

**objectKarel: A Didactic Microworld for Teaching Object-Oriented Programming**

S. Xinogalos  
Dept. of Applied Informatics, University of Macedonia  
156 Egnatia Str., Thessaloniki, Greece  
+30-2310-950613  
stelios@uom.gr

**Categories and Subject Descriptors:** K.3.2 [Computers and Education]: Computer and Information Science Education.

**General Terms:** Algorithms.

**Keywords:** object-oriented programming, microworld, structure editor, program animation, explanatory visualization.

In recent years, extensive research has been conducted on the difficulties encountered by novices when being taught programming. These studies have shown that an important factor of these difficulties is the teaching methodology used, which is based on: a) a general purpose programming language that is too big and too idiosyncratic, b) a professional programming environment for the chosen programming language that does not support students neither in understanding the semantics of the control structures and the flow of control nor in the process of debugging their programs, and c) a set of problems from the area of number and symbol processing that are far from the students’ everyday experiences and are not attractive to them. This ascertainment has lead many to develop new programming environments as well as alternative approaches to teaching programming. The most important approaches are Microworlds, Compilers with Improved Diagnostic Capabilities, Iconic Programming Languages and Program Animation.

Taking into account the results of the research, which are related to the students’ difficulties, we developed an educational environment for teaching object-oriented programming. Our programming environment - objectKarel - attempts to teach programming through a microworld and is based on the programming language of Karel++. The metaphor used is that of a world of robots and the main features of objectKarel are:

**e-lessons:** objectKarel incorporates a series of lessons that consist of theory and activities for supporting students in understanding the basic principles of object-oriented programming and the most common control structures.

**A special kind of structure editor:** writing a program is accomplished through a menu and dialog boxes. In contrast with typical structure editors, logic, semantical and some syntax errors are deliberately not prevented.

**Understandable and highly informative error messages.**

**Program animation – explanatory visualization:** students have three choices of executing a program – running the program, tracing through the program, executing it step by step. When students use the last two choices of executing programs, they are also presented with explanatory messages about the semantics of the command being executed, a feature known as *explanatory visualization.*

**Recordability of students’ actions:** the system autosaves the student’s program and its errors/warnings each time it is compiled.

objectKarel was tested and evaluated by 20 undergraduate students from the department of Applied Informatics of the University of Macedonia. All of them had failed in the 1st semester compulsory course “Programming”, which is based on the procedural programming paradigm and uses Pascal as the programming language. The students were separated into 10 pairs. Six lessons (2 hours each) were carried out, all of them at the laboratory, and the programming environment was used as the medium of instruction - occasionally the blackboard was used too.

In each one of the first 5 lessons we used the theory and the activities of the e-lessons in order to teach the following concepts: objects & classes, inheritance, polymorphism & overriding, conditional structures and repetitive structures. In each lesson the pairs developed programs without help from the teacher. The didactic aims of the lessons and the assignments, as well as the results will be presented. The 6th lesson was devoted to the assessment of students’ knowledge and the evaluation of objectKarel from the students. Students’ replies in relation to their difficulties and the advantages of the software will be presented.

The results of the empirical study that we carried out show that the programming environment offers invaluable help to both the teacher and the students in teaching and learning programming respectively. The ability to record students’ actions provides invaluable assistance to the teacher, so as to detect students’ conceptions about object-oriented programming and the techniques they use during program development. The structure editor solves the problem of focusing on the syntactic details and gives the chance to focus on understanding the concepts and principles of object-oriented programming, structure matters and developing problem-solving skills. The incorporation of the lessons (theory, activities) in the programming environment helps students during instruction and program development. Step by step execution, tracing through the program and explanatory visualization helps students understand flow of control and the semantics of control structures, and helps them overcome their difficulties and misconceptions. Finally, the pilot use of objectKarel showed that it is possible to teach object-oriented programming in a short period of time with specially designed educational environments, while this is impossible with the usual professional programming environments.