C++ Programming I

Fundamentals of Object Oriented C ++ Programming

C++ Programming
February 22, 2018

Dr. P. Arnold
Bern University of Applied Sciences

Welcome

Welcome to C++ Programming I

Time:

- Thursday 16:15h-18:00h Weekly
- Break: 17h
- 13 x 2 Lectures

Lecture Style:

- 50% Theory
- 50% Coding

Homepage:

`http://www.bme.master.unibe.ch/studies/curriculum/list_of_courses/c_programming_i/`

Course Material:

`https://ilias.unibe.ch/goto_ilias3_unibe_crs_1233454.html`

Exercises and Credits

Exercises

- **~10 Exercises**, when handed in on time: **15**%
- **2 written exams**, midterm exam **25**% and final exam **60**%
- Dates for midterm exam:
  1. 12.04.2018 (after Easter!)
  2. 19.04.2018
  3. 26.04.2018

Procedure

- Exercises are strongly recommended
- Submission of at least **7 exercises** is required for exam admission
- Time for exercise is **1 week**
- Exercises are discussed in the lecture

Content

FS2018 - Fundamentals of Object Oriented C++ Programming

1. **Welcome**
2. **Getting Started**, Compiler, IDE etc.
3. **Basics** - Refresher
4. **Functions**, Call-by-Value and Call-by-Reference
5. **Pointers and References**, Dynamic Memory
6. **Fundamentals of Object Oriented C++ Programming**
7. **Classes and Objects**, Constructor and Destructor
8. **Inheritance**
9. **Polymorphism** and Abstract Interfaces
10. **Operators** and Operator Overloading
11. **Templates** - Basics

Literature

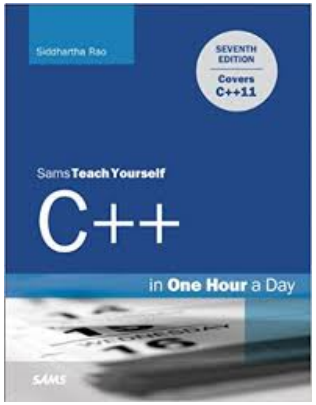
Text Books

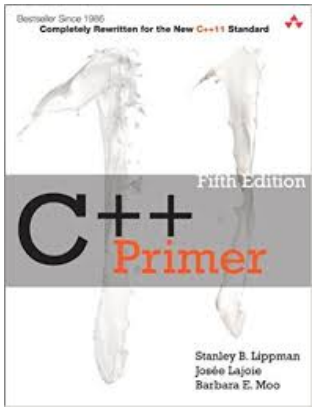
The course closely follows the following Literature:

- **Sams Teach Yourself, C++ in One Hour a Day** (8th Edition, 2017), Siddhartha Rao, ISBN-13: 978-0789757746

Comprehensive Reference book:

- **The C++ Programming Language** (4th Edition, 2015), Bjarne Stroustrup, ISBN/ISSN: 2244009029992
- **C++ Primer** (5th Edition, 2013), Stanley B. Lippman, ISBN-13: 978-0321714114





Literature

Links

Useful Links:

- `https://www.tutorialspoint.com/cplusplus/cpp_stl_tutorial.htm`
- `http://www.learncpp.com/`
- Training: `http://progressor.ti.bfh.ch`

Why C++

Programming Languages

Differences

Programming languages are differentiated in:

- Compiled vs. interpreted languages
  - C++ , C, Java, Pascal etc. vs. MATLAB, Python
- High level vs. assembler languages
  - Assembler (computer) languages are very primitive
  - Simple operations are divided into multiple steps
  - Language varies from computer to computer
  - High-level programming languages are human-readable and understood by the computer model (abstraction)
- Multipurpose vs. specific purpose languages
- Procedural, Object-oriented, function-oriented and logical languages
- Generic programming

Various modern and popular programming languages exist:

- Java, C#, Objective-C, Modula etc.

Common consensus:

- 'The' programming language for system programming is: C
- C has significant weaknesses. Biggest weakness: Missing 'Type safety'

Benefits of C++ :

- C++ adds the power of abstraction from high-level programming languages to C
  - Object-Oriented
  - Type Safety
  - Huge amount of tested and efficient libraries available $\rightarrow$ STL
- **High-level language for low-level problems**

Dr. P. Arnold

Bern University
of Applied Sciences

Have Fun !!!

Thank You