

Diploma thesis:
Teaching introductory programming with the
Inverted Curriculum approach

Michela Pedroni (pedronim@student.ethz.ch)
Supervising Assistant: Bernd Schoeller
Supervising Professor: Bertrand Meyer

June 6, 2003

Part I

Project Description

1 Overview

The increasing number of computer science students, their wide variety of prerequisites and previous experiences, as well as the goal to prepare them for the realities they will encounter in the industry, makes the design of an introductory course on programming a challenging task. A new approach, known as the Inverted Curriculum, will be used for the introductory programming course held the first time in winter 2003/2004 at the ETH Zurich. The approach - also known as "progressive opening of black boxes" - lets the students step-by-step grow from consumers to producers of a library that is especially developed for this purpose. This is done by designing the exercises in a way that first, the students' applications solely act as clients to the components of the library (using them as black boxes). Then gradually the students are encouraged to plunge deeper into the library code by requesting modifications and extensions of the library as exercise tasks. This approach shows the benefits of abstraction through reuse by taking full advantage of reusable libraries available in a good object-oriented environment and therefore introduces important concepts and principles of software engineering right from the start.

2 Scope of the work

The diploma thesis is composed of two parts: First, a theoretical review on teaching introductory programming will be written. This includes

- the goals and challenges of designing an introductory programming course
- a survey on selected approaches comprising their advantages, drawbacks and experiences already made
- advantages, drawbacks and experiences with the Inverted Curriculum approach
- thoughts about which programming paradigm (e.g. imperative, functional, object-oriented) should be used to learn programming
- general problems of choosing a programming language for the introductory course
- Eiffel as a "first" programming languages including advantages and possible problems
- a description of my own experience with the introductory programming course

In the second phase of the diploma thesis acting as an active critic to the new programming curriculum at the ETH forms the main job. This means that on the basis of the review the overall setup and the progression of the ongoing course preparation will be revised about its qualities, drawbacks, and problems that might occur. If possible, solutions to the encountered problems will be given and implemented. The overall setup includes

- the proposed programming environment (including the installation process and the overall usability)
- the availability of online-help/manuals/tutorials
- the library on which the course will be based
- the organization of the course

At every step during the preparation of the course the following points will be inspected:

- the text book
- the exercises
- the solutions to the exercises

Within the exercises and their solutions the task will be to assist in their design and production as well as revising them.

The ultimate goal of this diploma project is not to court and butter up the approach that will be taken at the ETH. On the contrary, by first writing a review about different approaches with special emphasis on the Inverted Curriculum approach, a general overview of advantages and drawbacks will be given and therefore the sensibility for possible problems will be strengthened. In the second phase, the gained knowledge will directly be applied to the introductory programming course at ETH and hopefully helps to clear away the deepest pitfalls for future students.

3 Intended result

3.1 Report

The final report will be a documentation of all the work that was performed during the thesis. It will be composed of two main parts:

1. Theoretical review
2. Evaluation of the introductory programming course

For more information see 2.

3.2 Practical work

1. Design of appropriate exercises and solutions
2. Elimination of any problems if possible (for example production of a recovery script for the Eiffel environment, if the need arises)

The dimension of the practical work (especially for 2.) highly depends on the evaluation of the course setup and progression and can therefore not be determined at this point of time.

Part II

Background material

4 Reading list

Learning psychology: [EDEL00] and [HEAP].

Inverted Curriculum: [MEY97, LAS97, COH91-2, MEY01].

Teaching languages: [KÖL99-2, KÖL95, KÖL99-3, MCI96, KÖL99-1].

Eiffel: [MEY97, KÖL99-2, KÖL95, NOR92, NOR95, KÖL99-1].

Object-oriented teaching: [MEY97, KÖL99-2, NOR92, WAZ00, MEY02].

And many more...

Part III

Project Management

5 Objectives and priorities

Being a student that had a very frustrating first encounter with programming, my personal objective is to do everything possible to keep future students from having to suffer from the same. In this context it is important to note that the theoretical survey is not what I see as having the highest priority or as being the part that makes my diploma thesis valuable. It is an important part, but can also be seen as a preparation for the practical part, where the achievements will heavily influence the work that will be done.

