

Spring 2019

CSE 101 : Introduction to Computers

LECTURE 0 – COURSE INTRODUCTION

Course Information

CSE 101 : Introduction to Computers

Earlier name: Introduction to algorithmic and computational thinking

Course webpage: <https://ppawar.github.io/CSE101-S19/index.html>

Meetings: Lecture: Tue/Thu 5:00-6:20 PM

Lab: Mon: 12:30-1:50PM

place: b103

Staff

Instructor

- Pravin Pawar
- Office: B424
- Email: Pravin.pawar@sunykorea.ac.kr
- Phone: +82-032-626-1227
- Office Hours: *Tue/Thu 10:30 AM - 12:00 PM, Wed 5:00 PM - 6:30 PM*

Staff

Teaching Assistants

- Graduate (grading TA)
 - Mayukh Maitra (MS), 010-5923-3726, mayukh.maitra@stonybrook.edu
 - Salman Qavi (MS), salman.qavi@stonybrook.edu
- Undergraduate (tutoring TA)
 - Sigauke Cogitater (UG), cogitater.sigauke@stonybrook.edu
 - Soyeon Jeon (UG), 010-6631-9309, soyeon.jeon@stonybrook.edu
 - Soomin Kim (UG), 010-9413-3996, soomin.kim.1@stonybrook.edu
 - Yoora Kim (UG), 010-4187-7700, yoora.kim@stonybrook.edu
 - Changhyun Park (UG), 010-5272-2810, changhyun.park@stonybrook.edu
 - Jong Kwon Park (UG), 010-5493-0150, jongkwon.park@stonybrook.edu

Announcements

Please bring a laptop to each class

- Classes will involve lecture segments, demos
- Labs will involve student exercises

Additional video lectures are noted in the syllabus. These are strongly recommended for extra instruction to help understand Python.

Misc Information

For non-CS majors: This course is an excellent way to get an introduction to what computer science is all about and learn how to program

For CS majors: This course is the launching point into the CS major for those who have no background in CS at all

Course Overview

CSE 101

- introduces the important, central ideas of computer science
- explores computational thinking and problem solving
- covers the fundamentals of computer programming

Computer science is the *study of problem solving with computers*

- Astronomers don't study telescopes. They *use* telescopes to study the stars!
- Likewise, computer professionals *use* computers to solve important problems in the modern world
- Computer scientists also build computers and software that makes the computer hum.

An important thread of this course is **computational thinking**, which is the way computer scientists think about and solve problems

Major Course Topics

Algorithmic thinking (how to devise solutions to problems)

Flowcharting

Introduction to computer programming using the Python programming language

Basics of computer hardware

Data representation (how does the computer save data?)

Data organization (how do we manage complex data?)

Program design, implementation (coding), testing and debugging

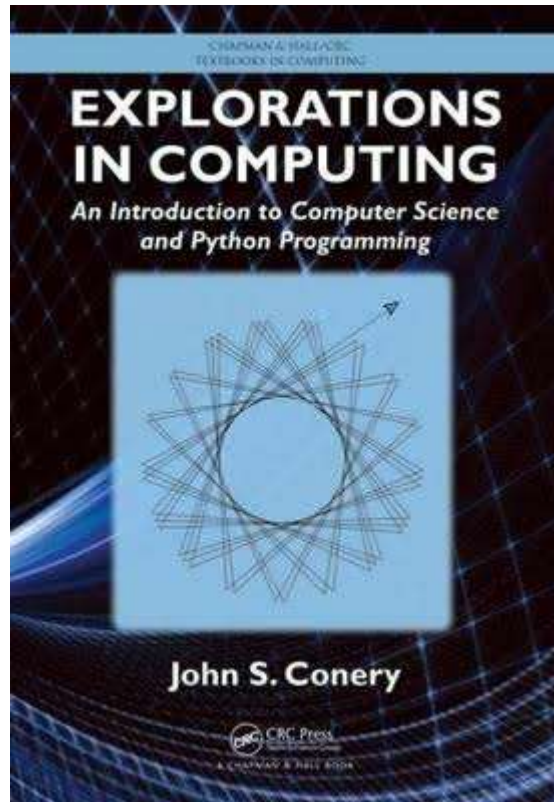
Limitations of computers

Introduction to natural language processing

Additional topics as time allows

Also, some of this list may be modified if we find other more interesting topics later

Textbooks



Blown to Bits

*Your Life, Liberty,
and Happiness After
the Digital Explosion*

Hal Abelson
Ken Ledeen
Harry Lewis

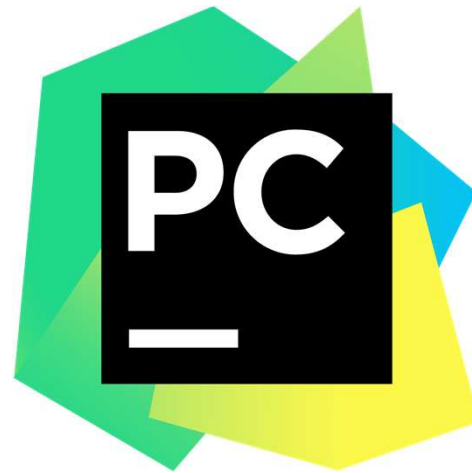
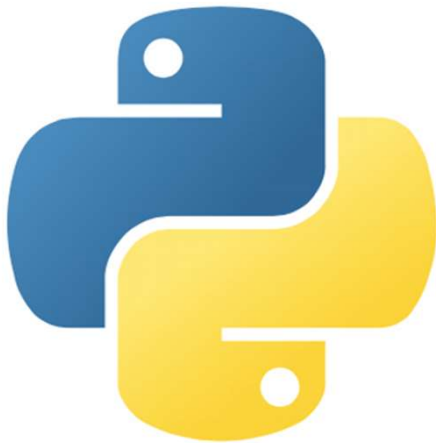
Software

Necessary software: Python (www.python.org/downloads) and PyCharm

- These are all free!

