

Syllabus

Introduction to Computing

EP Course – WS 2009/10

Teacher	Prof. Dr. Bodo Bartning
Email	bartning@nwt.fho-emen.de or info@bartning.org
Home page	www.bartning.org or (if not working) www.spot.fho-emen.de/hp/bartning
Office hours	before and after the lectures and by appointment
Telephone	best with private number (0 49 21) 95 40 95
Answ. machine	private at any time (0 49 21) 95 40 96
Current schedule	see link from the home page

Main page of this course: can be found via links from the home page

Course duration

This course is a course with 5 hours per week. It will last one semesters. About two thirds will be lecture, and one third computer lab.

Overview

In the short overview (ch. 1) you will learn the basic structure of a computer system (hardware and software).

In ch. 2–5 you will get to know the three basic algorithm structures and how to use them to solve smaller problems.

Ch. 6–9 will show different kinds of solving greater problems, they are dependent on the view on the problem field. First you will get to know older problem solving methods (procedural, and—very briefly—modular), then a more modern way which means a very different view of the problem area (object oriented).

In both parts (ch. 2–5, and 6–9) there are two kinds of approach: language independent (ch. 2 and 6), after this the implementation in C++. Both parts are important: the first approach will be valid even if you or your company will not use C++ (any more), the second is important as an example how to implement ideas in a modern computer language.

Since now the course is shortened compared to the courses beginning before 2008, the last three chapters (ch. 10–12) cannot be rendered. So knowledge of them is not required for the examinations. Nevertheless, they are available in the internet version of the lecture notes (detail see last chapter of this syllabus). Ch. 10 provides some necessary items for advanced application of this language in object oriented programming (ch. 11). The two chapters (ch. 9 and 11) describe the essential points of object orienting programming with C++—including the chapters necessary for their comprehension. Ch. 12 expands the general C++ knowledge, especially about dealing with pointers. This is important for advanced C++/C programmers.

Course level and type, prerequisites

This introduction course does not have any prerequisites. Base knowledge how to use a PC (operating system Windows 2000/XP/Vista, editing text) will be explained very shortly at the beginning, also how to use a compiler.

Presentation style

During the lecture, questions and discussion are welcome. The lab is very important. Without attending the computer lab, the course is (nearly) useless. The lectures will provide comprehension for programming structures and techniques which are applied during the lab—this is a *necessary additional step!* The exercises for the lab will be posted on the internet in time (available via the main page of this course). Since I am restricted to give only 4 hours (contract), the remaining hour (as part of the exercises, also mandatory) will be offered by a tutor.

The course is *very time consuming* because learning the lectures' contents and (especially) doing the exercises at the computer will require additional time *outside the course*.

Course objectives

- Gaining base knowledge how to use a computer: operating system, editing, using a compiler;
- *language independent*: learning how to use algorithmic structures, gain an insight into procedural programming, getting to know objects and classes (introduction to object oriented programming);
- learning how to implement these structures in C++.

Course requirements

- one written examination (60 min), it will be offered at the end of this semester;
- computer program, presumably issued in the middle of December, due within four weeks, with subsequent representation and explanation at the computer.

Books

No textbook is required, but it may be helpful. You will get summary lecture notes (very short) with the most important topics of this lecture, containing ch. 1–9. You will also get a syntax summary about C++ which you are allowed to use during the examination—but not any lecture notes will be allowed.

If you want to have a textbook: there are many books about C++. Please read some paragraphs in the book before you buy it: the book's style shall be convenient for you. **Important:** the book should teach C++ independently of the language C, or you will have to learn C, too.

Three books (as examples):

- Bjarne Stroustrup, *The C++ Programming Language*, 3rd edition, Addison-Wesley 1997; German translation: *Die C++ Programmiersprache*, 3. Auflage, Addison-Wesley 1998
Remark: If—and only if—you intend to use C++ intensively (esp. later on), this book will be a must, for Stroustrup is the inventor of C++.
- Ulrich Breymann, *C++, Eine Einführung*, 8. Auflage, Hanser, a German introduction, with Gnu compiler for various platforms on CD—a more didactic introduction.
- Brian W. Kernighan, Dennis M. Ritchie, *The C programming language*, 2nd edition, Prentice Hall, 1988; German translation: Hanser Verlag—*for learning C as a base for C++*.

