

# School of AI Convergence and Data Science

## Data Structures in Python – AIX20001

---

### Contents

- [General Information](#)
  - [Course Description, and Goals](#)
  - [Texts, Materials, and Resources](#)
  - [Exams, Projects and Grading](#)
  - [Policies and Advice](#)
  - [Tentative Course Schedule](#)
- 

### General Information

#### Class Meeting Information

Section	Credit	Day and Hours	Lecture Room	Language
03	3	Tue, Fri 11:30 – 12:45	HDH 320A	Lectures in Korean and PPT/Exams&/Quizzes in English

#### Instructor

Name	Youngsup Kim	Contacts	<a href="mailto:idebtor@gmail.com">idebtor@gmail.com</a> , 010-4939-2819
Office Hours	Right after the class at 김영길 Grace School #202		
Piazza	Use for the public open questions and comments, or an option "Post to Instructors".		
TA & Tutor	최윤영 22100748@handong.ac.kr,		

---

### Course Description and Goals

#### Catalog Description – 3 Credit Hours

This course covers some of the general-purpose data structures and includes some basics of algorithms. It aims helping students understand the reasons for choosing structures or algorithms for software development. Topics covered include managing abstract data types, time complexity, linked list, stack, queue, tree, heap, sorting, hash, and graphs. Students learn a systematic approach to organizing, writing, and debugging medium-sized programs through a useful set of algorithmic data structures. They learn to develop useful data structures for organizing, representing data to solve real problems, and practicing Python coding skills.

#### Prerequisites

Students are required to be familiar with Python programming language in a basic level.

#### Objectives

1. Review the Python programming basic skills such as list, tuple, dictionary, JSON and a bit of Object-Oriented programming as well.
2. Understand the concepts of algorithm, abstraction, recursion, and time complexity.
3. Program data structures such as stack, queue, linked list, tree, heap, sorting and graph.
4. Get familiar with IDE(Interactive Development Environment) such as Jupyter-Lab, Git and GitHub.

#### Program Outcomes

PO1 - Scientific Base: an ability to apply the knowledge and information of math, science and engineering.

PO2 - an ability to design and conduct experiments, as well as to analyze and interpret data.

#### My Own Objectives

1. Study hard to share.
  2. Give a fish, and you feed him for a day; teach a man to fish and you feed him for a lifetime.
- 

### Texts, Materials, and Resources

Required Textbook: None

## Video Lectures

There are many lectures on data structures subjects using available on YouTube.

## Joining Piazza (LMS) is required.

To join Piazza, go the [www.piazza.com](http://www.piazza.com) and follow the instruction to register. If you cannot get in, email me, and then I will invite you.

- School: Handong Global University
- Course: AIX20001 Data Structures in Python

Most of our communication between us will go through this site. You should post your questions here. Then your peers, TA or I will answer them. The average response time is about 15 minutes if we all work together and help each other.

## IDE(Integrated Development Environment)

They are the worst tools if you want to be a professional programmer because they hide what is going on from you, and your job is to know what is going on. An IDE, or "Integrated Development Environment" may turn you stupid. They are useful if you are trying to get something done quickly, but not for learning to code at the beginning, they are pointless.

- Install Anaconda Package that delivers Jupyter Lab (and Jupyter notebook)

## GitHub – the place we will go every day during this semester.

<https://github.com/idebtor/DSPy>

- Select "Watch" and "Star" buttons at the top of the GitHub page.
- Select and read "README" first.
- **Select and read "GettingStarted"** and follow the instructions to get started this course.

You may see the following topics and more:

- Install "Git" and "GitHub Desktop"

---

## Exams, PBLs and Grading

### Quizzes and Exams

**One** midterm and one final exam, and pop quizzes without a prior notice. You may expect to have about a quiz, a project, or a kind of test whenever every major topic is completed.

### Class Participation and Q/A's on Piazza

You are encouraged to post your questions such as homework questions, debugging, errors, anything that other students may also be concerned as well. You may post some recommended resources you find and share with your colleagues such as websites, tips, video lectures. Also, you are encouraged to help your peers by answering questions **on Piazza**.

### PBLs or Problem based Learning - programming assignments.

This course expects many hours of programming, and you'll work on your own. Programming assignments will be given almost every week. Upload your file(s) at least **by one hour before the midnight on its due**. Don't ask me one-minute or one-hour excuse. **Follow TA's instructions if any.**

## Grading

Grades will be assigned based on the following weights:

PBLs (Problem based learning)	45
About 15 pop-quizzes, and Exercises	1~5
Midterm, Final	20, 20
-0.5 per tardiness, -1.0 per absence	-5
Total	100

Letter grades will be assigned using the following scale:

Grade		+
-------	--	---

A	90.0	95.0
B	80.0	85.0
C	70.0	75.0
D	60.0	65.0
F	Below 60.0	

- **Study hard to give:**  
We may have labs and pop-quizzes during the classes, especially, in the beginning of the semester. When we meet in class in person, two students' team up loosely, study together, and take quizzes and do the labs and help each other.
- If you do not agree with my grading policy, you should let me know at the first week. **At the discretion of the instructor, grades may be "curved."**

---

## Policies and Advice

### Classroom Seat - n/a (not applicable this semester)

Within a week or two after the term begins, your seat will be fixed for the semester. We may try another seat shuffling, if majority of students wish, for the second half of the semester.