6 Criteria for success

6.1 Report

The final report should show the overall work that was done during the project period. It will include a theoretical survey that critically looks at different approaches to teaching introductory programming, and a more distinct course-related part that shows the progression and adaptations that take place during the preparation of the introductory programming course at the ETH.

6.2 Practical work

The exercises and solutions should be designed in a way that they clearly represent the knowledge gained from the theoretical review. They should be at the same time interesting, challenging, but still feasible for even a student that attends the course with no prior knowledge about programming.

Part IV

Plan with milestones

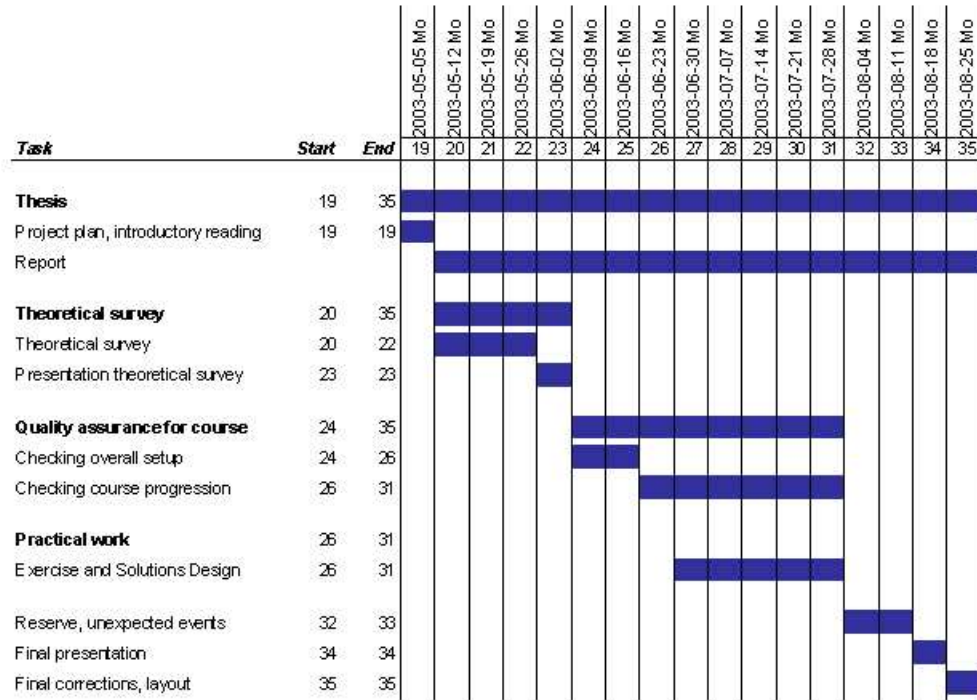
7 Project period

Project start: Friday, 05-02-2003 (week 18)

Project end: Monday, 09-01-2003 (week 36)

Total work time: 18 weeks + 2 days = 92 days

8 Tentative schedule



References

- [MEY97] Bertrand Meyer, "Object-oriented Software Construction", 2nd Edition, Prentice Hall, 1997.
- [MEY02] Bertrand Meyer, "Towards an O-O Curriculum", <http://archive.eiffel.com/doc/manuals/technology/curriculum/page.html> (April 2003).
- [MEY01] Bertrand Meyer, "Software Engineering in the Academy", IEEE Computer, August 2001.
- [GUZ02] Mark Guzdial, Elliot Soloway, "Teaching the Nintendo Generation to Program", Communications of the ACM, April 2002/Vol. 45, No. 4.
- [KÖL99-1] Michael Kölling, "The Problem of Teaching Object-Oriented Programming, Part 2: Environments," Journal of Object-Oriented Programming, 11(9): 6-12, 1999.

