



Hochiminh City University of Technology
Computer Science and Engineering
[CO1027] - Fundamentals of C++ Programming

Course Introduction

Lecturer: Huynh Tuong Nguyen
Credits: 3

Introduction

- ❖ Audience: students who have **no background** in computer programming
- ❖ Aims: provide basic knowledge and skill on programming with two important programming paradigms: **structure programming** and **object-oriented programming**.
- ❖ Demonstration language: C++
- ❖ Prerequisite: basic math knowledge
- ❖ Requirement:
 - ❖ Class attendance
 - ❖ Self-study
 - ❖ Work hard

Learning outcome

- ❖ What you will get from the course
 - Be able to describe the algorithm for your problem
 - Understand and be able to use structure programming techniques
 - Be able to implement a given algorithm using C++
 - Understand basic concepts of Object-Oriented Programming (OOP)
 - Improve your coding style
 - The process of solving problem

Course learning outcomes

No.	Learning outcomes
L.O.1	Express an algorithm for solving problem
	L.O.1.1 – Draw flowchart L.O.1.2 – Write pseudocode
L.O.2	Implement a given algorithm by a C++ program
	L.O.2.1 - Branch structure: if-else, switch-case L.O.2.2 - Loop structure: for, while, do-while L.O.2.3 – Function and recursion L.O.2.4 – Array, string, structured types L.O.2.5 – Pointer
L.O.3	Basic object-oriented concepts
	L.O.3.1 – Understanding of basic object-oriented concepts L.O.3.2 – Understanding of the issues involved in effective class design

Contents

- ❖ Basic of programming language (C++)
- ❖ Control structures
- ❖ Array and structure
- ❖ Pointer
- ❖ Recursive
- ❖ Class
 - ❖ Inheritance, template, polymorphism, and advanced topics

Syllabus

- ❖ Course meeting time:
 - ❖ Lecture: 2 hours/week for 12 weeks
 - ❖ Laboratory: 3 hours/week for 8 weeks
- ❖ Course mechanics:
 - ❖ Textbook: C++ How to program
 - ❖ Reference book: *Fundamentals of C++ Programming* – Richard L. Halterman
 - ❖ Lecture notes
 - ❖ Online materials

Syllabus

- ❖ Assessment
 - ❖ Assignment
 - ❖ Lab test
 - ❖ Final exam: 90'
 - ❖ Ratio: lab (30%), assignment (30%), final exam (40%)
(This is a tentative ratio, it may change a bit)
- ❖ Coding environment:
 - ❖ Recommend: Visual studio
 - ❖ Other IDEs are welcome

Lectures, Labs and Assignments

Session	Topics	Labs	Quizzes and Assignments
01	Libraries + Control Flow – If		Quiz 0 + Assignment 0 Out
02	Control Flow – Loop		Quiz 1
03	Array, Structure		Quiz 2 + Assignment 0 Due
04	String	Lab 1: Revision + array	Quiz 3 + Assignment 1 Out
05	Function + parameter passing	Lab 2: String + I/O	Quiz 4
06	Recursion	Lab 3: Function + parameter passing	Quiz 5 + Assignment 1 Due
07	Pointer	Lab 4: Recursion	Quiz 6
08	Pointer + linked list	Lab 5: Pointer (dynamic array, string, matrix)	Quiz 7 + Assignment 2 Out
09	OOP basic	Lab 6: Pointer + Linked list	Quiz 8
10	OOP Inheritance, vector	Lab 7: OOP basic	Quiz 9 + Assignment 2 Due
11	Advanced topic (optional)	Lab 8: OOP basic	
12	Advanced topic (optional)		

About Quizzes

- ❖ Each quiz:
 - ❖ contains (around) 10 multiple choices questions.
 - ❖ takes approximately 15 minutes at the beginning of each class.
 - ❖ is used for checking attendance.
- ❖ **IMPORTANT:**
 - ❖ Please be **on time** for the class.
 - ❖ If you miss **4 (or above)** classes, you are not qualified for the final exam.

Regulations

- ❖ Any plagiarism act will lead to zero in all tests!
- ❖ Final grade of assignment depends on the exam
 - ❖ $A_{\text{final}} = N \sum_{i=1}^N \frac{1}{T_i^{-1}}$
- ❖ Detail mapping of exam questions and assignments will be announced during the progress of the course.

“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies.”

– *C.A.R. Hoare, The 1980 ACM Turing Award Lecture*