

# Computer Science Programming 1

Syllabus Number

3C211

Basic Major Subjects

Elective Requisites 2

credit

MORI, Takuo

## 1. Course Description

Through the Java language, students learn basics of Object-Oriented Programming (OOP).

Java is one of the most used languages in the current software development, and the OOP is one of the most general software development methods.

Being able to utilize Java and the OOP makes it possible for students to develop practical programs.

Java has both features that compiler languages have and that intermediate languages of which executable codes run on virtual machines have. Developing and executing programs through a text-editor and a shell, students will deeply learn Java, such as the relations between source files and class files or package managements and directory structures.

Thus, students are basically required to watch video materials and to fill in a preparation worksheet to check for understanding before each class.

In addition, it is planned that except for general exercises, students learn and discuss the contents of tasks given in a video or a class, in a group under the supports of upper-class students called Student Assistants (SA).

Students acquire skills related to the diplomatic policy, DP4C.

## 2. Course Objectives

The goal of this course is that students can express typical concepts in the OOP, such as, encapsulation, inheritance and polymorphism through the grammar of Java.

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

Students can write a program in Java, giving a small or easy problem and its specification or algorithm.

Students can express basic OOP through Java.

Students can develop programs in Java by using a text-editor and a shell.

Students can increase the depth of one's own understanding on Java by actively joining group-works.

## 3. Grading Policy

Grading Policy;

Worksheets(20%), programming exercises (including both required and optional ones)(30%), examination(50%).

The way of feedback;

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

## 4. Textbook and Reference

Textbook

結城浩著 Java言語プログラミングレッスン 第3版(下) オブジェクト指向を始めよう SBクリ エイティブ,ISBN-13: 978-4797371260

Reference

結城浩著 Java言語プログラミングレッスン 第3版(上) オブジェクト指向を始めよう SBクリ エイティブ,ISBN-13: 978-4797371253

## 5. Requirements(Assignments)

Each class is provided in the form of a flipped classroom. Before each class, each student should spend about 1.5 hours watching the lecture video and working on the preliminary worksheet to understand what you have understood and what you have not understood.

After class, each student should spend about 30 minutes on the post worksheet again, execute each program, and check the results. Also, if there are optional exercises, please take the initiative to complete them.

If you have any questions about worksheets, exams, etc., please ask them in each class or use the office hours of the instructor to resolve them.

## 6. Note

In this course, it is planned that each student uses BYOD. Therefore each student needs to install a JAVA development environment on one's own BYOD.

The installation procedure will be shown in the lecture video. For students in previous years, we will respond separately.

The LMS will be used to support independent study.

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

Mathematical logic, Programming 1, Programming 2, Exercises in Programming 1.

At the same semester with this course, students should take the following courses;

Exercises in Programming 2, Data Structure and Algorithms, Programming Language Theory.

After taking this course, students should take all the courses related to programming.

This course is a required course, and relates to the mid term 4-2 of the attaining targets for learning and educating, in the JABEE program.

#### 7. Schedule

- [1] Java language
- [2] Operators, variables and types, if-statements, loop processing
- [3] Arrays and enhanced for-loop
- [4] Re-introduction to the object-oriented programming
- [5] Object variables, class variables and command-line arguments
- [6] Super class, sub class and inheritance
- [7] Super class, sub class, polymorphism, and abstract class
- [8] General exercise 1
- [9] Access modifiers, how to use packages
- [10] Interfaces
- [11] How to implement plural interfaces, Recursive processes in object-oriented languages
- [12] Exceptions, what is an exception ?, typical exceptions
- [13] Exceptions, various exceptions, remarks on programming using exceptions, new exception handling introduced in Java SE7
- [14] General exercise 2
- [15] General exercise 3