

## Introduction to C++ programming

The programming language for this course is C++.

The course begins with the basic concepts of computer programming. The C++ language fundamentals starts from the second chapter and in the last chapters the basics of object-oriented programming are introduced.

The course includes 11 theory chapters, including x pages. *Introduction to C++ programming* includes 23 programming exercises and 51 multiple-choice questions. Finishing this course takes about 60-120 hours.

### Table of contents:

<p><b>1 Countdown to the world of programming</b></p> <p>1.1 What does “programming” mean 1.2 Defining a problem and solving it 1.3. Implementing the solution into an action plan 1.4 Unambiguity of an algorithm (enhanced instructions to Barney)</p>	<p><b>2 From travelling algorithms to the world of computing</b></p> <p>2.1. From algorithm to source code 2.2 Your first C++ coding: calculating and printing the square of a number 2.3. Commenting in C++ 2.4. Formatting the source code 2.5. What kind of programs will I learn to do? 2.6. From source code to a real application</p>	<p><b>3 Variables in C++</b></p> <p>3.1. How does the computer memory work? 3.2. Using and naming variables 3.3 Primitive datatypes in C++ 3.4 Defining variables 3.5 Initializing variables 3.6 Handling strings 3.7 More on printing: concatenated print statements and formatting</p>
<p><b>4 Programming statements, expressions and operators</b></p> <p>4.2 C++ Programming statements 4.2 Expressions 4.3 Operators</p>	<p><b>5 Add some kick with if-statement</b></p> <p>5.1 What is an if-statement? 5.2 Switch() - choose your path</p>	<p><b>6 C++ loop structures: a lazy programmer's dream come true</b></p> <p>6.1 Make boring things easier, use for-statement 6.2 While-statement repeats until... 6.3 Do first, then test: do .. while</p>
<p><b>7 Functions – the gateway to larger programs</b></p> <p>7.1 Function - a simple example 7.2 Programming customized functions</p>	<p><b>8 Numerical arrays in C++</b></p> <p>8.1 One-dimensional arrays 8.2 Why to use arrays? 8.3 Multidimensional arrays 8.4 An example of manipulating a two-dimensional array</p>	<p><b>9 Object-oriented programming – the basics</b></p> <p>9.1 Classes and objects 9.2 Functionality for the class - tricks for the dog 9.3 Class constructor 9.4 From class to object</p>
<p><b>10 More about objects - inheritance</b></p> <p>10.1 Inheritance in C++ 10.2 Testing the Dog class 10.3 The bliss of object-oriented programming</p>	<p><b>11 Basics of file handling</b></p> <p>11.1 Concepts 11.2 Constant input stream cin and constant output stream cout 11.3 File objects 11.4 Reading a file - using ifstream class 11.5 Writing to a file - ofstream-class 11.6 Reading and writing - fstream-class 11.7 Reading different types of files</p>	