Sungkyunkwan University (SKKU) International Summer Semester (ISS) 2021

Algorithm and Problem Solving

Prof. Arthur Tang, Sungkyunkwan University

SHORT COURSE DESCRIPTION

This course covers the fundamental of computing and computational thinking techniques, including the basics of how to construct computer programs using sequences of logical instructions. This is a rudimentary programming course. No prior computer science knowledge is required, but we expect students with moderate computer experience and understanding of rudimentary mathematics and logic. Emphasis will be placed on implementing computer programs towards solutions for practical problems using **Python**. The objectives of this course are:

- to familiarize students with fundamental concepts of computing systems;
- to facilitate students with ability to identify, formulate and solve computational engineering problems;
- to develop students' ability to develop computer programs and to read code developed by others; and
- to provide students with hands-on experience in computer programing and computational problem solving.

READING MATERIALS

Recommended Textbook: An Introduction to Programming Using Python™, by David Schneider (ISBN 10: 1-292-10343-4; ISBN 13: 978-1-292-10343-3)

- The textbook is not mandatory: You may choose to using the recommended textbook, or any textbook that covers Python version 3.X, or other online materials (there are tons of online learning resources for Python 3.X).
- Your first reading assignment is the syllabus. It is provided as a separated file.
- Other reading materials/slides will be provided throughout the course.

COURSE REQUIREMENTS AND GRADING

Teaching Methodology

This course incorporated the Outcome Based Learning approach. It is a student-centered approach for teaching and learning. The course has a list of **Intended Learning Outcomes**. All Teaching and Learning Activities throughout the course are aimed to help students achieving these Intended Learning Outcomes. Throughout the semester, students will be assessed based on the **Assessments** aligned to the Intended Learning Outcomes, and the final grade will be determined based on these Assessments. In other words, any student who is able to demonstrate all the Intended Learning Outcome through all the Assessments, she/he will get a Pass grade.

Course Intended Learning Outcomes

On successful completion of this course, students will be able to:

1. Solve real-world problems that will arise in the future by using computational thinking methods;
2. Define problems computationally;
3. Develop computational algorithms to problems;
4. Develop computer programs using C;
5. Read and understand programs written by others; and
6. Apply computational thinking techniques to solve the problems in their domains.

Course Assessments

- Attendance and participation: 10%
- Homework: 36%
- In-class Programming Assignments: 24%
- Final Assignment 30%

Attendance and Participation (14%):

It is the student's responsibility to attend all class session on time and stay for the whole period. Attendance will be recorded randomly during class time. If you missed the attendance during class time (e.g. if you are late to class, or if you are not paying attention to class), you will not be given another chance to submit it. If your absence is excused, bring the documentation (such as medical documentation) within one week after the absence and you will be given credit for that attendance if approved. **Credit will only be given to documented absence**. According to university policy, a Fail Grade will be given to students who attend less than 8 0% of total number of class automatically.

Homework Assignments (35%):

A number of homework assignments will be assigned throughout the course. **No late assignment will be accepted**.

In-class Programming Assignments (21%):

A number of in-class programming assignments will be assigned throughout the course. in-class programming assignments are to be completed during class time; you should submit the in-class programming assignments before the end of the class. If your absence is excused, bring the documentation within one week after the absence and you will be given another chance to submit your in-class programming assignments if approved. Late submission will only be allowed with documented absence.

Final Exam (30%):

This course will have one final assignment. You will be given a programming assignment, and you will be given about 24 hours to complete it. Final Exam is cumulative; it covers the materials of the e ntire course. You should submit your solution before the deadline.

Grading

- Grade of this course is on a pass/fail basis.
- SKKU regulations require students to attend at least 80% of all classes. A Fail grade will be assigned to students who attend less than 80% of total number of class automatically.
- Plagiarism and cheating will not be tolerated at all:
  - Student cannot supply or use work or answers that are not one's own.
  - BOTH providing and/or accepting assistance in examinations is violation of academic honesty.
  - You may discuss homework assignments with other students, but the assignments you submit must be entirely your own work.
  - Plagiarism will not be tolerated at all. Works completed by others used in your assignments should be cited properly.
  - In all incident of academic honesty, a Fail grade will be given to student(s) automatically.

COURSE SCHEDULE

Note: This calendar is tentative and is subjected to modification throughout the semester.

– WEEK I –

Thursday (24 June)

Lecture 1: Orientation; Introduction; How computers work; The Python IDLE development environment.

Friday (25 June)

Lecture 2: Fundamentals of data representation

– WEEK II –

Monday (28 June)

Lecture 3: Variables, Type, Assignment Operator, Expression, Programming Sequences

Tuesday (29 June)

Lecture 4: Computational Problem Solving; Conditional Statement

Wednesday (30 June)

Lecture 5: Algorithm 1; Repetition Statements

Thursday (1 July)

Lecture 6: Repetition Statements; For Loop

– WEEK III –

Monday (5 July)

Lecture 7: String and Output Format

Tuesday (6 July)

Lecture 8: Data Abstraction and List

Wednesday (7 July)

Lecture 9: Processing Data; File Input/Output

Thursday (8 July)

Lecture 10: Tuple; Function

Friday (9 July)

Lecture 11: Dictionary; Random values

– WEEK IV–

Monday (12 July)

Lecture 12: Set; Recursion; Exception

Tuesday (13 July)

Lecture 13: Handling Web/HTTP data

Wednesday (14 July)

Lecture 14: Review

Thursday (15 July)

Final Exam