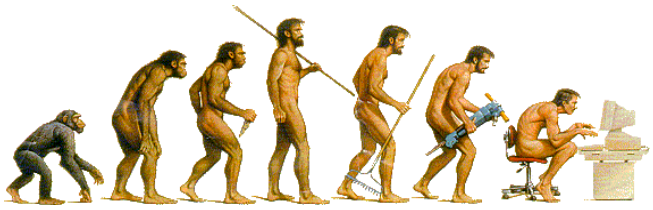
Computer Science



Contents

1. Course coordinator

Evgueni Smirnov (coordinator)
DKE, FHS, UM
St. Servaasklooster 39 (Room 1.001, 1st floor)
Tel. 043 388 2023
e-mail: smirnov@maastrichtuniversity.nl

2. Course material

The main textbook for the course is Schneider & Gersting (2015). The descriptions of the laboratory sessions and tasks are given in Lambert & Whaley (2010). In addition, supporting literature (online courses, hyperlinks, applets) can be found at the *Computer Science* site that is accessible through Blackboard (https://eleum.maastrichtuniversity.nl).

Schneider, M., and Gersting, J. (2015), Invitation to Computer Science, 5th Edition, Cengage Learning, ISBN: 978-1-305-07577-1

Lambert, K and Whaley, T. (2010), Invitation to Computer Science Laboratory Manual, 5th Edition, Cengage Learning , ISBN-10: 0324788630 ISBN-13: 9780324788631

3. Course schedule

Below you can find a rough tentative schedule. Please note that deviations that may occur during the course will be announced in the lectures in advance.

Week 1

Lecture *Introduction to Computer Science*
Lecture *Algorithm Discovery and Design*
Labs 1 and 2

Week 2

Lecture *Algorithm's Efficiency*
Lecture *Bin Numbers, Gates Logic Gates*
Labs 3 and 4

Week 3

Lecture *Computer Systems Organization*
Lecture *Assemblers and Assembly Languages*
Labs 5 and 6

Week 4

Lecture System software and Virtual Machines
Lecture Intro Java
Labs 7 and 8

Week 5

Lecture OO Java
Labs 9 and 10

Week 6
Lecture *Overview of Programming Languages*
Lecture *Overview of Programming Languages*
Labs 11 and 12

Week 7
Lecture AI
Exam Preparation

4. Introduction

Computer science is the study of algorithms on formal, hardware, linguistic, and application levels. This definition says that it is the task of the computer scientist to design and develop algorithms to solve a range of important problems. This process includes the following operations:

- Studying the behavior of the algorithms to determine whether they are correct and efficient (formal level),
- Designing and building computer systems that are able to execute algorithms (hardware level),
- Designing programming languages and translating algorithms into these languages so that they can be executed by computer systems (linguistic level), and
- Identifying important problems and designing correct and efficient software packages to solve these problems (application level).

Having the definition of computer science in mind, it is important from a practical point of view to understand why this scientific discipline is studied. The main reasons are:

- *Computers are everywhere.* Computer technology can be found in cars, in supermarkets, even in vending machines. More and more things are becoming computerized. The future offers many opportunities for those who have knowledge and skills in the area of computing.
- *Computer science is an interesting and exciting field.* It involves working at the cutting edge of technology. Robotics, computer graphics, electronic business, networking, the world wide web, and multimedia, are just a few of the hot topics in computer science.
- *Computer science offers diversity.* It can be applied not only in the fields of science and engineering but also in such diverse fields as commerce, psychology, medicine, art and music.
- *Many career opportunities.* The continuous growth of the computer industry allows computer science students to obtain financially rewarding and challenging careers with government bodies, computer companies (e.g., IBM, Siemens, Unisys), large companies and banks (e.g., ABN AMRO, Deutsche Bank), and software companies (e.g., CMG, Microsoft).

5. Aim of the course

The primary aim of the course is to provide an introduction to the fundamental concepts found throughout the field of computer science. As an overview of the discipline, the course covers a breadth of topics including algorithmic foundations of computer science; hardware issues such as number systems and computer architectures; and software issues such as operating systems, programming languages, compilers, networks, and human-computer interaction. The course is intended for beginners in computer science. Students

come to the University College Maastricht with a very wide range of backgrounds in computer science; this course seeks to provide a common foundation and unifying perspective.

