid.

Suggestions for Finalising the LOs

The future work on the field of visualizations in TUT is going to be mainly about further development of VIP. At the moment there is an ongoing project whose purpose is to add interactive features to VIP. This project is described in the publication [3].

Also the fact that the students had more difficulties with VIP than the students in the other universities had with the Shock Wave animations has to be considered in the further development. It is logical because the picture animations usually contain less features than VIP. E.g. it is possible to edit and compile your own code in VIP. The numerous complicated extra functions also cause new difficulties.

There exists also other interpreter-based visualization tools than VIP. To counterbalance the student difficulties with the interpreters, we have to bear in mind that the teachers and the visualization tool developers also have difficulties with the picture animations and this is also a hinder for using them in teaching. Albeit implementing an interpreter is laborious, implementing a visualization using it is effortless. Implementing a picture animation is always laborious if only not as laborious as implementing an interpreter. The picture animations are also very limited in their interaction with the user. The international research on the field of visualizations emphasizes that the learner engagement that interaction causes is the most important in making the student learn with the visualization. This should also be considered in the further development. It is impossible to add interactive student engaging functionality to a picture animation whereas adding it to an interpreter has already been done. Thus, the development of an interpreter-based tool is more sustainable.

In the further development of VIP we still need to bear in mind that an eminently interactive hyper tool that has dozens of functionalities but poor usability can be a hinder for learning. The student might just want a simple usable learning material that shows concretically how the program works. The new functionality is of little use unless it is possible for the students to work with it.

References

[1] *Ahoniemi, T. & Lahtinen, E.* 2006. Visualizations in Preparing for Programming Exercise Sessions. In: Proceedings of the Fourth Program Visualization Workshop. 29-30 June 2006, Florence, Italy.
[2] *Lahtinen, E. & Melakoski-Vistbacka, S.* 2006. Student Experiences on Program Visualizations, A Case Study on Visual InterPreter (VIP). Submitted for: SIGCSE 2007, Technical Symposium on Computer Science Education.
[3] *Lahtinen E. & Ahoniemi, T.* 2006. Annotations for Defining Interactive Instructions to Interpreter Based Program Visualization Tools. In: Proceedings of the Fourth Program Visualization Workshop. 29-30 June 2006, Florence, Italy.

TECHNICAL UNIVERSITY OF CIVIL ENGINEERING, BUCHAREST
DEPARTMENT OF CIVIL ENGINEERING
Programming Languages Course

Course Curricula

Course unit description

Course No. 5
Course title: Programming languages
Semester: 3
Number of credits: 3
Lecture: 2 hours / week
Laboratory: 2 hours / week

Item No.	Discipline	C	S	L	P	SI	TO	CR	Prerequisit e	Exam. form
5.	Programming Languages	2	-	2	-	2	6	3		C

Notations: C – course; S - seminar; L - laboratory; P - project or practice; SI- individual study: TO - total hours per week; CR credits granted; E – exam; C assessment.

Lecturer: Assistant Professor Dr. Ion MIERLUS MAZILU
University / Department:
Technical University of Civil Engineering of Bucharest
Department of Civil Engineering
Bd. Lacul Tei 124, Sector 2, RO 72302, Bucharest, Romania
Tel. (+40)-21-2421208, Fax (+40)-21-2420781
Status of the course in the study program: Compulsory course.
Course description: Programming languages basis. Algorithms. Flowcharts. Pseudo-code. Structured programming. Object-oriented programming principles. C/C++ language features. Basic knowledge. C/C++ program structure. Types of data. Constants. Variables. Statements. Operators. Expressions. Evaluation statements. Decision statements. Loop statements. Arrays and character strings. Pointers. Functions. Input/Output devices. Objects and classes.
Objectives of the course: The student should obtain theoretical and practical knowledge for program development.
Teaching method: Lectures and tutorials
Prerequisites: Basic knowledge of Computer Science.
Teaching aids: Textbooks are available for students.
Examination: Practical development of programs and oral examination.
Bibliography:
CRISTEA Gh. : “Programming Languages Notes, TUCEB, 1998.
BJARNE S.: “The C++ language”, Addison Wesley Pub. Co, 1993.
CODEWITZ LO’s: http://codewitz.utcb.ro

LO’s are used in all laboratory classes (Types of data, Constants, Variables, Statements, Operators, Expressions, Evaluation statements, Decision statements, Loop statements, Arrays and character strings, Pointers, Functions, Input/Output devices, Objects and classes).

