CSc 120
Introduction to Computer Programing II

00: Basic info about this class

Welcome to CSC 120

- Second programming class
  - prerequisite: CSC 110 (or some programming experience)
  - looks at
    - more complex programs and programming problems
    - how data are represented and manipulated
    - how to think about and understand program efficicency
  - start building the toolbox of a computer scientist
- Assumes you have at least a little programming experience
  - any language is OK—we start with a Python review

Instructional staff

- Instructor: Janalee O'Bagy, Ph.D.
  - Office: Gould-Simpson room 811
  - Email: jobagy@arizona.edu
  - Office hours (for this week only):

    Thurs 1:00 -2:00pm

    Fri: 12:00-1:00pm

    or by appointment (send email; put **CS 120** in Subject line)

Check the class website for updates to office hours:
https://obagy.com/cs120/

Instructional staff

- Teaching Assistants (TAs)
  - Introductions!
  - They will answer one of these questions:
    - What about 120 was the most useful or interesting thing?
    - What advice do you have for the students?
    - What cool project have you done in CS?
- TA contact info
  - Posted on the website
- TA Office hours
  - Will post on the website
  - Held in GS 856

Meet your neighbors (2 minutes)

- Find out where they are from

- Together, decide on answers to these questions (no Googling!)
  - What year was Python created?
  - How many websites are there today and how many were there 20 years ago?
  - The Python we use is written in C. How many lines of C code do you think it takes to implement Python?

Meet your neighbor (2 minutes)

- Find out where they are from
- Together, decide on answers to these questions
  - What year was Python created?
    - first posted in February 1991
  - How many websites are there today and how many were there 20 years ago?
  - The Python we use is written in C. How many lines of C code do you think it takes to implement Python?

Meet your neighbor (2 minutes)

- Find out where they are from
- Together, decide on answers to these questions
  - What year was Python created?
    - first posted in February 1991
  - How many websites are there today and how many were there 20 years ago?
    - today: over 1 billion! (not all are active)
    - 2004: 51 million (Google was launched in 98; Gmail, Facebook in 2004)
  - The Python we use is written in C. How many lines of C code do you think it takes to implement Python?

Meet your neighbor (2 minutes)

- Find out where they are from
- Together, decide on answers to these questions
  - What year was Python created?
    - first posted in February 1991
  - How many websites are there today and how many were there 20 years ago?
    - today: over a billion! (only 200 million are active)
    - 2004: 51 million (Google was launched in 98; Gmail, Facebook in 2004)
    - The Python we use is written in C. How many lines of C code do you think it takes to implement Python?
    - Approx. 550,000 lines, including comments

Basic info about this class

- Programming language: Python
  - we will use Python 3
  - first lectures : review basics
- Development environment: Visual Studio Code
  - https://code.visualstudio.com/
- If you don't know Python:
  - need to pick up the basics quickly
    - **make use of office hours!**
    - **TA office hours are in GS 856 – see website**

Are you in the right class?

- You have the pre-requisite
  - 110

  or
  - AP classes
  - ECE 175
  - ISTA 130
  - Community college transfer credits
- But...
  - At this point in your development, learning a new language is not easy

Are you in the right class?

- We want you to succeed!
- Use the first assignment as an assessment
  - there will be time to switch to 110
  - getting a strong foundation is the key to success in the CS program
- Flip side
  - 110 students and some others, the first few lectures will be review for you

Course communication

- Class Website
  - Important links: assignments, email contacts, syllabus, etc.
  - https://www.obagy.com/cs120/
- Discord
  - See the class website for link
  - Sign up ASAP
  - Questions are posted and answered there
  - Class communication takes place there
  - You must watch the announcements channel

Course materials

- Lecture slides (required)
- Textbook (for reference only)

  Problem Solving with Algorithms and Data Structures using Python (2nd ed.), by Bradley Miller and David Ranum.
  Franklin Beedle & Associates, 2011. ISBN 978-1-59028-257-1.
- Online here:
  - https://runestone.academy/ns/books/published/pythonds/index.html
- Additional resources
  - plenty of additional on-line resources available
    - realpython.com
    - w2schools.com
  - https://docs.python.org/3

Collaborative learning

- Why are we in this classroom?
  - research shows that students learn more effectively when they are actively engaged
- Groups
  - in-class activities (ICAs) will give you the chance to work with your table group
  - no grade component associated with the group
  - graded for participation/best effort

In-class activities

- In-class activities (ICAs):
  - Activities (problems sets) that are graded for participation
  - Work in groups (except when noted)
  - Upload your work to Gradescope by 11:00pm the day of the ICA
  - Include the Word of the Day
    - you get half credit if missing this
  - Attendance is not taken, but ICAs count towards grade
    - ICAs cannot be made up
    - ICAs are 4% of your grade
  - Don’t worry about missing a few ICAs
    - There about 40 in the semester!

Out-of-class activities

- Out-of-class activities (OCAs)
  - Approx. 1 short video per week
    (some weeks there is no video)
  - Simple quizzes embedded in video
  - OCAs are 3% of your grade
  - Accessed through D2L
  - Must watch by the Saturday ending the week at 11:00pm
    (but often the material needed may be for the next lecture)
  - Watch OCA-1 before Friday's lecture!

