# The “C++” programming language course syllabus – associate level

Course description

The course fully covers the basics of programming in the “C++” programming language and presents the fundamental notions and techniques used in object-oriented programming. It starts with universal basics, not relaying on object concepts and gradually extends to advanced issues observed in the objective approach.

Prerequisite Courses

The “C” programming language course – associate level (suggested)

Learning objectives

To familiarize the trainee with the universal concepts of computer programming.

To present the syntax and semantics of the “C++” language as well as basic data types offered by the language

To discuss the principles of the object-oriented model and its implementation in the “C++” language

To demonstrate the means useful in resolving typical implementation problems with the help of standard “C++” language libraries

Course outline

Introduction to compiling and software development

Basic scalar data types, operators, flow control, streamed input/output, conversions

Declaring, defining and invoking functions

Strings processing, exceptions handling, dealing with namespaces

Object-oriented approach and its vocabulary

Dealing with classes and objects

Defining overloaded operators

Introduction to STL

Chapters:

*Absolute basics*

machine and  high-level programming languages, compilation process

obtaining the machine code: compilation process

recommended readings

your first program

variable – why?

integers:  values, literals, operators

characters: values, literals, operators

dealing with streams and basic input/output operations

*Flow control and more data types*

how to control the flow of the program?

floating point types: values, literals, operators

more integral types: values and literals

loops and controlling the loop execution

logic, bitwise and arithmetic operators

*Functions*

functions: why do you need them?

declaring and invoking functions

side effects

different methods of passing parameters and their purpose

default parameters

inline functions

overloaded functions

*Accessing data and dealing with exceptions*

converting values of different types

strings: declarations, initializations, assignments

string as the example of an object: introducing methods and properties

namespaces: using and declaring

exception handling

*Fundamentals of the object-oriented approach*

class: what does it actually mean?

where do the objects come from?

class components

constructors

referring to objects

static members

classes and their friends

defining and overloading operators

*Class hierarchy*

base class, superclass, subclass

inheritance: how does it work?

types of inheritance

inheriting different class components

multiple inheritance

*Classes – continued*

polymorphism: the notion and the purpose

virtual methods: declaring and using

inheriting virtual methods

abstraction and abstract classes

*Exceptions – dealing with expected and unexpected problems*

what is *an exception*?

catching and throwing exceptions

different classes and hierarchy of exceptions

defining your own exceptions