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

COURSE SYLLABUS

Course Code: TH110

Title: Introduction to Software Engineering

Credits: 4

Workload:

Lecture hours:	**3** periods * **15** weeks = **45** periods
Laboratory hours:	**2** periods * **15** weeks = **30** periods
Preparative hours:	**2** periods * **15** weeks = **30** periods

Prerequisites: TH105 - Data structure 2
TH107 - Introduction to Database Systems

Course Objectives:

This course provides students with basic knowledge in software engineering (SE), including SE procedures, methods, and CASE tools, which will help them know how to develop software systematically. To gain a general view, students will be introduced to many different methods. Especially OMT (Object Modeling Technique) will be used to illustrate through the course.

Main Text: N/A

References:

- *Software Engineering - A Practitioner's Approach*, Roger S. Pressman - International Edition, 1997.
- *Practical Object-Oriented Design With UML*, Mark Priestley, McGraw Hill, 2004.
- *Object-Oriented Modeling and Design*, James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen. Prentice-Hall International Editions, 1991.

Course Outline:

Chapter 1 Software Engineering Overview

1.1 Basic concepts
- 1.1.1 Software
- 1.1.2 Software Engineering

1.2 Software Development Processes
- 1.2.1 Concepts and Classification
- 1.2.2 Classical Lifecycles
- 1.2.3 Other Processes

1.3 Software Development Methodologies
- 1.3.1 Concepts and Classification
- 1.3.2 Function-Oriented Method
- 1.3.3 Data-Oriented Method

1.3.4 Object-Oriented Method
1.3.5 Other Methods
1.4 Tools and Environments for Software Development
1.4.1 Concepts
1.4.2 CASE Tools and Tool Integration
1.4.3 Software Development Environments

Chapter 2 Requirement Analysis and Specification

2.1 The Analysis Process
2.1.1 Stages
2.1.1.1 Feasibility Analysis
2.1.1.2 System Analysis
2.1.2 Requirements
2.1.2.1 Functional Requirements
2.1.2.2 Non-Functional Requirements
2.2 Feasibility Analysis
2.2.1 Business Modeling and Problem Statements
2.2.2 Analyzing Proposed Solutions and Plans
2.2.3 Planning and Estimations
2.3 Modeling
2.3.1 Data Models
2.3.2 Constraints
2.3.3 State Models
2.3.4 Function Models
2.3.5 Other Models
2.4 Analysis Documents

Chapter 3 Designing Software
3.1 The Design Process
3.1.1 Stages
3.1.2 Data Design
3.1.3 Processing Design
3.1.4 User-Interface Design
3.1.5 Requirements and Design
3.1.6 Functional Requirements
3.1.7 Nonfunctional Requirements
3.2 Data Design
3.2.1 Design Principles
3.2.2 Logical Diagrams and Functional Requirements
3.2.3 Logical Diagrams and Nonfunctional Requirements
3.3 Interface Design
3.3.1 Design Principles
3.3.2 Designing Menus
3.3.3 Function Oriented Approach
3.3.4 Object-Oriented Approach
3.3.5 Work Progress Oriented Approach
3.3.6 Diagram-Oriented Approach
3.3.7 Designing User Interface
3.3.8 User Interface Classification
3.3.9 User Interface Diagrams

- 3.3.10 User Interface Specification
- 3.4 Processing Design
  - 3.4.1 Design principles
  - 3.4.2 Designing functions
    - 3.4.2.1 Function Diagrams
    - 3.4.2.2 Function Specifications
      - 3.4.2.2.1 Using VDM
      - 3.4.2.2.2 Using Z
      - 3.4.2.2.3 Using Other Formal Languages
- 3.5 Design Documents

Chapter 4 Software Testing

- 4.1 Introduction
  - 4.1.1 Concepts
  - 4.1.2 Testing Process
  - 4.1.3 Unit test
  - 4.1.4 Function test
  - 4.1.5 Module test
  - 4.1.6 System test
  - 4.1.7 Customer confirmation
- 4.2 Methods
  - 4.2.1 Static test
  - 4.2.2 Dynamic test
- 4.3 Testing strategies
  - 4.3.1 Top-down testing
  - 4.3.2 Bottom-up testing
  - 4.3.3 Branch testing
- 4.4 The black-box testing
  - 4.4.1 Objective and definitions
  - 4.4.2 Method
- 4.5 The white-box testing
  - 4.5.1 Objective and definitions
  - 4.5.2 Method
- 4.6 Test documents

Grading Final exam : 70%

Assignments: 30%