



VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY  
UNIVERSITY OF NATURAL SCIENCES  
FACULTY OF INFORMATION TECHNOLOGY

**COURSE SYLLABUS**

<b>Course Code:</b>	TH108
<b>Title:</b>	Object-Oriented Programming
<b>Credits:</b>	4
<b>Workload:</b>	Lecture hours: 3 periods * 15 weeks = 45 periods Laboratory hours: 2 periods * 15 weeks = 30 periods Preparative hours: 2 periods * 15 weeks = 30 periods
<b>Prerequisites:</b>	TH012 - Intermediate Programming A2 TH105 - Data structures 2

**Course Objectives:**

The objective of this course is to familiarize students with object-oriented programming (OOP), a very popular paradigm of programming now in the world. Students will learn how to analyze a program by OOP methods, with the basic concept of OOP such as abstract data types, inheritance, polymorphism, etc. C++ is used as a typical example of OOP languages throughout the course.

**Main Text:** N/A

**References:**

- *Object-Oriented Software Engineering*, Ivar Jacobson - Addison-Wesley Publishing Company 1992.
- *Borland C++ Version 3.1 Programmer's Guide*, Borland International Inc.
- *C++ how to program*, H.M.Deitel and P.J.Deitel. Prentice Hall 1994.
- *Efficient C programming: A practical Approach*, Mark Allen Weiss - Prentice Hall International.
- *C++ Kỹ thuật và Ứng dụng*, Scott Robert Ladd - Nguyễn Hùng - Scitec Company - 1992.
- *Practical Object-Oriented Design With UML*, Mark Priestley, McGraw Hill, 2004.

**Course Outline:**

Chapter 1 Differences between C ++ and ANSI C

- 1.1 Advantages of OOP in some projects comparing with traditional programming
- 1.2 The ::, new, delete operators
- 1.3 Reference
- 1.4 Inline function
- 1.5 Default argument values
- 1.6 Overloading

Chapter 2 Basic Concepts

- 2.1 Abstract data structures
- 2.2 Data members and methods

- 2.3 Object, class and instance
- 2.4 Inheritance
- 2.5 Polymorphism
- 2.6 Differences between OOP and procedural programming.
- 2.7 Analyzing problems with the Object-Oriented approach.

### Chapter 3 Class and Abstract Data Types Implementations

- 3.1 Examples
- 3.2 Definition of class: class name, attributes, methods, and access specifiers (public, protected, private).
- 3.3 Constructor and Destructor
- 3.4 Implementing methods and inline methods
- 3.5 Static components
- 3.6 Friend functions or classes
- 3.7 Overloading operators
- 3.8 Encapsulation

### Chapter 4 Inheritance and Polymorphism

- 4.1 Inheritance, base classes, and derived classes
- 4.2 Type of inheritance: private, protected, public.
- 4.3 Virtual operations and virtual base class
- 4.4 Polymorphism
- 4.5 Abstract base classes
- 4.6 Examples

### Chapter 5 Templates

- 5.1 Class template
- 5.2 Method template

### Chapter 6 Converting a software project developed by the procedural programming to OOP

### Chapter 7 I/O streams

- 7.1 General model of I/O streams in C++
- 7.2 Input/output operations through I/O stream
- 7.3 Formatting I/O
- 7.4 Formatting by using Manipulator
- 7.5 Redefining I/O operator
- 7.6 Handling errors

### **Grading**

Final exam : 70%

Assignments: 30%