

# Educational Matters

Hans-Jörg Kreowski

September 1999

A key question of educational matters – and hence of this column – concerns anything that may be titled “How to teach  $x$ ” or “Teaching  $x$  at  $y$ ”. Over the years, there were various contributions of this kind on a variety of subjects ranging mainly from theoretical computer science to programming. The following note by Franch, Gabarró, Gómez, Vázquez, and Vázquez continues this series. The authors report on their experiences in running a newly developed pilot course on *Introductory Programming* using Java. If one looks around, there is a surprisingly large number of books on the market which introduce to programming (may it be related to a particular programming language or not). One encounters quite a variety of disparate views on the subject, and the many attempts have not much in common. The computer science community seems to lack a common understanding of the fundamentals and essentials of the field. Hence it is quite important to discuss the matter and every contribution is welcome. I hope that the following peculiar and controversial report will provoke many reactions. In any case, agreeing comments, contradictory statements, and reports on similar experiments (as well as other contributions to the column) should be sent to

Prof. Dr. Hans-Jörg Kreowski  
Fachbereich Mathematik und Informatik  
Universität Bremen  
Postfach 33 04 40  
D-28334 Bremen  
E-Mail: [kreo@informatik.uni-bremen.de](mailto:kreo@informatik.uni-bremen.de)  
Fax: ++49-421-218-4322

# An Introductory Programming pilot course using Java. \*

X. Franch<sup>1</sup>, J. Gabarró<sup>1</sup>, and A. Gómez<sup>2</sup>, A. Vázquez<sup>1</sup>, J. Vázquez<sup>1</sup>

<sup>1</sup> Dep. LSI, Universitat Politècnica de Catalunya  
Mòduls C5–C6, C/ Jordi Girona,  
1-3, E-Barcelona 08034, Spain.  
e-mail: {franch, gabarro, jvazquez}@lsi.upc.es

<sup>2</sup> Laboratori de Càlcul de la FIB Mòdul B6, C/ Jordi Girona,  
1-3, E-Barcelona 08034, Spain.  
e-mail: antonia@fb.upc.es

**Abstract.** A new experience in programming education is explained. It has taken place at the Facultat d'Informàtica de Barcelona (FIB) in a course named *Iniciació a la Programació* (Introductory Programming), also nicknamed as *IniPro*. The main goal has been to achieve a higher degree of motivation in the students. We've used for the first time Java as programming language. This fact has allowed us to teach not only what students “should learn” (basic algorithms and program design methods), but also what students “want to learn” (windows, menus, mouse). Although the work performed by the students during the laboratory sessions has been highly valued, they have also been able (and encouraged) to make a part of their work at home by means of Internet connection, too.

*Keywords:* Introductory Programming, IniPro, windows, mouse, Java, ftp, telnet, social impact.

*IniPro beyond 2000.* The role of Computer Engineering in the end of this century is complex [3]. Teaching an Introductory Programming course, known as *IniPro* at the Facultat d'Informàtica de Barcelona, is not easy. Often, the students get confused especially during the first weeks of the course: they expect to learn program design, but they feel that what they learn is far away from the kind of applications they are used to see and touch, and this fact eventually leads to the student's disinterestedness. This situation is critical as an increase in disinterestedness results in a decrease in academic profit. A pilot course has been designed to avoid this phenomenon. Our goal has been to teach not only the items that students should learn (programming languages constructs, formal stuff, basic algorithms and elementary program design) but also what they like to learn (windows programs, buttons, menus and mouse). The degree of interaction of both facets has been the main challenge of the course.

*Course contents.* The objective of the course is to introduce the students to *Object-Oriented* Programming using *Java*. Compared to the previous *IniPro* program, the following changes have been made:

---

\* This work has been partially supported by ESPRIT LTR Project no. 20244–ALCOM–IT, DGICYT under grant PB95-0787 (project KOALA), CICYT Project TIC97-1475-CE, CICYT Project TIC97-1158, CIRIT 1997SGR-00366, and by PR98-11 of the UPC.

- First, the focus has been changed from imperative programming languages to object-oriented ones. This change has mainly an impact on the transition from variables to objects, and on the appearance of classes and hierarchies (these last ones, in a very limited way); other possible consequences are avoided because the previous focus of the course had been already somehow object-oriented. Also, as part of this process, some new features (remarkably creation and destruction of objects) have arisen in our algorithmic (language-independent) notation.
- As part of the methodology, we have introduced the students to the HTML language and Internet using Netscape Communicator, as well as telnet and ftp applications (from a user point of view). These tools have allowed them to work at home and to connect to FIB machines, to send or get information.
- As a milestone in our motivation goal, we have presented the means to program a Graphic User Interface (windows, buttons, menus and mouse) from the first weeks of the course. To be more precise, we've taught a subset of Java's AWT package explaining very intuitive classes, such as Button or TextField.
- Algorithms on sequences play a central role in the course. As a particular kind of sequence, we have shown an integrated file processing system, that is, the InputStream, OutputStream, Read and Write classes in Java.
- The stepwise refinement [12] in programming has been replaced by the Object-Oriented Design [9].

