

Work In Progress - Transversal Competencies Contributions to Computer Science 1 Course

Inés Friss de Kereki

Universidad ORT Uruguay, kereki_i@ort.edu.uy

Abstract – A competence may be generically defined as the capacity of doing something and is more than just skills or knowledge. A transversal competence is one which can be re-used when shifting to another context. Our goal is to contribute to improve Computer Science 1 (CS1) course’s results by the systematic incorporation of activities based on transversal competencies. We designed a set of activities to enhance the development of transversal competencies, specially planning and time management, team work and communication skills. Hypothetically, we propose that including this kind of activities helps to obtain better results in the course. An initial experimentation was done in 2010. The preliminary results show a remarkable positive difference in student’s results. In the selected group the fail rate was only 12% (3 of 25 students). In other groups, the fail rate was 32% (8 of 25 students) and 40% (8 of 20 students). In previous years, this rate was approximately 40%.

Index Terms – Computer Science 1 (CS1), Object Oriented Programming, Skills, Transversal competencies

INTRODUCTION

Learning to program is hard and difficult, and often programming courses have high drop out rates [1][2]. Bennedsen and Caspersen [3] refer in their study of 63 institutions that the pass rate in introductory programming courses is approximately 67%, so 33% of the students fail the course for some reason (for instance, abort the course before the final exam, do not show up for the exam, or directly fail the course, so, the students do not complete the required tasks). Denning and McGettrick [4] refer that drop rates of 35% to 50% are common.

Our goal is to contribute to improve Computer Science 1 (CS1) course’s results by the systematic incorporation of activities based on transversal competencies. In this paper we include some definitions of “competence” and possible classifications. Also, a description of our CS1 course, the detailed proposal of including some specific tasks, the experimentation, results and conclusions are presented.

COMPETENCIES

A competence may be generically defined as the skill needed to do a particular job or, the ability to do what is required [5]. Competencies represent a dynamic combination of knowledge, understanding, skills and

abilities [6]. Sicilia [7] uses the term “competency” in the sense of an observable human capacity to perform some task or function than leads to a concrete objective that has a value in a work context. Tovar and Soto [8] define competence as the ability shown by the use of knowledge, technical, personal, social and methodological skills, that can lead to success in professional and academic environments.

Competencies can be distinguished in “subject” specific and “generic” ones (also called transversal or general). In particular, generic competencies are: instrumental (cognitive abilities, methodological abilities, technological abilities and linguistic abilities); interpersonal (social skills) and systemic competencies (abilities and skills related to whole systems). [9].

Sicilia [7] refers to transversal competencies as those that are not specific for some given particular context. They are considered an important element, as they are transferable to different situations makes them valuable for graduates. However, introducing transversal ability development activities in the course structure represents a problem for which little guidance is available.

Tovar and Soto [8] refer top ten core competencies in Computer Engineering. Some of them are: planning, coordinating and organizing, and oral and written communication. [10] includes leadership and teamwork. Fernández-Sanz [11] refers also that successful computing professionals will need personal skills like teamwork, proactivity, communications skills, creativity and autonomy.

CS1 COURSE

CS1 course at ORT Uruguay University is aimed at teaching programming using Object Oriented Programming (OOP). The course prepares the learner for constructing simple programs using the OOP paradigm. By the end of the semester, the student will be ready to analyze simple situations, to design possible solutions and to implement them with an OOP approach. The programming language used is Java. A brief description of the 15-week course is shown in Table I.

Each week has 4 hours of lectures and 2 hours of lab practice. There are 2 compulsory programming assignments (on computer, in two-student group) and an individual test, plus a final exam. The first project is worth a maximum of 25 points, the second one 30 and the test 45. If the student achieves 86 points or more, he or she passes the course and does not have to take a final examination. Between 70 and

85 points, the course is approved but the student must take the final examination. With less than 70 points the course is failed and must be retaken.

TABLE I
COURSE DESCRIPTION

Week	Topics
1-3	Variables, pseudo code, control structures
4	Classes and objects
5-7	Relations between classes: Association
8	Relations between classes: Inheritance
9-10	Relations between classes: Aggregation. Collections
11	Enumerations
12	Sorting and Searching
13-15	Advanced use of collections

PROPOSAL

Our purpose is to improve the students' result in the course. We designed a set of activities to enhance the development of some transversal competencies, selected as relevant from the bibliographical review. We concentrated on: planning and time management, team work and communication skills. The main activities are:

- **Diary:** we give to the students a detailed semester schedule with tasks and evaluations, and an additional spreadsheet as a diary. On it, we ask them to record each activity done out of class related to the course. The goal is to take into account how much time it is necessary to develop some tasks.
- **Self evaluation:** based on a set of sentences, each student should identify which are more familiar to him or her. The sentences are related to how they manage deadlines and expected quality of the work. The objective is to motivate introspection about attitudes and skills.
- **Solving an equation:** the task asks to develop a Java program to calculate the roots of a second grade equation. It may be developed individually or preferably with a partner. It includes pseudo code, Java code and testing. Some detailed information about the task is asked. It is related to estimate and contrast time and results and also to evaluate team work.
- **Choosing:** this activity focus on which features are relevant in order to choose another student for the group tasks. Some profiles are presented and students should point out the main items to take into account. They must choose one profile and explain their selection.
- **Milestone:** this activity asks to write down and communicate tasks advances, estimation of missing topics, frequency of meetings, and reports of problems. The notes are compared with his or her partner's notes.
- **Texts:** based on two software related texts written by two programmers, the students must identify all possible problems (grammatical errors, typo errors, presentation, etc).
- **Review:** after the first compulsory task, the student reports his or her personal experience of team work,

main difficulties and benefits, and a final evaluation of his or hew own initial work estimation. The focuses of each activity are presented in Table 2.

