Software Engineering

Moscow Institute of Physics and Technology

Author of the Course

PhD Ivan Sergeevich Makarov

- Graduate of the Moscow Institute of Physics and Technology
- Candidate of Technical Sciences in specialty 1.2.1
- Associate Professor and Course Lecturer at the MIPT
- Software Developer and AI Researcher since 2010
- Author of 20 publications in peer - reviewed scientific journals
- Author of 11 reports at international conferences

Currently, I am engaged in the design and development of various computing systems using concurrent and network programming, in particular, I am working on optimizations of infrastructure components of automated high - frequency crypto trading financial systems.

C++ Definition

C++ is a compiled general - purpose programming language based on weak static type system. This language supports multiple programming paradigms and provides both low - level and high - level features.

- The processor understands only low - level machine code
- The developer writes high - level source code
- The compiler translates the source code into machine code
- The types of all objects are known at compile time
- Various automatic implicit type conversions are allowed

C++ Evolution

- Originally developed as a set of the C language extensions
- Currently is an independent and full-fledged language
- Inherited components of Ada, Fortran, Simula and others
- Influenced Java, Go, Python and many other languages
- Has taken place in the market and has several competitors

The first commercial release of the C++ language was on 14/10/1985.

C++ Standards

- C++98 – fundamental standard
- C++03 – patch
- Technical Report 1 2007 and various Boost libraries
- C++11 – significant extensions
- C++14 – patch
- C++17 – patch
- C++20 – significant extensions
- C++23 – patch
- C++26 – the next standard under development

Additional features are provided by libraries such as Boost and Qt.

C++ Use Cases

- Operating systems and embedded software
- Highly loaded data processing servers
- Game engines and mathematical modeling
- High - frequency trading infrastructure
- Solutions for high - responsibility industries

The C++ language provides both low - level and high - level features.

Programming Paradigms

- Declarative programming – SQL and HTML
- Imperative programming – statements
- Procedural programming – subroutines
- Functional programming – Lisp, Erlang and Haskell
- Structured programming – sequences, selections and loops
- Object - oriented programming – classes
- Generic programming – templates
- Event - driven programming – events and callbacks
- Concurrent programming – threads, processes and networks

The same software can be implemented in many different paradigms.

Developer Tools

Tool	**Considered**	**Alternative**
GNU Linux distribution	Ubuntu	Fedora, Manjaro
Environment	Visual Studio Code	CLion
Toolset: Compiler	g++ from GCC	Clang
Toolset: Builder	CMake	Bazel
Toolset: Debugger	GDB, Valgrind	LLDB
Toolset: Profiler	Google.Benchmark	gperftools
Version control system	Git	no
Git client	SmartGit, IDE	GitHub Desktop
Project hosting service	GitHub	Bitbucket

References

- learncpp.com – basic educational materials
- cppreference.com – language reference
- boost.org – Boost libraries documentation
- github.com – open projects and libraries
- stackoverflow.com – questions and answers
- packtpub.com – books from developers

The list of all recommended books is available in my Google-sheet.