Study Setup

The study was organized on the same course in two years:

- in the first year students do not have the program visualization learning objects as learning material available

and

- in the second year they have the program visualization learning objects available. The students study exactly the same course.

The effects of the program visualization learning objects on the results are then analyzed by the final course points and grades and activity of the students and also with a survey about all learning materials available held at the end of the course.

The study was made in Technical University of Civil Engineering during two years 2004-2006 for Civil Engineering students department who are non-major students in programming. These civil engineering students have one obligatory programming course included in their studies. This course is called Programming Language and the course covers the first steps of programming like variables, selection, loops, arrays, functions, etc. The Programming Language course is timed in the second semester of the second year of their studies and the scale of the course is three credits. The course consists of lectures and lab exercises. Each student has two lecture hours and also two lab hours per week.

Each year four groups of around 25 civil engineering students are formed. The groups have the Programming Language course parallel with the exactly same content. It is assumed in the course everyone has no previous knowledge about programming.

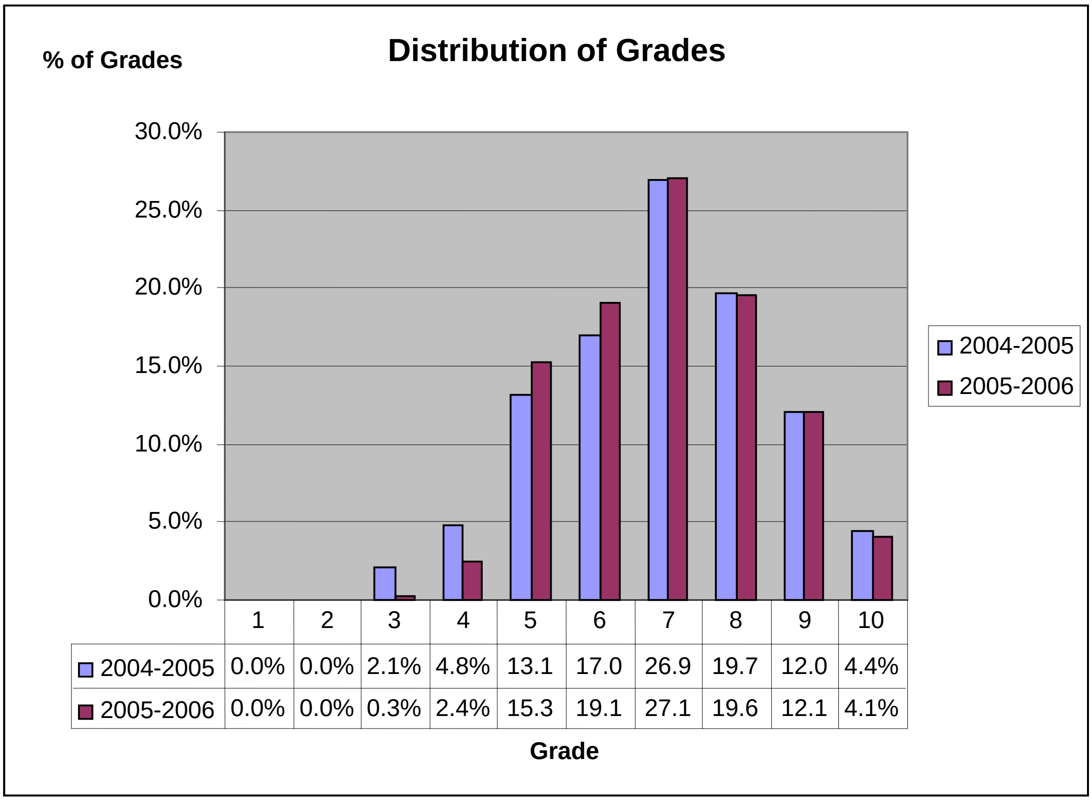
The organization of the study was divided in two years. In 2004-2005 the students had the courses in a traditional way with no program visualization learning objects available. The 2005-2006 courses were organized with program visualization learning objects available for the students as learning material in the local network throughout the course. In 2006 all the students were guided by the teacher to use the learning objects as the extra learning material.

