

Data Structures (CS 206A)

Lecturer: Otfried Cheong

Lecture time: Mon, Wed 10:30–11:45

Course webpage: <http://otfried.org/courses/cs206>

From a previous course evaluation:

도대체 왜 Piazza를 사용하는겁니까? KLMS를 사용하면 좀 더 접근성이 있고,

- Nice Wiki format, where users can work together to answer a question. Student answer / instructor answer.
- Notifications and smartphone app let me answer questions very quickly.
- Students can ask questions anonymously.
- I'm teaching three courses this semester, and I work with Piazza much more efficiently.

No textbook

But we have lecture notes, copies of the slides, example code, and Wikipedia articles. I expect you to take some notes during the lecture.

Piazza

You must regularly check the announcements on Piazza (see webpage). If you register there, they will be emailed to you automatically.

We will use Piazza for answering all your questions about the course contents. You can ask questions anonymously. You can ask questions in English or Korean.

Homework

Graded programming projects (1 – 2 weeks time),
Ungraded practice exercises (some reviewed in class)

Homework requirement

You must submit **all** programming projects.

Participation

We will take attendance in every class. You have four missed classes free—use this for doctor appointments, interviews, etc. You do not need to send me email about missing a class.

Grading Policy

Programming projects (20%), Midterm exam (30%), Final exam (40%), Participation (10%).

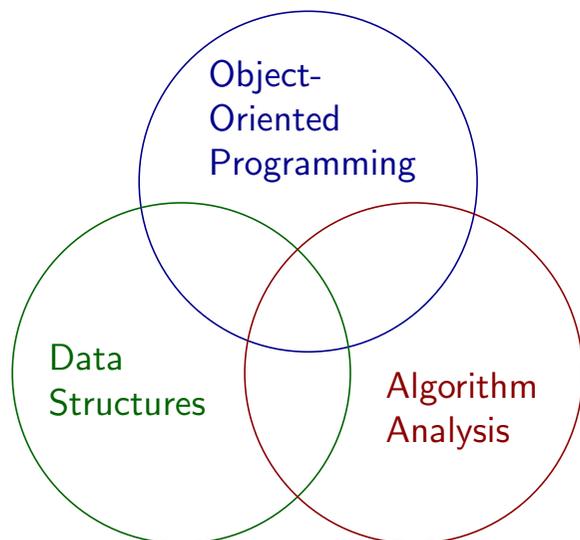
Improving your programming skills!

Know about standard data structures and can use them in your own programs.

Understand how to implement simple efficient data structures.

Programming is fun!

Can you recognize beautiful program code, and ugly one? An elegant solution and a bungled one?



- Abstract data types (ADT)
- Linked data structures
- Recursion
- Basics of algorithm analysis
- Standard ADT: Lists, stacks, queues, maps
- Applications of stacks, queues, and maps
- Implementation of data structures using lists, trees and hash tables

Many of you know some Python, Java, or C.

In my experience, somebody's first programming language shapes their programming style.

It is difficult to write beautiful, elegant code in C.

Java is a boring, verbose, ugly, dying language.

Java would already be dead without Android. Android uses Java because there are millions of Java programmers.

C++ is a powerful language, but has a steep learning curve, and it is very easy to make mistakes that are hard to find.

We will use **Scala**, a modern and clean language that encourages good style and concise, elegant code.

You will have to use Scala for all homework assignments, and all example code will be in Scala.

Of course, many (most?) of you will still have to learn C, C++, and perhaps Java. Some of you may even have to use C in this semester.

You may not have a chance to use Scala again soon (but more and more companies use it for server-side programming).

Knowing Scala, you can quickly learn modern languages like Apple's Swift or Kotlin, which borrow many concepts from Scala.

And you are not one of millions of Java programmers. . .

The only problem with Scala is that it is not commonly taught to beginning programmers, so there are no great teaching materials for beginners.

I try to make all the documentation that you will need available on my Scala page. (Link on course webpage.)

I wrote a complete tutorial for programming in Scala for CS109.

Scala on-line documentation is not easy to read. Don't despair if you do not understand it—it is not because you are not smart enough!

If you don't understand—**ask!** (on Piazza)

TA Office hour—come and get help programming (when?).