

Programming Language Theory

Syllabus Number

3B221

Special Subjects

Elective 2 credit

MORI, Takuo

1. Course Description

In this course, students learn each programming paradigm and its computational model, syntax of programming languages and semantics of programming languages.

Especially, students learn the operational semantics of the C language which is one of the mostly used and the most practical languages, and the Java language which is an object-oriented language, and of which class files run on virtual machines, thus the Java language is environmental neutral.

Students also learn the summaries of functional programming languages of which computational model is the lambda calculus, logic programming languages which are focused on Artificial Intelligence (AI), and interpreter languages also called light weight languages) which are used in for example, string processes.

Students acquire skills related to the diplomatic policy, DP4C and DP4M

2. Course Objectives

The goal of this course is that students master the following abilities;

Students understand the features of representative computational models, programming paradigms and programming languages and can explain those described above.

Students can classify programming languages from the viewpoint of translating source code to machine languages, choose an appropriate programming language according to the purpose of the use.

Students can explain the relation between notations used in defining programming languages and the context-free grammar, can show the derivation process by a context-free grammar by using a syntax tree and can explain how to remove the ambiguity from a context-free grammar.

Students can explain the behavior of the Random Access Machine(RAM) whose computational model is a Von Neumann computer.

Students can explain how to invoke recursive functions/methods on a stack machine.

Students can explain the feature of procedural languages, such as C, and can explain how to work those executable codes, can explain those operational semantics of those.

Students can explain the feature of object-oriented languages, such as JAVA, and can explain how to work those executable codes, can explain those operational semantics of those.

3. Grading Policy

Grading policy:

Midterm report(50%), Examination(50%).

The way of feedback;

Answers for questions or feedback for the contents of class and examination will be given in a class, through LMS or ioffice hours.

4. Textbook and Reference

Textbook

大山口通夫,五味 弘共著 プログラミング言語論 コロナ社、ISBN-13: 978-4339027044

5. Requirements(Assignments)

Before each class, materials related to the class will be published through LMS. Students should download them to their own devices or print them to make it possible to refer to or to take notes.

Students should read these materials and grasp what they do not understand and they understand in an hour.

After each class, student should review the class through tests on the LMS in half an hour.

6. Note

Students can hardly earn credits not submitting the mid-term report. Thus, it is expected students to observe the deadline.

As for the self-learning support students are expected to utilize materials, such as slides, handouts and quizzes on the LMS

Before taking this course, students should take the following courses;
Programming 1, Programming 2,

At the same semester with this course, students should take the following courses;
Data structure and Algorithms, Computer Science Programming 1

After taking this course, students should take computer architecture, information security, Computer Science Programming 2, Introduction to the theory of automata and computation, and Computer Architecture

If a student has a question on quizzes or mid-term report or examinations, ask the question in the class or during the day or through LMS.

This course is a required course, and relates to the mid term 4-4 of the attaining targets for learning and educating, in the JABEE program.

7. Schedule

- [1] Features and classification of programming languages 1, low level languages and high level languages, compiler language, interpreter language, intermediate language
- [2] Features and classification of programming languages 2, computational models, Turing machine, Random Access Machine and imperative languages
- [3] Features and classification of programming languages 3, lambda expressions, lambda calculus and functional programming languages
- [4] Features and classification of programming languages 4, prolog as a logic programming language, object-oriented languages and its computational model.
- [5] Notation techniques for programming languages, Backus Naur Form, Context-free grammar, the front-end and the back-end of compilers
- [6] Imperative programming languages 1, structured programming, data type
- [7] Imperative programming languages 2, procedures
- [8] Semantics of programming languages, operational semantics, the operational semantics of Random Access Machine(RAM)
- [9] Semantics of programming languages, the operational semantics of the C
- [10] The operational semantics of the small C, definition, invoking of functions, returning values from a function, pointer, array
- [11] Object-Oriented Programming languages 1
- [12] Object-Oriented Programming languages 2, interface, inheritance/composition/delegation
- [13] The operational semantics of a subset of Java 1, non object-oriented functionality
- [14] The operational semantics of a subset of Java 2, object-oriented functionality
- [15] Summary and examination