The program visualization learning objects were occasionally also used as program examples by the teacher. The both year courses were held by the same teacher with the same material and with the same outside classroom assessments for the students. At the end exactly the same paper exam was given to the students in both years. The exam papers were not given back to the students. Thus the questions in the exam are considered not to be known by the next year students.

This study consists of two parts. The first one is the results of the courses in two years. The first study takes also into account also the activity of the outside classroom assessments made by the students during the course. The second study is the survey made for the students at the end of the course. In this survey the students answered to the questions concerning their background and the usefulness of different kind of learning materials in their learning process. This survey was conducted during the last lessons of the course.

Results

The results consist of the final grade and the activity of the students. Final grade is between 1 and 10 where 4 is failed, 5 first grade for the completion of the course and 10 is the best grade (Figure 1).



	1	2	3	4	5	6	7	8	9	10
🟦 2004-2005	0.0%	0.0%	2.1%	4.8%	13.1	17.0	26.9	19.7	12.0	4.4%
🟥 2005-2006	0.0%	0.0%	0.3%	2.4%	15.3	19.1	27.1	19.6	12.1	4.1%

Figure 1

In the survey the students were asked about their programming skills before the course, about the skills in using computers in general, about all learning materials used in the course and program visualization learning objects (used only in 2004-2005 courses). The survey result about visualization learning objects was very good (Figure 2).

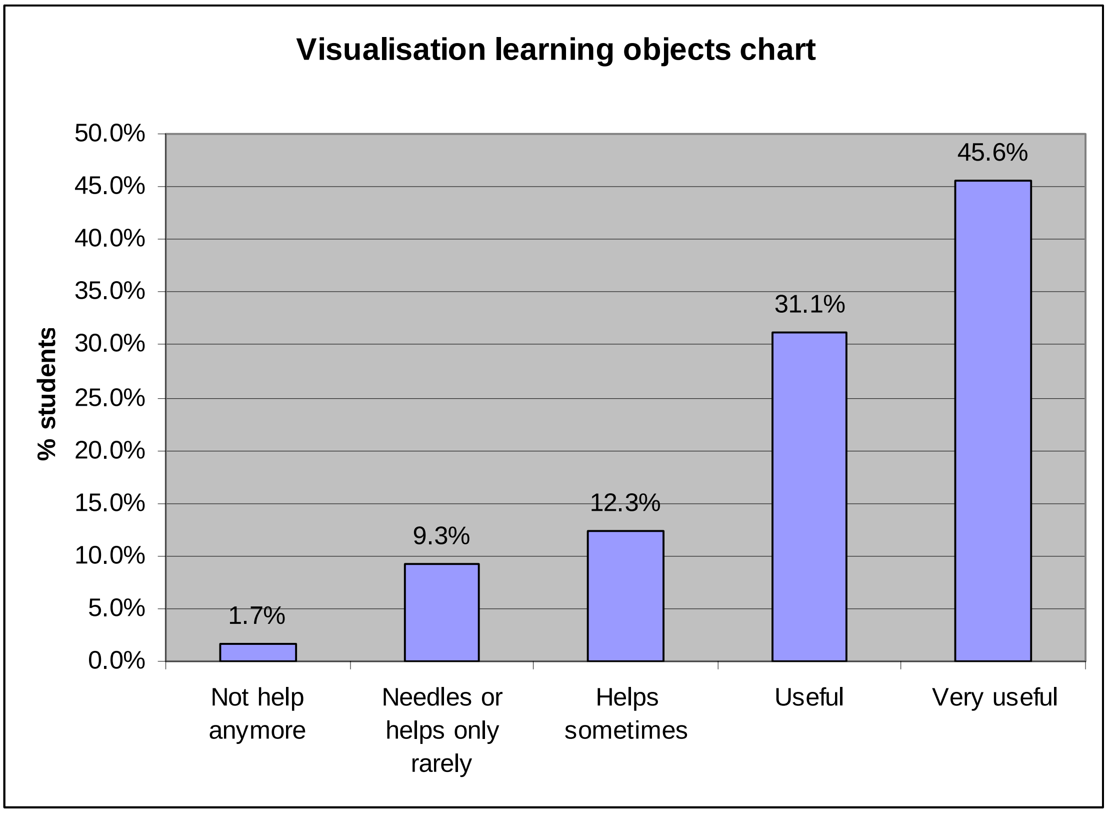


Figure 2

Analysis

Interactive learning object is an idea that many teachers welcome in their search for new methods and support for novice programming students. What subjects to explain with the help of learning object is always a question and in the Codewitz project the need analysis was helpful for the project partners to choose where to begin. To introduce the learning objects to teachers and students is still an ongoing process and all teachers can become partners in Codewitz, get access to the material bank and take part in developing more interactive learning objects.