6. Required background

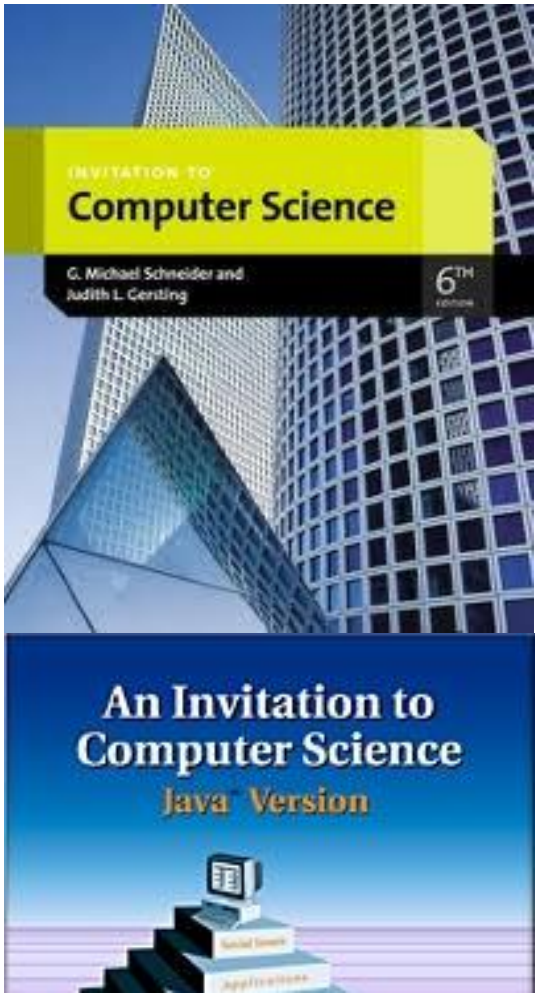
The course is a breadth-first introduction to the discipline of computer science. It assumes absolutely no background in computer science, programming, or mathematics.

7. Software environments

The main software environment used in the course is the software provided with the *Lab Manual* (that accompanies the main book of the course). The software can be downloaded from the website (http://www.cengage.com/) if you have bought the *Lab Manual*. For the Java laboratory sessions the Java compiler and run-time environment will be used. They can be freely downloaded from the Sun website (http://java.com). Additional software (e.g., Microsoft Word, Microsoft Excel, Microsoft Access, Microsoft Explorer) is available at the University College Maastricht.

8. Course book

The course book "*An Invitation to Computer Science: Java Version*" is an easy-to-read introduction to computer science. It provides an overview of the central topics studied in this field. The authors present these topics in the context of the big picture, using a six-layer hierarchy of abstractions. The hierarchy is based on the algorithmic foundations of computer science and works upward from low-level hardware concepts through virtual machine environments, languages, software, and application programs to the social issues raised by computer technology. Each layer in the hierarchy builds on ideas and concepts presented earlier. An accompanying lab manual provides exploratory lab activities.



9. Course outline

The course is based on the hierarchy of abstractions proposed in the main textbook "*An Invitation to Computer Science: Java Version*". It considers only the first five levels of the hierarchy. The levels are given below together with their main sub-topics.

Level 1. *The Algorithmic Foundations of Computer Science*

- Algorithm Discovery and Design
- Algorithm's Efficiency

Level 2. *The Hardware World*

- Binary Numbers, Boolean Logic, and Gates
- Computer System Organisation

Level 3. *The Virtual Machine*

- System Software and Virtual Machines

Level 4. *The Software World*

- Introduction to Java Programming Language
- Overview of Programming Languages

Level 5. *Applications*

- Spreadsheets
- Databases
- Computer Networks
  - The Internet
  - The World Wide Web

10. Course composition

The course consists of three components: lectures, laboratory sessions, and a final exam. The lectures will present the main theoretical and practical aspects of computer science. Each lecture is followed by a laboratory session. During the laboratory session students will be given tasks described in the *Lab Manual* (that accompanies the main textbook). The solutions of the tasks should be submitted individually after at most one week. The course ends with a final closed-book exam.

11. Evaluation

Students are evaluated according to their performances on:

- **Laboratory Tasks** (individual). There will be 13 laboratory tasks (each for each laboratory session). The maximum number of points if all the tasks are solved successfully is 10.
- **Final Exam** (individual). The maximum number of points for the exam is 10.

A student's overall score is based on the evaluations of the lab tasks, homework assignments, and final exam, in the following way:

Overall score = $0.4 \times$ score on lab tasks + $0.6 \times$ score on final exam

Please note that the students, who have missed the group meetings but not more than 30% of them, will be given a provisional overall grade. They will receive credits for the course only when they have successfully completed an additional assignment.