TABLE II
ACTIVITY FOCUS

Activity	Planning/Time	Team work	Communication
Diary	✓		
Self evaluation	✓		
Solving an equation	✓	✓	✓
Choosing		✓	✓
Milestone	✓	✓	✓
Texts			✓
Review	✓	✓	✓

We use one activity per week. After each activity, the results were discussed in class and strategies and recommendations were offered to the students.

EXPERIMENTATION AND RESULTS

One random group of 25 freshmen was selected to participate in the activities; other two random control groups (with 25 and 20 students) received the normal course. The material, orientation and evaluation were the same in all groups. The students were 18-21 years old. Hypothetically, we propose that including this kind of activities helps to obtain better results in the course.

An initial experimentation was done in 2010. The preliminary results show a remarkable positive difference in student's results. In the experimental group the fail rate was only 12% (3 of 25 students). In other groups, the fail rate was 32% (8 of 25 students) and 40% (8 of 20 students). The results are in Figure 1. In previous years, in similar courses at our University, the fail rate was approximately 40%.

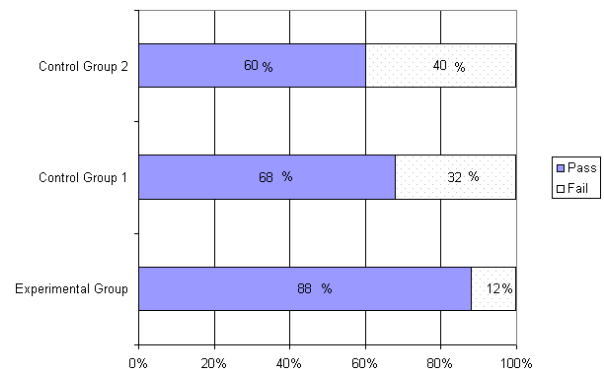


FIGURE 1
CS1 RESULTS: PASS VS FAIL

CONCLUSIONS AND FUTURE WORK

Transversal competencies are important and useful for the graduate. The incorporation of activities based on them was analyzed in the context of learning OOP. Our proposal is that including these kinds of activities contributes to improve CS1 student's results. The initial result shows a remarkable decrement of fail rate (from 30-40% to 12%). In 2011 we will replicate the experience, with some improvements, in order to validate or not the results.

REFERENCES

- [1] Robins, A., Rountree, J. and Rountree, N., "Learning and teaching programming: a review and discussion", *Computer Science Education*, Vol 13, No 2, 2003, pp. 137-172.
- [2] Kinnunen, P., Malmi, L., "Why Students Drop Out CS1 Course?", *ICER '06*, United Kingdom, September 2006, pp. 97-108.
- [3] Bennedsen, J., Caspersen, M., "Failure Rates in Introductory Programming", *Inroads The SIGCSE Bulletin*, Vol 39, No 2, June 2007, pp 32-36.
- [4] Denning, P., McGettrick, A., "Recentering Computer Science", *Communications of the ACM*, Vol 48, No 11, November 2005, pp. 15-19.
- [5] Longman Group, "Longman Dictionary of Contemporary English", England, 3rd edition, 1995.
- [6] González, J., Wagenaar, R., editors, *Universities' contribution to the Bologna Process, an introduction*, 2nd. edition, Spain, Publicaciones de la Universidad de Deusto, 2008.
- [7] Sicilia, M., "How Should Transversal Competence be Introduced in Computing Education?", *Inroads SIGCSE Bulletin*, Vol 41, No 4, December 2009, pp 95-98.
- [8] Tovar, E., Soto, O., "Are New Coming Computer Engineering Students Well Prepared to Begin Future Studies Programs Based on Competencies in the European Higher Education Area?", *Proc. of 39th ASEE/IEEE Frontiers in Education Conference*, Texas, 2009, pp. M2D-1-M2D-6.
- [9] Tuning Project, Final Report Pilot Project – Phase 1, Available online:
http://www.tuning.unideusto.org/tuningeu/index.php?option=com_docman&task=docclick&Itemid=59&bid=17&limitstart=0&limit=5. Last accessed: 3/3/2011
- [10] Vicent, L., Margalef, J., Montero, J. y Anguera, J. "Work in Progress-Developing Transversal Skills in an Online Theoretical Engineering Subject", *Proc. of 37th ASEE/IEEE Frontiers in Education Conference*, 2007, pp. F3B-11 F3B-12.
- [11] Fernández-Sanz, L. "Personal Skills for computing professional.", *Computer*, 2009, pp. 110-112.