It is quite clear that students believe that learning objects can be useful for them as novice programming students. But it is also quite clear that more introductions and better integration of learning objects is needed to encourage students to use them more frequently as a normal part of their programming study. Only a part of the students seem to use extensive material outside the classroom and although they know about good material they somehow do not use it. Here we might have to deal with students learning style and their immaturity as learners. Codewitz learning objects are not the witchcraft we might need in teaching programming today but I believe it could be useful especially if it becomes integrated into teaching and learning and a natural part of students programming life.

Suggestions for Finalising the LOs

The idea is to increase the number of LOs, to have more and more examples for our students.

From Iceland I

Usefulness of the learning objects

At the Reykjavik University we have not so far used the learning objects as an integrated part of the teaching programming; it has been more as a comprehensive (additional) material for the students to use when working on the course. The teacher have shown the students some learning objects, both in lectures and in workshops, and pointed out to them where to find them on the web.

Survey at Reykjavik Univeristy

To get some ideas about the students' attitudes toward the learning objects we put one question in an ongoing online web questionnaire for first year students in programming with C++. Of 33 students who answered a question about the usefulness of the learning objects 20 (61%) said that it was very useful or relatively useful in their study and only 5% sayd they whre usless

To get more information about the students' attitudes a question was sent by e-mail asking them about what they thought was positive and negative about the learning objects. Only 9 responded and three of them had not used the learning objects, only seen them explained in the classroom. One complained that he could not use the material on the web browser Opera because of the plug-in needed. The student wanted more information about what is needed and how to get the plug-ins if you are not using the web browser Internet Explorer. One student was extremely negative and said the material was rubbish. One student wanted more explanation of the tasks and one said that some of the objects needed to be simpler to understand. The positive comments were the following:

- *"Over all this was good and I used it a lot, especially the one where I was supposed to find the answer my self."*
- *"Codewitz exercisers are very good to get a grab of what is going on when you run a program, one can see how for example the loops are working and so on."*
- *"The main advantage is the understanding that the user gets for the function of each action in C++ and this step by step practical demonstration in the many examples and exercises gives a good support. The Codewitz learning objects are very useful as an interactive exercises and I can se no drawback."*
- *"What I saw seems to be clever but as I had god experience of b-programming before I came her I did not use this much."*

CONCLUSIONS

Interactive learning object is an idea that many teachers welcome in their search for new methods and support for novice programming students. What subjects to explain with the help of learning objects is always a question and in the Codewitz project the need analysis was helpful for the project partners to choose where to begin. Designing the plans was also good idea, you could plan what you wanted to do and get comments on your ideas before you started to make the object and there are now about 300 plans waiting for someone to turn them into real interactive objects. To introduce the learning objects to teachers and students is still an ongoing process and all teachers can become partners in Codewitz, get access to the material bank and take part in developing more interactive learning objects.

It is quite clear that students believe that learning objects can be useful for them as novice programming students. But it is also quite clear that more introductions and better integration of learning objects is needed to encourage students to use them more frequently as a normal part of their programming study. Only a part of the students seem to use extensive material outside the classroom and although they know about good material they somehow do not use it. Here we might have to deal with students learning style and their immaturity as learners. Codewitz learning objects are not the witchcraft we might need in teaching programming today but I believe it could be useful especially if it becomes integrated into teaching and learning and a natural part of students programming life.

