



Vietnam National University – HCMC
Ho Chi Minh City University of Technology
Faculty of Computer Science and Engineering
Syllabus

PRINCIPLES OF PROGRAMMING LANGUAGES

Number of Credits	4 (3.0.8)			Course Number	CO3005	
Class hours	Total: 60	Lecture: 60	Tutorial:	Lab: 0	Assignments: Y	
Project, Internship, Thesis						
Assessment	Ex: 10%	Lab:	Mid-term exam: 20%	Assignments: 30%	Final exam: 40%	
Methods of assessment	- In-class activities (10%) - Midterm: Writing, 80 minutes (20%) - Project (30%) - Final exam: Writing, 120 minutes (max) (40%)					
Prerequisite (have completed)	None					
Prerequisite (have studied)	Fundamental of programming Data structures and algorithms				CO1011 CO2003	
Corequisite	None					
Applicable Programs	Computer Science					
Course Level	3					
Notes						

1. Course Objectives:

This course is to introduce the knowledge of principles of programming language design and implementation, basic components of programming languages. A declarative programming language introduced in this course is functional programming language.

2. Course description:

This is an introductory course to the field of programming languages. This course helps students prepare for learning effective any new programming languages. This course will cover the following topics:

- Principles of programming language design
- Formal description of lexical and grammar
- Data type systems in programming languages
- Control structures in programming languages
- Data control mechanism in programming languages

3. Textbook:

- [1] Maurizio Gabbrielli and Simone Martini, *Programming Languages: Principles and Paradigms*, Springer, 2010.

Reference books:

[2] Robert W. Sebesta, *Concepts of Programming Languages*, 8th edition, Addison Wesley, 2008.

[3] Kenneth C. Louden, *Programming Languages – Principles and Practice*, First edition, Thomson Brooks/Cole, 2003.

4. Course learning outcomes:

No.	Course learning outcomes	CDIO
L.O.1	Define formally lexical and grammar requirements of a simple programming language.	1.1
	L.O.1.1 – Write lexical specification (using regular expression) of a simple programming language	1.1.2
	L.O.1.2 – Write syntax specification (using BNF or EBNF) of a simple programming language	1.1.2
L.O.2	Describe and explain the mechanisms of components of a programming language.	1.1
	L.O.2.1 – Describe and explain the mechanisms of scope and reference environment	1.1.2
	L.O.2.2 – Describe and explain the mechanisms of data types	1.1.2
	L.O.2.3 – Describe and explain the mechanisms of control structures	1.1.2
L.O.3	Implement an interpreter/compiler using a functional programming language such as Scala.	4.1; 4.2
	L.O.3.1 – Write a static checker	4.2.2
	L.O.3.2 – Write a simple code generator	4.2.2

5. Learning guidelines and assessment details:

Learning guidelines:

- Before class, students must watch corresponding lecture videos, view slides and read textbook. Students must complete hand-on exercise after every video.
- In class, students must complete some short quizzes, exercises and short assignments. Some exercises require programming so students should take your own laptop. Students will work in group to perform all programming exercises.
- After class, students must complete the exercise if it has not been completed in class. Students are recommended to review all lecture videos
- In addition, students must complete all assignments at home.

Assessment details:

- Quiz and Exercise (10%):
- Assignments (30%): Students must complete all programming assignments and the relating questions in the examination. The result of this part is calculated by harmony equation.
- Midterm (20%): written in 65 minutes
- Final (40%): written in 120 minutes

6. Course Lectures:

- Dr. Nguyễn Hứa Phùng

- A/Prof. Quản Thành Thor

Nội dung chi tiết

7. Course outlines

Week	Content	Learning outcomes	Learning and teaching activities	Assessment means
1	Chapter 1. Introduction 1.1 Introduction to programming languages 1.2 History of programming language development 1.3 Criteria to evaluate a programming language	Explain how to evaluate a programming language	- Learning before class: watch introduction video - Teaching: Discuss about PL implementation	- Midterm and final
2	Chapter 2. Lexical Analysis 2.1. Basic concepts 2.2. Regular expressions 2.3. Automata and token recognition methods 2.4. Lexer-generator tool	L.O.1.1 Write lexical specification (using regular expression) of a simple programming language	- Learning before class: watch lexical analysis video; complete hand-on quiz; read reference materials given in videos. - Teaching: group exercise to write RE; programming exercise to use tool to generate lexer - Learning after class: Complete part 1 (lexer) of assignment 1	- In-class quiz - Assignment 1 - Midterm - Final
3-4	Chapter 3. Syntax Analysis 3.1. Basic concepts 3.2. Context-free grammar and BNF 3.3. Parse tree and abstract syntax tree 3.4. Some issues when writing a grammar 3.5. Parser-generator tool	L.O.1.2 Write syntax specification (using BNF or EBNF) of a simple programming language	- Learning before class: watch syntax analysis video; complete hand-on quiz; read reference materials given in videos. - Teaching: group exercise to write BNF; programming exercise to use tool to generate parser - Learning after class: Complete part 2 (parser) of assignment 1	- In-class quiz - Assignment 1 - Midterm - Final
5	Chapter 4. Abstract Syntax Tree 4.1 Introduction to immediate language 4.2 AST 4.3 AST traversal 4.4 From Parse tree to AST	Programming a visitor to convert a parse tree to abstract syntax tree	- Learning before class: watch ast video; complete hand-on quiz; read reference materials given	- In-class quiz - Assignment 2 - Midterm