Lecture notes, contents

You will get a printed copy of the core lecture notes (ch. 1–9) at the beginning of the course.

You can find the most recent version of the (complete) lecture notes also as a PDF file on the internet (and a German translation if needed). You will get it via a link from the main page of this course. On the main page, you will also find links to the exercise files and to further example files belonging to the course.

For reading the lecture notes file, you have to start the Acrobat Reader on the computers in our exercise room. If you want to read it on your own computer, you need this reader, too (available as free ware).

Important: since the lectures notes are very much compressed, you cannot use the notes to learn the lectures' contents for the first time, they will be useful for *re-learning* and for looking up.

Hereafter, you find the chapter headlines of the (complete) lecture notes without page numbers.

0 Introductory remarks

- 0.1 Legend, language remarks, picture symbols
- 0.2 Lecture notes
- 0.3 Advices for applying older C++ compilers

I Introduction to Computing (course)**1 Computer systems (short overview)**

- 1.0 Overview
- 1.1 Basic terms
- 1.2 Computer structure, hardware
- 1.3 Software, operating systems
- 1.4 Numbers, characters

**2 Algorithm structures, operators
(language independent reflection)**

- 2.0 Overview
- 2.1 Introduction to algorithms
- 2.2 Sequence
- 2.3 Selection
- 2.4 Iteration
- 2.5 Nesting the three algorithmic structures
- 2.6 Operators, logical combinations

3 First steps with C++

- 3.0 Overview
- 3.1 The first programs
- 3.2 Simple data types, strings, operators
- 3.3 Expressions, side effects

4 Algorithm structures in C++

- 4.0 Overview
- 4.1 Sequence, scope within blocks
- 4.2 Boolean expressions, data type `bool`
- 4.3 Selection
- 4.4 Iteration
- 4.5 The *condition* in control structures, scopes with more recent compilers
- 4.6 Example
- 4.7 Recommendations for naming and layout, symbolic constants

5 Input and output, array and string type, supplements

- 5.0 Overview
- 5.1 Overview of operators
- 5.2 Standard streams, error handling
- 5.3 Handling text files
- 5.4 Arrays
- 5.5 Strings (C strings)
- 5.6 Overview of types
- 5.7 Preprocessor

**6 Creating construct units,
problem of distinction between hidden and accessible
(language-independent reflection)**

- 6.0 Overview
- 6.1 Design of systems: construct units and hiding principle
- 6.2 Procedural programming

6.3	Storage classes, modular programming	
6.4	Object oriented programming	
7	Procedural programming	
7.0	Overview	
7.1	Function definition and function call, scope within blocks	
7.2	Function declaration	
7.3	Reference type	
7.4	Parameter passing modes value and reference	
7.5	Global and local names	
7.6	Overload of function names, default arguments	
7.7	Recursive functions	
7.8	Function guidelines	
8	Storage classes, modular programming	
8.0	Overview	
8.1	Static and automatic storages class	
8.2	Modular programming: splitting into several translation units	
9	Object oriented programming: encapsulation of data and functions with access control	
9.0	Overview	
9.1	Classes and objects	
9.2	Constructor and destructor	
9.3	Examples	

II Supplements, not included in the course

10	Some C++ supplements, pointer type, free store	
10.0	Overview	
10.1	Symbolic constants, macros and <code>inline</code> functions	
10.2	Data type pointer, type interpretation, array and pointer	
10.3	Free store	
11	Object orientation: supplements, inheritance, polymorphism, static class members	
11.0	Overview	
11.1	Classes and objects: supplements	
11.2	Operator overloading	
11.3	Single inheritance	
11.4	Polymorphism	
11.5	Static class members	
12	Operators, types, supplements to pointers, binary files	
12.0	Overview	
12.1	Operators	
12.2	Enumeration type, type conversion	
12.3	Pointer arithmetic	
12.4	Pointer as function parameters	
12.5	Applications	
12.6	Dealing with binary files	
12.7	Multidimensional arrays and pointers, command line parameters	
12.8	Generic pointer, C library functions	