Tampere Polytechnic
Esa Kujansuu

Course Curricula

`Course:` Programming 1 (Ohjelmointi 1), 3,00 ECTS
`Language:` Finnish
`Class Hours:` weekly 2 theory hours and 3 lab hours per student
`Dates:` 24.10.2006 – 21.12.2006
`Evaluation:` By exam and assignment activity

`Organisation:` Two groups had this course parallel with the same teacher and the same materials, assignments and exam

`Required Skills:` Basic Computing Skills. No previous programming skills are required

`Course content with used LOs:`

Introduction (5 hours)
- No LOs in use

Variables, Arithmetic, Decision Operators (5 h)
- 1 LO in use

Algorithms, Pseudo code, Control Structures (`if`, `if/else` and `while`) (5 h)
- 3 LOs in use

Composition of Algorithms, Counter Controlled Loops, Loops, Nested Control Structures (5 h)
- 8 LOs in use

Assignment, operators ++ and `--`, `for` loops (5 h)
- 10 LOs in use

`switch` Structure, `do-while` Structure, Keywords break and continue, Logical Operators (5 h)
- 9 LOs in use

Arrays (5 h)
- 4 LOs in use

Use Of Arrays, Multi Dimensional Arrays, Exam (10 h)
- 6 LOs in use

The LOs were there available for the students. There was guidance at the beginning of the course, but there was no requirement to use the LOs.

Study Setup

This study was conducted during the last week of the studies. The students filled the form during lab hours. The last week was most likely the reason for only some of the students were present in the lab.

Results

General

These results have no aspect of differences between the two parallel groups because the background of the students and the organisation of the course were exactly the same. These results show both groups together.

65 % of the students participated this study, which is 32 of total 49 students.

The majority of the students (88%) had started to study programming during this course. According to the teacher the rest of the students were retaking the course. 66% of the students had no previous programming skills and only one student had had a lot of studies before the course.

More than half of the students (59%) thought the course was challenging, but they believed they will manage it. 34 % thought they have now problems with the course. In practice all the students had at least moderate computing skills and they used the computer mostly at home.

About using visualizations

63 % of the students said they had used the visualizations during the course.

In the following part the percentage of the student refers to the number of students in the study, not only to the ones saying they had used them.

Only two of the students reported they had difficulties in using the visualizations. These two students reported the reason was the visualizations where difficult to understand.

34 % of students said they had used the visualizations more than 6 times during the course. When teacher had shown the visualizations during lecture or exercise 38 % percent said visualizations were beneficial for them at least a couple of times.

The students found the visualizations as most beneficial when they were using visualizations by themselves. 44 % said visualizations had helped them at least a couple of times when studying on their own time.

About visualization topics

In the area of learning variables visualizations were considered to be beneficial at least a couple of times by 25 % of the students (Table 1.). In learning selection and loop structures 53 % of the students thought visualizations had been beneficial at least a couple of times. In learning arrays 25 % said visualizations to be beneficial at least a couple of times.

	Variables	**Selection structures**	**Loop structures**	**Arrays**
don't know / haven't done that	23	15	15	23
never	1	0	0	1
a couple of times	7	5	4	1
sometimes	0	2	3	4
often	1	10	9	2
almost always	0	0	1	1

Table 1.

In the areas of the course the visualizations were most beneficial according to the students in learning selection structures and in learning loop structures (Figure 1.).