*Teaching.* Besides the theory and exercise solving sessions, the laboratory sessions have been essential. In laboratory, there has been a redefinition of the roles of teachers and students, giving the students the main role. In each session there has been a *Laboratory Paper* specifying the work to do. The students, grouped in pairs, have had to solve these exercises. Teachers have had, deliberately, a secondary role, just answering questions and encouraging students. This rule has helped students to loose their fears of programming and supported them to develop initiative.

*Training the teachers.* A precondition of designing a pilot course is an acceptable knowledge of the concepts one wishes to teach. This is not easy in Computer Engineering, due to the continuous and dizzy changes it has. It's important to mention the effort of the UPC to introduce Internet [7]. Our first contact with *Java* was in Euro-Par 97 [11]. The project ESPRIT-LTR-2044-ALCOM-IT, paid the attendance. Another project, KOALA, DGICYT PB95-0787 paid tintin, the PC on which the laboratory was prepared. The UPC paid the attendance to [5]. We would like to emphasize the deep relationship between teaching and research.

*Basic teaching material.* To teach Emacs and Linux we have used [10]. To introduce Internet and WEB pages we have used [5, 7, 10]. Finally, to teach *Java* we have used [1, 2, 11].

*Evaluation.* The evaluation of this experience has been highly positive, considering both the academic results and the course atmosphere. With respect to the first point, the

results have been a bit higher than in the traditional groups, although the differences are not significant enough; in fact, the results mainly show the feasibility of the experience. With respect to the atmosphere, it is important to remark the high degree of assistance of the students of the pilot course to the lessons.

However, a few last remarks should be made. Some colleagues have asked “Have the students learned more stuff?”. We guess not. The students have learned the same as former students had learned in the past, but the new ones have learned it by means of a new methodology and modern tools. This “modernity” has allowed the University to approach the non-academic world.

Another important aspect is the evaluation of the contents of the course. In addition to a lot of small problems, two main questions have been faced. First, working with pre and post conditions in the presence of objects. Second, doing object-oriented design with the kind of exercises in the student’s exercise lists. Both of them have been solved in a reasonable way.

*Social impact.* This pilot course has had an important impact. One of the main Spanish newspapers, “El País” [8], has reported on it in a positive way and highlighted the fact that, this time, the University has been ahead the business environment in the use of Java. Even “l’Oasi” [6], the magazine edited by the students of the FIB and often very critical of the teachers’ actions, has considered the IniPro experience as positive, too. In both of them the updating and modernity of the contents has been remarked.

*Extending to the whole IniPro.* During the second semester, a second pilot group has been added. As the supporting material already exists, more time is being invested in improving the experience and preparing the total change, which will be made in the next century (year 2000). Also, other programming courses are currently being redesigned to link properly with IniPro.

*Remark.* A preliminary version of this work (written in Catalan) has appeared in [4].

*Special thanks.* To M. Colillas for her support in the laboratory sessions and the lot of times she had to adjust the paper of the Pipjava printer. To J. Osorio for sharing her knowledge in Java, solving the doubts as nobody else could do.

## References

1. K. Arnold and J. Gosling. *The Java Programming Language, Second Edition*. Addison-Wesley, 1998.
2. M. Campione and K. Walrath. *The Java Tutorial, second edition*. Addison Wesley, 1998. <http://java.sun.com/docs/books/tutorial/>
3. P. Denning and R. Metcalfe. *Beyond Calculation. The next fifty years of computing*. Copernicus, 1997.
4. X. Franch, J. Gabarró, A. Gómez, A. Vázquez, and J. Vázquez. Curs pilot d’iniciació a la programació, IniPro, usant Java a la FIB. In *Les tecnologies de la informació i les comunicacions en l’educació a la UPC*, pages 63–64. UPC, 1999.
5. M. Lluch. Programació amb CGIscripts i JAVA, 1998. ICT.
6. L’OASI. Esos recuerdos... Year 5, number 12, Facultat d’Informàtica de Barcelona, page 19.

7. J.L. Montero, A. Navarro, and A. Villar. Curs d'introducció i cerca d'informació a Internet, 1998. Organitzed by: Vicerectorat de Personal Acadèmic, Gabinet de Recursos Humans. Network information: <http://www.upc.es/docs/> .
8. EL PAIS. La Universidad se adelanta al mundo de los negocios. october 1 , 1998, ciber@pais page 5.
9. B. Stroustrup. What is object-oriented programming? *IEEE Software*, pages 10–20, May 1988.
10. Laboratori de Càlcul UPC-FIB. Laboratoris Docents, Quadrimestre Zero: Entorn Unix i Xterminals, Nocions Bàsiques de Xarxes i Protocols de Comunicacions, 1997.
11. M. Wirsing and A. Knapp. Concurrency in Java: Programming techniques and semantics, 1997. Tutorial given in Euro-Par 97, Passau.
12. N. Wirth. Program development by stepwise refinement. *CACM*, 14(4):221–227, april 1971.