CSE 1310

Intro

Alexandra Stefan
Introductions

• Alexandra Stefan
  – Instructor and UGrad advisor
    • You can enroll for next class before getting the grade in the current one.
    • You could have tested out of this class – let me know if you missed it.
  – Romanian
  – Mother of 2
  – Enjoying teaching and programming
    • Appreciate: feedback, questions, office hours visits

• Non- CS majors
• Undecided
• CS-majors, intended, intending
Privacy issues

• Class is being recorded
  – Sound only from you
  – Video of me and the projector, NOT the board
  – Link accessible from Blackboard
Syllabus

• Topics

• Grading
  – 2 midterms
  – Standard scheme: 90, 80, 70, 60

• Academic honesty
  – Collusion
  – Are the consequences worth it for less than 10% of your final grade?
    • Parents, employer, future schools, permanent records, risk of getting expelled.

• Students with disabilities
  – Temporary
  – Please come by my office
Communication

• UTA email
  – Sent welcome email

• Course website
  – All the course materials and information will be posted here
  – Explore it

• Blackboard
  – will only be used to:
    • Submit homework – see location, follow links
    • Post announcements – see location
    • Send email
  – Hw (not announcements, see difference)
  – See hw feedback
  – Ignore the ‘Total’ section in Grades. IRRELEVANT for your grade.
  – MyMav 4-th week
Attendance and Emergencies

• Class attendance is not mandatory, but you are responsible for the material you have missed.
  – You do not need to let me know that you will miss a class, be late for a class or have to leave early

• Exam attendance is mandatory
  – The lowest midterm grade will be dropped. If you want, you can choose to skip a midterm and use this policy for it, but I recommend against it.

• Emergencies should be documented and you should let me know of it as soon as possible (e.g. ‘coming up with a cold’)

• The following are NOT emergencies:
  – Computer crashes,
    • Create backups: Save your homeworks on a flash drive and/or email yourself
  – Network failures,
    • Submit early (even if it is not the final version)
  – Blackboard failure
    • Submit early (even if it is not the final version)
    • Check submission
    • Let me know if Blackboard seems to be misbehaving.
    • If there is a problem with your Blackboard account, email your homework to the TA (and myself).
Grade calculation

- Final score = .2 * hw_average_score
  + .3 * Final_exam
  + .25 * max(Final_exam, min(E1,E2))
  + .25 * max(E1,E2)

- Apply standard grading scheme to Final score

- Hw_average_score
  - If 200 points project, count it as 2 homeworks
  - E.g.: (100 + 100 + 100 + 200)/5
Office hours

• Come and see both me and the TA with questions.
• Location: ERB 644
• Times: see course webpage
• TA - TBD
Python

• 3.3
• Python webpage
• Download and install
• Run it
  – Create and save a .py file
  – Location of saved files
Homework submission

• See homework description on the website

• Create a folder for the current homework

• Place all your answer files in this folder

• Zip it (right click on the folder name)

• Submit it: Blackboard -> Course materials -> click on Hw title -> upload the zip file (zipped folder)

• Download it from Blackboard and verify it.
Goal of this course

- NOT to learn Python
- Develop algorithms for simple problems
- Implement the algorithms in Python (using basic instructions)
- ‘Read’ and write Python code
- Understand the principles of good programming:
  - Easy-to-read code
  - General solution (algorithm related)
  - Correct, robust
  - Design and testing
- Data types: integers, floats, strings, lists, dictionaries,
- Manipulate text files
Computer Science areas

• Graphics
• Computer vision
• Artificial intelligence
• Human-computer interaction
• Software engineering
• Algorithms and data structures
• Operating systems
  – Protect memory (OS vs applications, applications vs applications)
  – Schedule resources
• Parallel computing

• The programming language is just a tool to write the solution to a problem. The problem and the solution are [part of] the computer science.
Computer organization

• I/O
• Processor/CPU:
  – ALU (arithmetic and logic unit),
  – Control Unit,
  – registers, cache
• Main memory
  – RAM
  – volatile
• Secondary memory: disk
  – Slower (mechanical device)
  – permanent
• Bus
• Network

• Smallest unit of memory : bit (0/1)
Sorting: as a human vs as a computer

• Understand what you do when you are sorting.

• Understand what the computer can do (and what it can not do) and how the problem looks for the computer.

• Identify the general steps in solving the problem.

• Rewrite those steps in terms of what a computer can do (this is the algorithm).

• Implement the algorithm in a specific programming language to run on a specific computer. (This is the program)

• Algorithmic complexity

• What to do to succeed in this class: practice (=write code) like crazy!
Program vs Algorithm

• Algorithm
  – Description of how a problem can be solved
  – Independent of the programming language
  – Good properties:
    • Effective
    • Detailed
    • Specify behavior
    • General purpose

• Program
  – Implementation of an algorithm in a specific language
  – It runs
  – Desired properties:
    • Readable (names, comments)
    • Robust (does not break easily; recovers gracefully from ‘bad situations’)
    • Correct
    • Secure