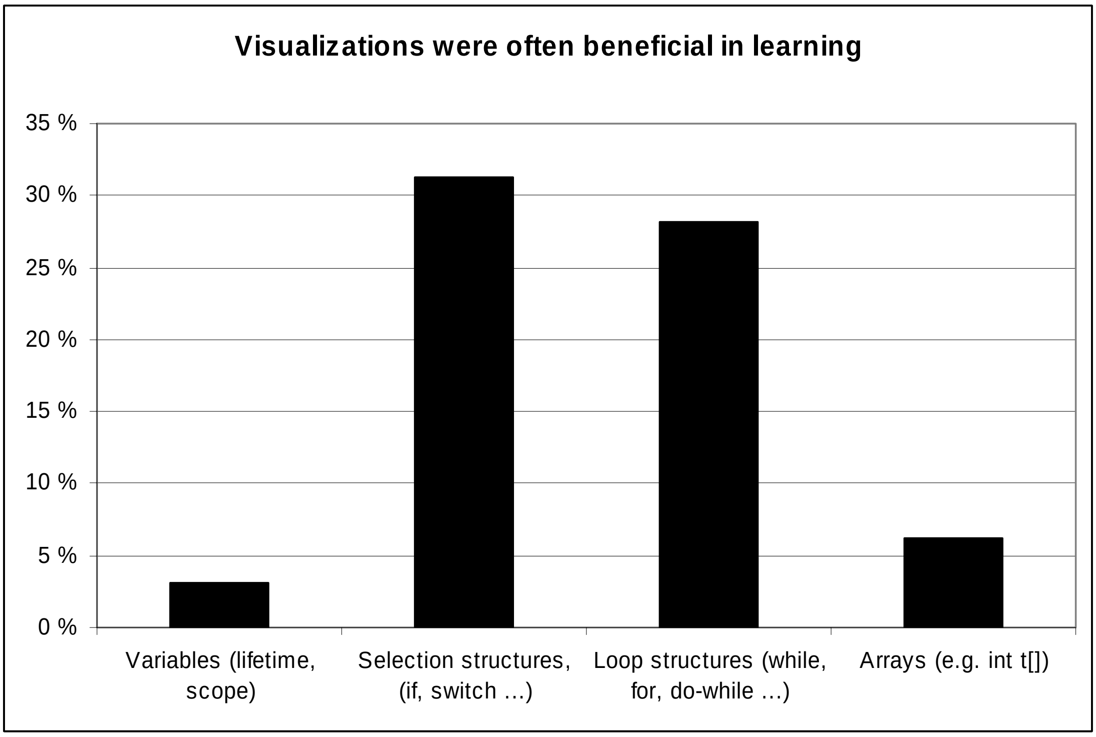


Figure 1.

Analysis

Visualisations obviously help some students when they learn the first steps of programming skills. The most helpful area is when there are clear structures like selection or loops to be learned.

The problem with visualisation material is the same as with any material: How to get the students to use it? Many of the students never used the visualisation materials. The students need motivation for the learning process and if there is a lack of motivation present no material can help these students.

To increase the motivation of the students the visualisation material should be better integrated into course materials. The use of visualisations as learning material should be basic part of the course. At the moment the biggest problem is the amount of the LOs. There should be much more LOs for the courses of programming.

Suggestions for Finalising the LOs

This study did not bring up any mistakes or other problems in the LOs used during this course.

COMPARATIVE TEST RESULTS REPORT -

University Furtwangen, Faculty of Computer Science

Author: Friedbert Kaspar

University Furtwangen,

Faculty of Computer Science

Course Curricula

The title of the course is: Programming 1, Programming with C++. The course in the winter semester 05/06 has been attended by about 40 students. The summer semester 06 was very small with at the beginning 18 and at the end 12 students. This reflects the problems with the unusual bad qualification of the students. In the winter semester the students had been grouped in two groups for the exercises. For the lectures the students are taught together. The course covers essentially the procedural part of C++. The course is divided in 13 chapters. It covers the basics of programming, data types, variables, constants, operators, expressions, assignments, control structures (if, else, for, while, do while, switch), pointers, functions, structures and unions, visibility and life cycle of variables, dynamic memory management, the pre processor and formatted input from and output to the standard IO device.

Learning Objects (LOs) are used throughout the whole course. The LOs are offered to the students by the webpage:

http://webuser.hs-furtwangen.de/~kaspar/local/cpp1/index.html

In fact all material of the course: the script for the lecture, the practical assignments, additional material like a C++ online book and step by step exercises are presented to the students in this way.

The presentation of the LOs is done according to the chapter structure of the course. That means, there are demonstration and exercise LOs offered for all chapters. The coverage of the topics is however different for different chapters, because the availability of LOs from the Codewitz Material Bank has been different for different topics. The LOs have been referred to in the lecture, as well in the exercises.

Study Setup

The survey was conducted in the second last week of the course. The students were asked in the exercises to fill the survey. The first time the feedback how many students filled the form revealed that too few students filled the form. For the first course this feedback came to late to repeat the survey in the same semester, so the survey has been repeated in the first part of the next semester. Additionally to the survey, personal conversations with students have been done to get additional information about the opinion of students in respect of the LOs.

Results

In the average the students studied 4 months at the time of the survey, i.e. in the second last week of the semester. In the average the students had little programming experience, when starting the course. However there have been big differences in the knowledge in programming the students had at the beginning of the course. For all students the material was new, for half of the students the course was challenging. All students indicated having made a good progress in the course. The general knowledge of the students in using a computer is good. The students spent a lot of time working in the computer lab to do the exercises. This is partly due to the fact that they are obliged to have a certain time of attendance, partly due to the problems the students had with the exercises. A considerable part of the work the students also do at home. The main motivation of the students when using LOs was to check their knowledge and to learn more about select topics, which are considered to be difficult.

