

L0930300	Principles of Programming Languages	Isao Sasano
----------	--	--------------------

College	College of Engineering
Department	Department of Information Science and Engineering
Grade	2 nd Year Students
Semester	Second Semester
Credit	2
Course Type	Compulsory Elective
Course Classification	Specialty
Mode of Delivery	Lecture

Course Outline

Computers operate according to programs, which are described in programming languages. Programming languages should be defined so that the semantics of programs are clear to anybody. Programming languages are defined by giving their syntax and semantics. We illustrate how to define the semantics of programming languages by giving operational semantics and axiomatic semantics to tiny languages. We also illustrate the concepts of type systems by giving a type system for a tiny subset of the C language. As the classification of programming languages by their computational models we give explanations to imperative, functional, logic, and object-oriented languages. We also present major concepts of programming languages such as the scope rules of variables and the mechanisms of parameter passing.

Achievement Objectives

1. Understanding the formal description of the semantics of programming languages and being able to describe semantics of short programs in tiny languages based on axiomatic and operational semantics
2. Understanding the scope rules (static and dynamic) of variables and being able to explain how the meaning of programs changes according to the scope rules
3. Understanding the mechanisms of parameter passing and being able to explain how the meaning of programs changes according to the mechanisms
4. Being able to explain the classification of programming languages based on their computational models
5. Being able to check whether or not simple programs in tiny C language have the type consistency according to the given simple type system

Course Plan

【Course Plan】	【Assignment (including preparation and review)】
1. Classification of programming languages and a simple example of	Section 8.1 of the reference book

programming languages

- Imperative, functional, logic, and object-oriented languages
 - A simple language --- Little quilt
2. Imperative languages (1) Section 3.1, 3.2, and 3.3 of the reference book
 - Assignment
 - Structured programming
 - Control flow
 3. Imperative languages (2) Section 3.4 of the reference book
 - Break statement, continue statement, goto statement, and return statement
 4. Imperative languages (3) p. 109 and p. 87 of the reference book
 - Compilation of sentences
 - Short-circuit evaluation
 - Assertion
 5. Imperative languages (4) Section 3.6 of the reference book
 - Hoare triple
 - Hoare logic
 6. Imperative language (5) and a small examination Section 5.2 and 5.3 of the reference book
 - Procedure
 - Parameter-passing mechanisms
 - Scope rules of variables
 7. Operational semantics (1) Section 13.3 of the reference book provides an operational semantics, although it is for a functional language. This lecture provides one for an imperative language, but the basic ideas are similar. Since imperative languages support assignments, we have to care about the states, which the reference book does not provide the explanation for. Each student is recommended to try to consider how to define an operational semantics for an imperative language.
 - An operational semantics for a tiny C language
 - Definition of the meaning of expressions
 - Definition of the meaning of sentences
 8. Mid-term examination and explanation of the answers Review the contents of all the lectures until the last one
 - Paper-and-pencil test for checking the understanding of the contents of the lectures from the first to the seventh
 9. Type system Review variable declarations of

<ul style="list-style-type: none"> ● Variable declarations in the C language ● A type system for a tiny C language 	the C language
10. Lambda calculus	Chapter 14 of the reference book
<ul style="list-style-type: none"> ● Syntax of the lambda expressions ● Beta transformation 	
11. Object-oriented languages (1)	Chapter 6 of the reference book
<ul style="list-style-type: none"> ● Class 	
12. Object-oriented languages (2)	Chapter 7 of the reference book
<ul style="list-style-type: none"> ● Inheritance ● Virtual function 	
13. Logic programming	Chapter 11 of the reference book
14. Functional programming and a small examination	Chapter 8 and 9 of the reference book
15. Final examination and explanation of the answers	Review the contents of all lectures
<ul style="list-style-type: none"> ● Paper-and-pencil test for checking the understanding of the contents of the lectures from the first to the fourteenth 	

Evaluation Method and Criteria

Mid-term exam is evaluated on a 40-point scale, final exam a 50-point, and small exams a 10-point. When the mid-term exam is M point, the final exam F point, and the small exams S point, the overall score is $S+M+F*(100-(S+M))/50$.

Textbooks and Reference Materials

We make materials public on a web page.

A reference book is:

- Programming languages concepts & constructs 2nd edition, Ravi Sethi, Addison-Wesley, 1996.

This class is largely based on this book. Currently it takes time to purchase this book.

A recommended book is:

- Concepts in programming languages, John C. Mitchell, Cambridge University Press, 2001.

Pre-Course Preparation

Introduction to computer science and introduction to programming 1 and 2.

Office Hours, Contact Method

Before and after each lecture or any time agreed on by email
--

Relevance to Environmental Education

None