### Late Work

There will be 25% late penalty for the first 24 hours. No credit after 24 hours of the due date.

### Absences

Attendance matters. Two tardy and two absent are excused without penalty. I would not consider your oversleeping, hangover, birthday, cold, or body ache considered as an excuse.

### Collaboration and Cheating

All incidents of cheating will be reported to the Office of Student Affairs, who will maintain records of your academic misconduct.

1. Never have a copy of someone else's program in your possession either electronically or on paper and never give your program to someone else.
2. Discussing an assignment without sharing any code is generally acceptable. Helping someone to interpret a compiler error message is an example of permissible collaboration. However, if you get a significant idea from someone or internet sources, acknowledge them in your assignment.
3. No cheatings whatsoever in exams and quizzes.
4. In group projects (if any), you share code freely within your team, but not between teams. Each individual in a team is responsible for the entire project.
5. Cheating on homework or project will lower your letter grade by one at the first time. Cheating on an exam, project or cheating twice in any way, will earn you an F in the course. I reserve the right to assign an F in the course to anyone who cheats even once, though I might not exercise it.
6. Never post a complete program on Piazza for help or question, but a line of code that causes an error. In that case, you do not forget to post the entire error message along with a line of code.
7. You must include the following line at the top of your source file with your name signed. On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this programming assignment. Signed: \_\_\_\_\_

### Advice

In learning programming, a must is **to practice (which means both coding and debugging)**. As you read the lecture notes, try out the examples. Moreover, if you are unsure how some new construct works, write a small sample program and see! **If you approach the course by saying, "I will have fun learning to think in new ways" then you will do well. If you instead say, "I will go through this course and manage to get a pass grade." then you will get frustrated.**

### Reservation of Rights

I reserve the right to change this syllabus, including without limitation, these policies, without prior notice.

---

# Weekly Course Schedule

We are going to build this table as we progress this course.

Wk	Topics and Contents	Quiz, Homework
1	Chapter 1: Introduction Piazza, GitHub, Jupyter-Lab	PBL1 PythonReview.ipynb Ex1: ListComp.ipynb
2	Chapter 1: OOP	PBL2 OOP.ipynb
3	Chapter 2: Exception and JSON	PBL3 OOP Fraction.ipynb PBL4 JSON Coding.ipynb
4	Chapter 3: Performance Analysis	PBL5 – Profiling.ipynb
5	Chapter 3: Stacks and Queues	PBL6 Postfix.ipynb
6	Chapter 3: Stacks and Queues	PBL7 Deque Profiling.ipynb, PBL8 Circular Queue.ipynb
7	Chapter 3: Linked List	PBL9 ListSortedUnsorted.ipynb PBL10 DoublyLinkedList.ipynb
8	Midterm exam	
9	Chapter 4: Recursion	PBL11~14 Recursion
10	Chapter 5: Binary Search	PBL15 – Searching.ipynb
11	Chapter 5: Sorting	PBL16 – SortingAdvanced.ipynb PBL17 – EmpiricalAanalysis.ipymb
12	Chapter 6: Hashing	PBL18 – Hashmap
13	Chapter 7: Binary Tree	PBL19~20 – Treee
14	Chapter 7: AVL Tree	PBL22 – AVLTree.ipynb
15	Chapter 8: Heap	PBL23 – Hashing
16	Final Exam	
	(Optional) Chapter 10: Graphs	PBL - graph

Things to do during the first week:

1. **Read and follow instructions in** <https://github.com/idebtor/DSpy/README> and GettingStarted
2. **Join Piazza.** ([www.piazza.com](http://www.piazza.com))  
1 주차 월요일까지 Piazza 등록하고, 목요일까지 a simple nickname (점수 공개용 별명)을 Piazza 에 **instructors** 만 읽도록 올리십시오. (미이행시, 각각 -0.5 씩 감점합니다.)  
Use a Piazza folder for your homework submissions, Q&A and any messaging with me.
3. Install “Git” and “GitHub Desktop” and clone github/idebtor/DSpy repository.
4. Install **Anaconda Package to use Jupyter Lab.**
5. Finish Ex1: ListComp.ipynb and PBL1 PythonReview.ipynb.

Written by Youngsup Kim ([idebtor@gmail.com](mailto:idebtor@gmail.com))