[Online Only] Introduction to Python Programming

CREDIT	3	INSTRUCTOR	Keeheon Lee
OFFICE	IVHC407	OFFICE HOURS	TBA
TIME	TBA	CLASSROOM LOCATION	TBA
E-MAIL	keeheon@yonsei.ac.kr		

[COURSE INFORMATION]

COURSE DESCRIPTION & GOALS	The goal of this course is to develop code literacy. This course is intended for students in a wide variety of areas of study. Students will learn to write a computer software using Python (the most popular computer programming language). Topics include but not limited to computer organization, compiler, data types, variables, statements, decisions, loops, expressions, operators, reusing code with functions, using library code, selecting the optimal data structure to manage complex data, web programs, and sharing data with other programs. The following book is the required text for this course. • Barry, P., & Griffiths, D. (2009). Head First Programming: A Learner's Guide to Programming Using the Python Language. " O'Reilly Media, Inc.".
PREREQUISITE	N/A
COURSE REQUIREMENTS	• Fully Online. • The class will be flipped. In a flipped classroom, students watch online lectures, collaborate in online discussions, or carry out research at home while engaging in concepts in the classroom with the guidance of a mentor (Flipped Classroom, Wikipedia). • You must watch the recorded lectures online and then attend the online (live) classroom meetings. There are homework assignments at the end of the recorded lectures and their deadlines are before the classroom meetings. o The recorded lectures will be ready in LearnUs (open.yonsei.ac.kr) from 00:00AM of the previous day. o The time for the online (live) classroom will be discussed on the first day. The first online (live) classroom will be held at 3 PM KST. You will find the zoom link in LearnUs. o Daily online lab session will be held by TA. The time for this session will be discussed on the first day. • During the online (live) classroom meetings, we will focus on questions that you have until then. Please, post your questions in Q&A board before you come to class.

GRADING POLICY	Total score: 100 Attendance (15) - One additional assignment for each absence - Absent for more than 4 times will result in F - E-attendance system only 15 Assignments (75) 1 Midterm exam (5) 1 Final exam (5) * I will share a Google Colab notebook with each student. A student will write Python codes on the notebook for solving problems in homework assignments, midterm exam, and final exam. Grading table (for Score X) 100 > X >= 95: A+ 95 > X >= 90: A0 90 > X >= 80: A- 80 > X >= 75: B+ 75 > X >= 70: B0 70 > X >= 60: B- 60 > X >= 55: C+ 55 > X >= 50: C0 50 > X >= 40: C- 40 > X : F * Collaboration and plagiarism on exams is strictly prohibited. Anyone caught plagiarizing or collaborating on exams will be dealt with in accordance with the Student Code of Conduct. The penalty for plagiarism or collaborating on the first exam is a 0 (zero) grade for the exam. The penalty for plagiarism or collaboration on the final exam is a "F" grade for the course. ** Communication channels: Mainly, all will be posted in LearnUs (open.yonsei.ac.kr). You will submit assignments through LearnUs. All the questions should be posted in LearnUs and shared with classmates. Late submission is not allowed.
TEXTS & NOTES	Two textbooks are used: (1) Head First Programming
INSTRUCTOR'S PROFILE	Keeheon Lee is an assistant professor of Creative Technology Management, Underwood International College.

[WEEKLY SCHEDULE]

WEEK (PERIOD)	WEEKLY TOPIC & CONTENTS	COURSE MATERIAL & ASSIGNMENTS	NOTES
1	Day 1. – Course introduction with syllabus – Computer and Programming Language o What is a computer? How does a computer work? o What is computer programming? o What is computer programming language?	Assignment 1. Computer organization – Explain what a computer is and how a computer works.	
	Day 2. Guess the Number! – Variables – Basic data types – Control statements (for, while, if, if-else, if-elif-else) – Write a Python program, 'Guess the number' game.	Assignment 2. Rock, paper, scissors – Write a simple Python game, 'rock, paper, scissors'.	
	Day 3. Robotic Process Automation (Coffee Trader) – learn to retrieve data in a form of string from the Internet (web) using the urllib package and the packages functioning like urllib such as requests and beautifulsoup. Additionally, students learn how to deal with string-type data. – Hyper Text Transfer Protocol – Requests – Strings	Assignment 3. Buy or sell coffee beans – Write a simple program that reads data on web, extracts some parts selectively, and recommends if one should buy coffee beans or not.	
	Day 4. Robotic Process Automation (Stock Trader) – learn to retrieve data in a form of string from the Internet (web) using the urllib package and the packages functioning like urllib such as requests and beautifulsoup. Additionally, students learn how to deal with string-type data. – BeautifulSoup – Strings	Assignment 4. Buy or sell stocks – Write a simple program that reads data on web, parses data using BeautifulSoup, and recommends if one should buy stocks or not.	

WEEK (PERIOD)	WEEKLY TOPIC & CONTENTS	COURSE MATERIAL & ASSIGNMENTS	NOTES
	Day 5. Organizing Code as a Function – learn the concept of function and write simple functions. – 4 types of functions based on input and output. – Built-in functions – User-defined functions	Assignment 5. Iteration and recursion – Write simple programs that use iterations and recursions such as computing factorial, generating Fibonacci numbers, checking if a word is a palindrome, with functions	
2	Day 6. Data Handling (File, Database) – learn to deal with data in files and databases. Here, students learn list and functions related to list, function open, package sqlite, tuple, hash, and dictionary. Day 7. Point-of-Sales System and Modules – learn string formatting, modules, and code reusability. Day 8. Make User Interface Graphical – learn to use pygame and tkinter packages and widgets to make GUIs Day 9. Make User Interface Interactive – learn to make a graphical user interface that gets data and handles the data. Additionally, students learn the concept of MVC (model, view, controller). Day 10. Make User Interface More Interactive – learn to make a GUI more interactive by	Assignment 6. Relational DB – Write a simple program to perform CRUD using DB Assignment 7. POS system with DB – Write a simple program for POS system Assignment 8. GUI for POS with DB – Write a simple GUI program for POS system Assignment 9. Interactive GUI for POS with DB Assignment 10. UX/UI for POS with DB	

WEEK (PERIOD)	WEEKLY TOPIC & CONTENTS	COURSE MATERIAL & ASSIGNMENTS	NOTES
	adding handling exceptions and message boxes		
3	Day 11. Make User Interface Robust – learn to make a simple media player using pygame and tkinter Day 12. Object Orientation – learn the concept of object oriented programming Day 13. Data Structure and Algorithm – learn simple data structures such as stack and queue. Additionally, students learn simple algorithms such as sorting algorithms. Day 14. Regular Expression – learn to use the 're' package to extract patterns from string data. Day 15. Web App using Flask – learn to build simple web apps using the 'flask' package.	Assignment 11. Exceptions Assignment 12. Object-oriented POS with DB Assignment 13. Efficient POS with DB Assignment 14. Membership Assignment 15. POS web App	