Download links will be available on the course website

We will use one of the first few classes to setup the software and get familiar with it



Homework Assignments

Over the course of the term you will be required to solve computational problems by writing software in Python

These homework assignments will reinforce concepts from class and have you explore new concepts, too

All work will due on fixed dates and times

All work will be completed on an individual basis (write your own code) *unless otherwise instructed!*

You will use **Blackboard** to submit your completed assignments

Please start early on the assignments! Most students find that completing the homework assignments for CSE 101 takes **a lot** longer than they anticipated

Late Homework Policy

Assignments must be turned in by the due date and time.

- Any part of an assignment that's late means the entire assignment is late.
- If your assignment is incomplete or not entirely working by the due date, turn in what you have to get some partial credit.

If you have an emergency situation, email me before the due date and I may be able to work something out

Bottom line: Plan ahead, start early!

Cooperation vs. Copying

Cooperation (talking over problems) is a good way to learn and is encouraged

Do not copy code. Do not let others look at or copy your code.

Copying is not allowed on homework or exams no matter the source (written or verbal)

- When you submit your homework or tests, **you are pledging that the work is your own and you have not copied it.**
- You are also pledging that you have not allowed others to copy it.

DO NOT COPY! (*Software tools catch it easily*)

Lab

Lab exercises will involve a variety of programming tasks, such as:

- running existing programs and collecting data about them
- writing your own, original, short programs to solve problems
- fixing errors in programs

Examinations

Examination dates will be posted on the schedule page of the course website.

Tentative dates are:

- Midterm exam 1: Thu, 11 April
- Midterm exam 2: Thu, 16 May
- Final exam: June 18, 3:15 – 5:45 PM

Do not miss exams

Arrange your work and travel schedules as needed to be present for examinations

Makeup exams will only be given for verified, officially sanctioned university activities.

All examinations will be closed-notes and closed-book, except one sheet of notes (A4 or 8.5x11), both sides, handwritten

Grading

Quizzes - 15% (75 points) - Short quizzes [4 given, lowest grade dropped]

Problem Sets - 20% (100 points) - Problem sets [5 assignments given, lowest grade dropped]

Labs - 10% (50 points) - Labs [~10 graded lab sessions]

Class Attendance/Participation - 5% (25 points)

Midterm Exam 1 - 15% (75 points) - First midterm exam

Midterm Exam 2 - 15% (75 points) - Second midterm exam

Final Exam - 20% (100 points) - A cumulative final exam

Policies:

- Makeup exams will only be given for verified, officially sanctioned university activities
- All makeup exams may be oral

Re-Grading

To promote consistency of grading, questions and concerns about grading should be addressed first to the TA and then, if that does not resolve the issue, to the instructor.

You are welcome to contact the TA by email or come to his/her office hour. If you would like to speak with the TA in person, and have a schedule conflict with his/her office hour, you are welcome to make an appointment to meet the TA at another time.

For the assignments, quizzes and mid-term exams, request for re-grading must be made within one week from after the announcement of grades.

TA Assistance

TAs are available almost every day each week

- Schedule is forthcoming (posted on course web)
- In “CS Commons” (next to CSD office)

Come with specific questions and/or code with which you need help

- TAs strive to spend time with everyone that comes to a session so be courteous and share the TA’s attention

Electronics in Class

Cell phones should be put away during class

Laptops may be used during periods where you are asked to work on an exercise during class

Lecture slides are available on the course website for study before class

Talk to me after class if there's an issue with this policy

Disability

If you have a physical, psychological, medical or learning disability, please contact the Student Services and Career Team.

- Location: Academic Building A208
- Phone: 626-1190

The DSS will determine with you what accommodations, if any, are necessary and appropriate

All information and documentation of disability is confidential

How to Succeed in this Class

Attend class and be on time!

- Not all information is in my lecture notes or in the book
- I sometimes do in-class demos that emphasize non-obvious details

This is an introductory course, true, but we're going to cover a lot of ground and move quickly starting from scratch

The assigned work will take a lot of your time, so practice good time management

Read the reading assignments and review the lecture notes and try out example code

- Practice is the only way to become proficient at coding
- Very often your first, second, or third attempt at solving a problem will not be successful. It is **essential** that you give yourself enough time to try different ideas, taking breaks along the way!
- Those who write extra code for problems not assigned ("for fun") generally do best in this class
- Learning to code involves learning to read other people's code

Ask questions right away if confused. Ask in class, ask a TA, come to my office hours or send email. Don't stay confused and don't get behind!

Welcome and I hope you enjoy the class!