



Vietnamese - German University

# C++ PROGRAMMING

<b>Module:</b>	Software Engineering
<b>Module Number:</b>	13
<b>Semester:</b>	Summer 2019 (March 18 - 30)
<b>Lecturer:</b>	Dr. Thien Binh Nguyen
<b>Language:</b>	English
<b>Part of Curriculum:</b>	Mechatronics and Sensor Technology (M. Sc.)
<b>Workload:</b>	Contact time 44h, homework 46h
<b>Credit Points (ECTS):</b>	3
<b>Prerequisites:</b>	None
<b>Recommended Requirements:</b>	Basic programming knowledge
<b>Examination:</b>	Written examination (90 min) with marks
<b>Media:</b>	Lecture slides + personal laptop for in-class exercises

# Objectives

C++ Programming teaches the basics in C++ language including object-oriented programming (e.g., classes, inheritance, templates.) Algorithms for technical and mathematical problems are developed and implemented using C++.

Topics that are discussed in detail include the followings

- C++ Basics
- Variables, References, Pointers, and Arrays
- Functions
- Object-Oriented Programming: Classes
- Inheritance and Polymorphism
- Namespaces and Templates
- Debugging and Exception handling
- A Simple PDE Approximation

# Lecturer

Dr. Thien Binh Nguyen

Computational Engineering Study Program

Office: B.110 (VGU Binh Duong campus)

Email: [binh.nt@vgu.edu.vn](mailto:binh.nt@vgu.edu.vn)

# References

1. Capper, *Introducing C++ for Scientists, Engineers, and Mathematicians*, 2<sup>nd</sup> Edition, Springer, 2001

<https://www.springer.com/gp/book/9781852334888>

2. Pitt-Francis, and Whiteley, *Guide to Scientific Computing in C++*, Springer, 2012

<https://www.springer.com/la/book/9781447127369#otherversion=9781447127352>

3. C++ online courses and tutorials:

Learn Cpp: <https://www.learncpp.com/>

TutorialPoint: [https://www.tutorialspoint.com/cplusplus/cpp\\_overview.htm](https://www.tutorialspoint.com/cplusplus/cpp_overview.htm)

Geeks for Geeks: <https://www.geeksforgeeks.org/c-plus-plus/>

<https://math.nist.gov/~RPozo/c++class/>

<https://ece.uwaterloo.ca/~dwharder/aads/Tutorial/>

4. C++ libraries:

[https://en.wikipedia.org/wiki/Comparison\\_of\\_linear\\_algebra\\_libraries](https://en.wikipedia.org/wiki/Comparison_of_linear_algebra_libraries)

## Tentative schedule

Period: March 18 – 30, 2019

Contact time: 44h

Classes: 9:30 – 11:15 a.m. (3 hours in the morning)

1:00 – 2:30 p.m. (2 hours in the afternoon)

Examination: Written exam in 90 minutes with marks

Friday, April 05, 2019 (10:15 – 11:15 a.m.)

Lecture	Topic	Contents
1 (4h)	C++ Basics	<ul style="list-style-type: none"><li>- Course Introduction</li><li>- What is C++? Why Learning C++?</li><li>- Getting Started</li><li>- The First C++ Program</li><li>- C++ Basics: Data types, Operators,</li></ul>

		Control flows, Coding 1
2 (4h)	Variables, References, Pointers, and Arrays	<ul style="list-style-type: none"> <li>- Variables vs. references</li> <li>- Pointers</li> <li>- Constness</li> <li>- Arrays</li> <li>- Matrix and vector operations: Coding 2</li> </ul>
3 (4h)	Functions – Part I	<ul style="list-style-type: none"> <li>- Blocks and local variables</li> <li>- Global variables</li> <li>- Static variables</li> <li>- Extern variables</li> <li>- Function declaration and definition</li> <li>- Function default arguments</li> <li>- Call by value, reference, array</li> <li>- Return types</li> <li>- Function overloading</li> <li>- Recursive functions</li> <li>- Inline functions</li> </ul>
4 (4h)	Functions – Part II	<ul style="list-style-type: none"> <li>- The preprocessor: #include, #define, conditional compilation directives</li> <li>- Header files</li> <li>- Applications to linear algebra: Coding 3</li> </ul>
5 (4h)	Class	<ul style="list-style-type: none"> <li>- Why using classes?</li> <li>- Declaration and definition</li> <li>- Constructors and destructors</li> <li>- Operator overloading</li> <li>- Applications to linear algebra: Coding 4</li> </ul>
6 (4h)	Inheritance and Polymorphism – Part I	<ul style="list-style-type: none"> <li>- An introductory example: Matrix types</li> <li>- Introduction to inheritance</li> <li>- Access privileges for derived classes</li> <li>- Constructors and destructors</li> <li>- Calling inherited methods</li> </ul>
7 (4h)	Inheritance and Polymorphism – Part II	<ul style="list-style-type: none"> <li>- Polymorphism with virtual methods</li> <li>- Abstract classes</li> </ul>

		<ul style="list-style-type: none"> <li>- Applications to linear algebra: Coding 5</li> </ul>
8 (4h)	Namespaces and Templates	<ul style="list-style-type: none"> <li>- Namespaces</li> <li>- Templates</li> <li>- Applications to linear algebra: Coding 6</li> </ul>
9 (4h)	Debugging and Exception Handling	<ul style="list-style-type: none"> <li>- Debugging with gdb</li> <li>- Exception handling with try and catch</li> <li>- Coding 7</li> </ul>
10 – 11 (8h)	A Simple PDE Approximation	<ul style="list-style-type: none"> <li>- 1D heat equation: second-order in space and implicit first-order in time</li> <li>- Coding 8</li> <li>- Input and Output the results</li> <li>- Plotting with MATLAB</li> <li>- Error analysis</li> </ul>