Programming Assignments

- Given out on Mondays
- Short and long portions
  - several short problems:
    - done on CloudCoder
    - auto-graded
    - due Thursday at 7:00pm*
  - one or two longer problems:
    - due following Tuesday at 7:00pm*
    - turned into Gradescope
  - graded feedback back to you by following Tuesday
  - assignments are 30% of your grade
- ~ 12 assignments over the semester

*the first two weeks have different due dates

Programming Assignments

- Due at time specified
  - one Late Day (long problems only)
    - cannot use the Late Day for the last assignment
    - may submit up to 24-hours late
    - no permission is needed, but notify your TA that you have a submission the the Gradescope Late folder
  - the last assignment is an opportunity to re-do a prior assignment and have it regraded
  - given the above, no extensions
  - contact me (email) if you have extenuating circumstances

Programming Assignments

- Regrade requests
  - must be requested within 7 days of receiving feedback
  - make a regrade request in Gradescope and notify your TA via email

Programming Assignments

- Grading:
  - coding style
    - code structure, comments, etc.
  - functionality
    - tested using a computer program
    - you need to follow directions *exactly*
      - file names
      - function names
      - input/output format
      - … anything else specified…

You'll need to follow the spec carefully!

Absence and Class Participation Policy

- ICAs
  - Each ICA is worth about .1% of your grade
    - ✓ missing 10 would lower your grade by 1%
    - ✓ (overall grade of 100% would then be 99%)
    - ✓ don’t stress if you miss a few classes (and ICAs)
- Administrative Drops
  - Will drop all students who have not turned Programming Assignment (PA) 1 (shorts and longs) by the no-W Drop Day (1/28)
  - After 1/28, will drop any student who fails to attend class & turn in PAs and ICAs for a 2-week period

EXERCISE

Work with your group to answer questions 1-6 in the Class Policies exercise handout.

EXERCISE

Work with your group to answer questions 1-6 in the Class Policies exercise.

1. b) False
2. b) 3%
3. a) once every week
4. b) False
5. f) If I have not used my Late Day
6. c) 7 days

Exams

- Two midterms
  - approx. 5 weeks apart
    - see syllabus for dates
    - midterms are 40% of final grade (2 x 20%)

- Final exam:
  - see syllabus for dates
  - final is 20% of final grade

Midterms

- Given in the regular classroom during lecture
  - start at the beginning of lecture period
- 50 mins each
- No make-up exams
  - notify me via email if you have an emergency
  - the lowest midterm score will be replaced by your final exam score IF the final score is higher
  - (only 1 midterm score is replaced)
- Regrade policy
  - request a regrade for a question in Gradescope
  - must be done within 7 days of the midterm

Labs

- CSC 120 is a 4-unit course!
- One 60-minute lab per week
  - gives opportunity to solve more problems
  - practice is crucial when learning programming
  - gives a chance to ask questions in a smaller group setting
  - labs are 3% of final grade
- See the class website for more details
- We will have a sign-up for labs on Friday

Grading policy

Components of your final grade:

programming assignments	30%
in-class activities	4%
out-of-class activities (videos)	3%
labs	3%
midterms	40%
final exam	20%

Grading policy

Grade boundaries:

90% and above:	A
80% and above, but below 90%:	B
70% and above, but below 80%:	C
60% and above, but below 70%:	D
Below 65%:	E

(I may lower the cut-offs but will not raise them.)

Behavior and conduct

- treat each other with respect and courtesy
- don't be disruptive

- these behaviors will not be tolerated in class:
  - phone conversations, texting
  - reading newspapers or magazines
  - games, tiktok, other social media
  - extended conversations (not class related)

please leave the room if you *have* to do any of these activities; come back when done.

Academic integrity

- Work submitted programming assignments & exams must be your own work
- For programming assignments
  - OK:
    - may discuss the spec generally: “What is the spec asking us to do”?
  - NOT OK - Violation of the code of Academic Integrity:
    - discussing how to code the assignment
    - partnering with someone else on an assignment
    - soliciting help on online forums (e.g., stackoverflow, ChatGPT, etc.)
    - looking at someone else’s code
    - exchanging code

Academic integrity

- Helping someone else cheat is just as bad as cheating yourself:
  - don't show your code to anyone else
  - don't post your assignment code publicly
    - don’t post code on Discord
    - don’t post code on github.com or any other public repository (even after the end of the semester)
- See syllabus for link to the UA policy

Academic integrity

- Always OK to talk with TAs & Instructor:
  - Ask us anything!
  - We might say, “can’t tell you”
  - You are never penalized for asking a question
- Discord a good place for general discussion
  - Clarification of assignments
  - OH for individual help

Academic integrity

- Penalties
  - Minimum
    - Formal process that involves reporting the violation to the Dean of Students office
    - 0 on the assignment
  - Maximum
    - Can be much more!
    - Heavy penalties if this is a second offense
  - Better to skip the assignment than to cheat!

EXERCISE

Work with your group to answer questions 7-13 in the Class Policies exercise.

EXERCISE

Work with your group to answer questions 7-11 in the Class Policies exercise.

7. c) pgm_file.py
8. b) False
9. f) None of the above
10. a) False
11. Practice! Also gives chances to ask questions in a smaller group
12. a) True
13. a) True

How to succeed in this class

- Understand the material
  - if you don't: *ask questions!*
  - office hours
- Attend lectures & labs
  - *participate!*
- Do the programming assignments
  - start early (only 1 Late Day)
  - follow directions *exactly*
  - test your code thoroughly

Expectations for next time

- OCA-1 Video
  - D2L -> Content
    - Choose the Python Review 1 video
  - Getting started with Python (28 min.)
  - Watch before next lecture
  - Covers posted Python review slides up to slide 48