- [KÖL99-2] Michael Kölling, “The Problem of Teaching Object-Oriented Programming, Part 1: Languages,” *Journal of Object-Oriented Programming*, 11(8): 8-15, 1999.
- [KÖL95] Michael Kölling, Bett Koch and John Rosenberg, “Requirements for a First Year Object-Oriented Teaching Language”, *Basser Department of Computer Science, University of Sydney, Australia, ACM SIGCSE Bulletin, ACM, Nashville*, 173-177, March 1995.
- [KÖL99-3] Michael Kölling, “The design of an object-oriented environment and language for teaching”, *PhD Thesis, Basser Department of Computer Science, University of Sydney*, 1999.
- [NOR92] Kurt Normark, “Experience with Object-oriented Programming and Eiffel in an Introductory Programming Course”, March 1992.
- [NOR95] Kurt Normark, “An Evaluation of Eiffel as the first Object-oriented Programming language in the CS Curriculum”, No. R-95-2017, *Department of Mathematics and Computer Science, Institute of Electronic Systems, Aalborg University*, June 1995.
- [MCI96] Linda McIver, Damian Conway, “Seven Deadly Sins of Introductory Programming Language Design”, *Proc. Software Engineering: Education and Practice (SE:E&P’96)*, *University of Otago, Dunedin, NZ*, pp. 309 -316, *IEEE Computer Society*, 1996.
- [CAM03] William Campbell, Ethan Bolker, “Teaching Programming by Immersion, Reading and Writing”, *Department of Computer Science, University of Massachusetts Boston, ASEE/IEEE Frontiers in Education Conference*, 2002.
- [LAS97] Anita J. La Salle, “An Inverted Computing Curriculum:Preparing Graduates to Build Quality Systems”, *Department of Computer Science and Information Systems, American University, Washington DC, ASEE/IEEE Frontiers in Education Conference*, 1997. <http://fie.engrng.pitt.edu/fie97/papers/1397.pdf>
- [HIL99] Thomas B. Hilburn, Donald J. Bagert, “A Software Engineering Curriculum Model”, *ASEE/IEEE Frontiers in Education Conference*, 1999. <http://fie.engrng.pitt.edu/fie99/papers/1185.pdf>
- [WGSEET02] Working Group on Software Engineering and Training (WGSEET), “Specification for an Introductory Software Engineering Curriculum”, submitted to CCSE, June 2002.<http://faculty.erau.edu/hilburn/se-educ/se-intro-curr.pdf> (May 2003).

- [WGSEET03] Carnegie Mellon, Software Engineering Institute, Working Group: Software Engineering Education & Training (WGSEET). webpage: <http://www.sei.cmu.edu/collaborating/ed/workgroup-ed.html> (May 2003).
- [HIL] Thomas Hilburn, Software Engineering Education, webpage: <http://faculty.erau.edu/hilburn/se-educ/> (May 2003).
- [WGSEET98] Susan A. Mengel, "Guidelines Proposal for Undergraduate Software Engineering Education", Proceedings of the Frontiers in Education Conference, Computer Press, November 1998. <http://fie.engrng.pitt.edu/fie96/papers/1307b.pdf> (May 2003).
- [WGSEET99] Donald J. Bagert, Thomas B. Hilburn, Greg Hislop, Michael Lutz, Michael McCracken, Susan Mengel, "Guidelines for Software Engineering Education", CMU/SEI-99-TR03, ESC-TR-99-002, 1999. <http://faculty.erau.edu/hilburn/se-educ/99tr032.pdf> (May 2003).
- [DAL03] Charlie Daly, John Waldron, "Introductory Programming, Problem Solving and Computer Assisted Assessment", The 6th International Computer Assisted Assessment Conference. Burleigh Court International Conference Centre, Loughborough University, UK pages 95-107 9-10th July 2002.
- [COH91-1] Bernard Cohen, "IT Education", National Economic Development Office, 1991.
- [COH91-2] Bernard Cohen, "The Inverted Curriculum", Report, National Economic Development Council, London 1991.
- [COH87] Bernard Cohen, "The Education of the Information Systems Engineer", Electronics and Power, March 1987.
- [HEAP] N. W. Heap, K. L. Kear, C. C. Bissel, "An Overview of ICT-based Assessment for Engineering Education", Telematics Department, Faculty of Technology, The Open University.
- [WAZ00] R. S. Wazlawick, A. C. Mariani, "The use of an actor and stage metaphor for introducing object-oriented programming", Proceeding of ICEUT 2000, Section 8, 2000. <http://www.ifip.or.at/con2000/iceut2000/iceut08-02.pdf> (May 2003).
- [EDEL00] W. Edelman, "Lernpsychologie" (in German), 6th Edition, BeltzPVU, 2000.