Course Introduction

Principles of Programming Languages

MEng. Tran Ngoc Bao Duy
Department of Computer Science
Faculty of Computer Science and Engineering
Ho Chi Minh University of Technology, VNU-HCM

Course description

MEng. Tran Ngoc Bao Duy

1. Principles of programming language design
2. Introduce some programming language paradigms: object-oriented programming, functional programming, ...
3. Formal description of lexical and grammars
4. Data type systems in programming languages
5. Control structures in programming languages
6. Data control mechanism in programming languages
7. Sequence control mechanism in programming languages

Course materials

1. **Maurizio Gabbrielli and Simone Martini, Programming Languages: Principles and Paradigms, Springer, 2010.**
2. Kenneth C.Louden, Programming Languages – Principles and Practice, First edition, Thomson Brooks/Cole, 2011.
3. Michael L. Scott - Programming Language Pragmatics-Morgan Kaufmann, 2016.
4. Alfred V. Aho, Ravi. Sethi, Jeffrey D. Ullman - Compilers, principles, techniques, and tools, 2006.

2 -> 1 -> 3 -> 4

Course objectives

MEng. Tran Ngoc Bao Duy

After complete this subject, students are able to:

- describe formally lexicon and grammar of a programming language
- describe and explain some mechanism of a programming language
- implement a interpreter/compiler for a simple programming language

Learning outcomes

- L.O.1 Define lexical and grammar requirements of a simple programming language by writing formal specification (lexical and grammatical)
  - L.O.1.1 Write lexical specification (using regular expression) of a simple programming language
  - L.O.1.2 Write syntax specification (using BNF) of a simple programming language
- L.O.2 Explain the mechanisms of components of a programming language
  - L.O.2.1 Explain the mechanisms of scope, reference environment, data types, sequence control and data control
- L.O.3 Implement a simple interpreter/compiler
  - L.O.3.1 Write some components of an interpreter/compiler such as intermediate code generation, static checker, or target code generation

MEng. Tran Ngoc Bao Duy

Study guidelines

- Materials (lecture slides), video lectures and homework exercises are posted on the subject's website. Students watch video lectures (can be viewed multiple times), read books, lecture slides and practice exercises before going to class.
- At the beginning of each class, students will do an online quiz before doing classwork.
- After each lesson, students review the lecture video and continue with the exercise if not completed.
- Students can attend at least one class (with the same lecturer) for a unit so do not need to require a make-up class from your lecturer.
- Students have to complete the video lecturers, do an online quiz and programming code for attendance. *If you do not complete over 20%, you are not allowed to take the final test*.

MEng. Tran Ngoc Bao Duy

Content and schedule

No	Topics	Assignment
1	Course intro, Introduction to programming languages and compilers	
2	Lexical analysis	A1 - In
3	Syntax analysis	A2 - In
4	Programming languages paradigms (OOP and FP)	A1 - Out
5	Abstract syntax tree generation	A3 - In
6	Midterm Review	A2 - Out
7	Naming, scope and referencing environment	
8	Type (Part 1)	A4 - In
9	Type (Part 2)	
10	JVM and Code generation	A3 - Out
11	Sequence control	
12	Control abstraction	A4 - Out

MEng. Tran Ngoc Bao Duy

Assessment

MEng. Tran Ngoc Bao Duy

- Quiz: 10%
- Assignment: 30% (A1, A2, A3: 30%, A4:10%)

$$X_i = \frac{2 \times A_i \times B_i}{A_i + B_i}$$

A_i: scored by testcases grading
B_i: scored by some questions in midterm or final test.

- Midterm: 20% (80 mins)
- Final Exam: 40% (120 mins)

Benefits of Studying

- Increased capacity to express idea.
- Improved background for choosing appropriate languages.
- Increased ability to learn new languages.
- Better understanding of the significance of implementation.
- Better use of languages that are already known
- Overall advancement of computing.

MEng. Tran Ngoc Bao Duy

MEng. Tran Ngoc
Bao Duy

THANK YOU.