Week	Content	Learning outcomes	Learning and teaching activities	Assessment means
			in videos. - Teaching: programming exercise to write a visitor to convert a parse tree to ast - Learning after class: Complete of assignment 2	
6	Chapter 5. Functional Programming 5.1 Basic concepts 5.2 High-order functions 5.3 Immutability 5.4 Closure	Using functional programming style to solve some simple problems	- Learning before class: watch FP video; complete hand-on quiz; read reference materials given in videos. - Teaching: programming exercise to write some simple function using functional PL	- In-class quiz - Midterm
7	Chapter 6. OO Programming 6.1 Basic concepts 6.2 Scala	Define AST using some hierarchy classes and write some methods in OOP style	- Learning before class: watch OOP video; complete hand-on quiz; read reference materials given in videos. - Teaching: programming exercise to write some simple classes and their methods using OPPL	- In-class quiz - Midterm
8	Midterm			
9-10	Chapter 7. Name, scope and referencing environment 7.1 Attributes and constraints 7.2 Declaration, block and scope 7.3 Symbol table 7.4 Variables and constants 7.5 Name, dangling reference and garbage 7.6 Referencing Environment	L.O.2.1 Describe and explain the mechanisms of scope and reference environment	- Learning before class: watch name video; complete hand-on quiz; read reference materials given in videos. - Teaching: group exercise on name; programming exercise to write simple semantics analysis - Learning after class: complete part 1 of Assignment 3	- In-class quiz - Assignment 3 - Final

Week	Content	Learning outcomes	Learning and teaching activities	Assessment means
11-12	Chapter 8. Types 8.1 Type expression 8.2 Primitive types 8.3 Composition types 8.4 Type equivalence, checking and inference 8.5 Type abstraction	L.O.2.2 Describe and explain the mechanisms of data types L.O.3.1 Write a static checker	- Learning before class: watch type video; complete hand-on quiz; read reference materials given in videos. - Teaching: group exercise on type; programming exercise to write simple semantics analysis - Learning after class: complete part 2 of Assignment 3	- In-class quiz - Assignment 3 - Final
13-14	Chương 9. Control structures 9.1 Expressions 9.2 Statements 9.3 Control abstractions 9.4 Parameter passing	L.O.2.3 Describe and explain the mechanisms of control structures	- Learning before class: watch type video; complete hand-on quiz; read reference materials given in videos. - Teaching: group exercise on type; programming exercise to write simple semantics analysis	- In-class quiz - Final
15	Chapter 10. Code Generator 10.1 Virtual machine 10.2 Code generator	L.O.3.2 – Write a simple code generator	- Learning before class: watch JVM and code generator videos; complete hand-on quiz; read reference materials given in videos. - Teaching: group exercise on type; programming exercise to write simple code generator - Learning after class: complete Assignment 4	- In-class quiz - Assignment 4 - Final

Assignments

1-4	Lexer and Recognizer	L.O.1.1 & L.O.1.2 Write a lexer and parser using a compiler-generator tool such as ANTLR	-Students use ANTLR to write the lexer and recognizer for a simple PL	- Test students'program
4-7	Parser		- Students write a visitor to convert parse tree into abstract syntax tree	-Test students'program
8-11	Static checker	L.O.3.1 Write a static checker	- Students write a static checker	-Test students'program
11-15	Code generator	L.O.3.2 Write a code generator	- Students write a code generator	- Test students'program

8. Contact point

Department/Faculty charge	Computer Science department/Faculty of Computer Science & Engineering
Office	Block A3, HCMUT
Tel	(84- 28)-3864 7256, Ext: 5848
Course Cordinator	Dr. Nguyen Hua Phung
Email	nhphung@hcmut.edu.vn

Hồ Chí Minh City, 26th December, 2017

Dean

Department Chair

Course Coordinator