Foreword

Computer science is an emergent, scientific, and practical approach that deals with the theoretical foundations of information and computation, combined with techniques for their implementation and application. As a discipline, computer science ranges from theoretical studies of algorithm correctness and complexity to the practical issues of implementing computing systems.

Computing Sciences Accreditation Board (CSAB)—represented by the Association for Computing Machinery (ACM) and the IEEE Computer Society (IEEE-CS)—identifies four key areas to the discipline of computer science: theory of computation, algorithms and data structures, programming methodology and languages, and computer elements and architecture. In addition to these four areas, CSAB also identifies other important fields of computer science such as software engineering, artificial intelligence, computer networking and telecommunications, database systems, parallel and distributed computation, computer-human interaction, computer graphics, and operating systems.

Major growth in the European Union is expected to occur in services based on information, finance, communications technology, electronic commerce, mobile applications, and games. These types of applications demand computer science skills to strengthen the field as one of the most important in the new century, as well justifying its inclusion in the curricula of all levels of education in schools.

Despite the importance of computer science, high failure and dropout rates are very common in introductory programming courses in many education institutions worldwide. This situation affects mostly novice students, since those courses are usually placed at the beginning of the curricula. Many causes for the learning difficulties have already been identified, from subject complexity to teaching methodologies and pedagogical strategies. Other causes are also identified related with the student background knowledge.

Several approaches have been proposed in the literature, ranging from psychological studies to computer-based teaching tools. The introduction of specialised services to automate tasks traditionally made by teachers (for instance, evaluation) is one of the biggest trends, but its use is not yet widespread, primarily due to interoperability issues and the scarce number of available programming exercises. In addition, worldwide initiatives have also been appearing to try to motivate young people to learn to program computers, therefore reducing the gap regarding student background knowledge. Despite all these efforts, the situation remains mostly unchanged.

This book provides a valuable window on information for computer science teachers presenting different solutions to foster computer science education. These solutions include the use of new strategies and approaches for teaching computer programming and the use of new frameworks and e-learning tools to motivate novice students to break the barrier of learning computer programming.

The first section, titled "State of the Art," presents several reviews regarding the teaching and learning of programming, more precisely, pedagogical strategies for introductory programming courses, the use of open source networks in education, and the adoption of small-group versus competitive learning in computer science classrooms.

Follows a section, "Teaching Strategies," presents different perspectives on how teachers can foster computer science education. One such strategy is using the emergent concept of Massive Open Online Courses (MOOC) to facilitate the access, dissemination, and communication of knowledge using the Web as the main vehicle. Other trend strategies use simulation games as educational approaches to motivate students to learn by using video game design and game elements in learning environments. The goal is to maximise enjoyment and engagement through capturing the interest of learners and inspiring them to continue learning.

Finally, the third section focuses on "Frameworks and Tools" and presents several models to help with the computer programming teaching-learning process. Many of the solutions presented rely on specialised Web services responsible by the storage, dissemination, evaluation, and classification of the students' progress. These 24x7 services are essential for the 21st century novice students. One such service is the assessment system that allow students to practice programming, anytime and anywhere. This feature represents the cornerstone of computer programming learning, since often one of the major barriers to learning computing is related to the lack of monitoring and feedback from teachers outside of class periods. These automatic evaluators analyse the resolution of students and provide readable and relevant feedback guiding students in a more autonomous manner.

In conclusion, in my point of view, all the pedagogical and technological strategies presented in the book are excellent educational contributions for the challenging field of computer science, more precisely, for one of its main areas, computer programming.

Alberto Simões
Universidade do Minho, Portugal

Alberto Simões *is a PhD in Natural Language Processing affiliated with the Polytechnic Institute of Cávado and Ave (Portugal) and works as a researcher at the Centre for Humanistic Studies of the University of Minho. His research interests focus on parallel corpora alignment, probabilistic translation dictionaries, and bilingual terminology extraction. Some of his major publications are "NATools: A Statistical Word Aligner Workbench," in Procesamiento del Lenguaje Natural, 31 (2003), "Makefile: Parallel Dependency Specification Language," in Anne-Marie Kermarrec, Luc Bougé, and Thierry Priol, editors, Euro-Parl 2007, volume 4641, and "Portuguese English Word Alignment: Some Experiments," in LREC 2008 – The 6th Language and Resources Evaluation Conference, Marrakech (2008).*