The students reported no difficulties, when using the LOs, except that the LOs, which needed a Shockwave plugin for the browser could not be used with Linux, because there is no Shockwave plugin for Linux. This missing support for Shockwave under Linux was criticized by an appreciable part of the students.

The students used the LOs mainly after the lectures to check specific topics or before exercises to prepare for the exercises. For the topic "pointers" the students found LOs most beneficially. Further topics for which students found LOs especially beneficially are functions and arrays. When investigating this point in more detail, we recognized that the most difficult aspects of functions and arrays for students grounds in their relationship to pointers. Object oriented aspects have been treated in the course only marginally. Accordingly usage of LOs in this respect has been low.

Analysis

Pointers are obviously the most difficult part of C++ for beginners. Topics which have aspects related to pointers are as well considered to be difficult by the students. This suggests focussing the development of LOs in these areas.

The usage of LOs by the students is closely related to their programming course. To use an LO is normally motivated by a concrete topic of the lecture or a concrete topic of an assignment or a question of an examination. This means the LOs have to be closely related to the course. The selection of LOs offered to the students and the single LO presented have to be adapted to the course to be accepted, i.e. used by the student.

The impossibility to use LOs based on Shockwave under Linux has been a considerable obstacle for the usage of these LOs by an appreciable amount of our students. The opinion leaders in our courses are often in favour of open source software and Linux. This influences also the students using Windows. To improve the acceptance of the LOs in our courses it is good if they are usable under Linux.

Suggestions for Finalising the LOs

The analysis done gave several hints, that some formulations are not well understood by the students. Eventually it would be a good idea that all LOs are reviewed by students in a formal way to check what formulations are unclear for the students.

Interesting suggestions could be concluded from the analysis for the future production of LOs.

The analysis suggests producing LOs in such a way that they are easily adaptable to the specific content and logic of a course, i.e. to develop them in a form flexible to changes. The analysis further suggests using platform independent technologies to support a wider acceptance of the LOs.

A suggestion beyond the results of the survey is to lower barriers for the development of LOs by using open source tools for the development. This may be especially useful when collaborating with developing countries.

Ventspils University College

Course Curricula

Codewitz LOs are used in Ventspils University College in such courses:

- Programming (basics of programming language C++) – 2nd semester;
- Object Oriented Programming (advanced level of programming language C++ based on object oriented approach) – 3rd semester;
- Programming for WWW (acquirement of programming language JAVA) – 4th semester.

LOs are more useful, when students start studying programming – in the course Programming. In this course students acquire all basic themes in programming: basics, operators, loops and selections, arrays, functions, recursions, pointers and so on. As well as acquire something of basic idea of object oriented programming: classes and structures. In the course Object Oriented Programming we use LOs about files and classes. In the course Programming for WWW it is not necessary to use LOs so much, because in 4th semester students have good knowledge base about programming.

Study Setup

All already mentioned courses are realized in the same level – one lecture and one practical class in the week. In the practical class students are divided in the smaller groups. Mainly LOs are used to manage students' independent work in the school and at home. However sometimes LOs are used in the lecture, when lecturer introduce students with new theme. It is very useful to use LOs in the lecture, when the new theme is about loops, arrays or functions. To stimulate students use LOs more often, in the virtual learning system Moodle together with lecture slides are placed links to the appropriate LOs.

Results

To evaluate LOs at Ventspils University College we asked 50 computer science students in the course "Programming" to express their opinion about positive and negative aspects of LOs. We have put the most mentioned answers in the table:

Positive aspects of LOs	Negative aspects of LOs
Help understand basics.	Difficult to understand terms in English.
Understandable and user friendly interface.	Difficult to keep track of programs' process.

It is possible to see, how program works step-by-step.	Shockwave needed.
Useful to use in practical classes.	When lecturer writes it down on blackboard, it is easier to follow what is going on.

Analysis

< Results analysed>

Suggestions for Finalising the LOs

To develop such template, where it is easier to modify LOs, for example, to change parameter values or rows of source code, to add some rows, to translate LOs in native language.
To develop LOs in native language.
To develop compiler where students can write and compile their own programs.