

Introduction to C programming

The programming language for this course is C.

The course begins with an introduction to the world of programming, familiarizing with programming terms, for example. C language is brought in in the second chapter and starting with chapter 9 students are introduced to more advanced subjects.

Introduction to C programming includes 30 programming exercises and 66 multiple-choice questions. Finishing this course takes about 90-150 hours.

Table of contents:

1 A quick introduction to the world of programming 1.1 What is programming? 1.2 Defining and solving a problem 1.3 Unambiguity of an algorithm (better instructions for Tahvo)	2 From travelling algorithms to the world of computing 2.1 From an algorithm to source code 2.2 First C code: calculating and printing the sum of two numbers Summary of Section 2.2 2.3 Comments in the C language 2.4 What kinds of programs will I learn to write? 2.5 From source code to a functional program	3 Variables in the C language 3.1 How does computer memory operate? 3.2 Using and naming variables 3.3 Numerical variable types in the C language 3.4 Variable declaration 3.5 Variable initialisation 3.6 The printf() function 3.7 The scanf() function
4 Program statements, expressions and operators 4.1 Program statements in the C language 4.2 Expressions 4.3 Operators	5 If statements add power to your programs 5.1 What is an if statement? 5.2 switch() - select a path	6 Loop structures in the C language: the lazy programmer's dream 6.1 for statements make dull jobs easier 6.2 The while statement repeats until... 6.3 Do first, test afterwards: do-while
7 Functions - a route to more extensive programs 7.1 A simple function example 7.2 Programming your own functions 7.3 Functions and structured programming 7.4 Preprocessor, library files and standard functions	8 Numerical arrays in the C language 8.1 One-dimensional arrays 8.2 Why arrays? 8.3 Multidimensional arrays 8.4 Example of handling a two-dimensional array	9 Handling files in the C language 9.1 File names 9.2 Opening a file 9.3 Writing to a file - fprintf() 9.4 Reading from a file
10 Handling characters and strings in the C language 10.1 Character variable char 10.2 From a character variable to strings and arrays	11 Simple data structures in the C language 11.1 Declaration and definition of structures 11.2 Members of a structure 11.3 An example of the usefulness of structures	12 Pointers in the C language 12.1 More on computer memory operation 12.2 Address and pointer 12.3 Pointers and "basic variables" 12.4 Summary of Chapter 12.
13 Arrays and pointers 13.1 The name of an array is a pointer 13.2 Storage of arrays in memory 13.3 Strings and pointers 13.4 Passing an array to a function using a pointer 13.5 Files and pointers 13.6 Pointers and operators		