# ECS 32A: Exam #2 Details

Instructor: Aaron Kaloti

Summer Session #1 2020

## Contents

## 1 Changelog

You should always refer to the latest version of this document.

- v.1: Initial version.

## 2 Probable Exam Format

This exam will be a timed Gradescope quiz that will be proctored over Zoom. While in the Zoom meeting, you are required to have your face visible through a webcam at all times. The purpose of proctoring through this Zoom meeting is to ensure that it is truly you who is taking the exam. If I cannot confirm that it is you who is truly taking the exam, then you will get a zero. The Zoom meeting will be open at 2:10 PM. The exam (i.e. the Gradescope quiz) should become available around 2:13 PM. Once you start it, you will have 75 minutes to finish. (Thus, if you start at 2:15 PM or 2:20 PM instead of 2:13 PM, you will still have 75 minutes.) You are expected to start around 2:13 PM - 2:20 PM. If you do not start around that time range, then I will probably give you a zero unless you provide some good enough reason over email ASAP.

All audio will be muted during the meeting. This means that you can use headphones or ear buds and listen to music, if you want. I am allowing this because I know that for some students, they may not have access to a quiet environment or control over how loud the others they live with are. Please do not use any of the disruptive features such as the "Raise Hand" feature during the Zoom meeting; if you need my attention for whatever reason, use email, as I will constantly check it during the exam.

The exam is open note, open keyboard (since you need to type your answers somehow), etc. I might also recommend that you have scratch paper ready, just in case. You are allowed to access slides and notes from online, such as the slides from Canvas. **You are not allowed to outright search for answers online, e.g. with a search engine such as Google or DuckDuckGo. You are not allowed to search directly for answers to the questions on the exam, and you are not allowed to copy significant portions of code/answers from the Internet. You are not allowed to use "tutoring" services such as Chegg. You are not allowed to ask each other for help. As stated in the syllabus, cheating on exams will be rewarded with an F in the entire course.** You are allowed – and for some problems, *advised* – to use an editor such as the Mu editor or Python IDLE, so do make sure that you have an editor that you are comfortable with. You are ***not*** allowed to consult me or the TAs for hints on the correct answers, but you *are* allowed to ask *me* for clarifications on the questions, over email.

If you finish the exam early, leave the Zoom meeting; no need to let me know. Zoom generates a log of all of those who enter and exit the meeting.

If you want, you can set a virtual background on Zoom.

If you are having connection issues due to being in the Zoom meeting, please shoot me an email immediately (yes, during the exam).

# 3 List of Topics

Below are a list of topics that you should make sure that you understand before taking this exam. Not all of these topics will be tested on the exam. As with the previous exam, this exam will have a good amount of emphasis on small coding questions, so do not fall into the trap of only studying the below topics in a manner as if this were a test of rote memorization. Make sure you understand the code that you have written for the programming assignments, and make sure that you understand the examples done during the lectures.

## 3.1 Introductory Lectures (Week #1 and Slide Deck #1)

- `print()`
  - `sep`
  - When whitespace matters.
  - Newline character.
  - Printing with `.format()`.
- Variables. Assignment operator (`=`).
- `round()`
- Math.
  - Addition, subtraction, multiplication, division, division with truncation, modulo, exponentiation.
  - Order of operations.
  - Arithmetic shorthands.
- Using single quotes vs. double quotes and when escaping a quotation mark is necessary.
- Conditional statements.
  - Relational operators: `==`, `!=`, `<`, `<=`, `>`, `>=`.
  - `if`, `elif`, and `else`. Chained conditionals.
  - `pass`
  - Nesting.
  - Logical operators. Short-circuit evaluation.
  - Boolean values.
- String comparisons. What it means for one string to be "less than" another string. Know the rules on slide #30 of slide deck #1.

## 3.2 Slide Deck #2: Iteration with `while` Loops

- Understand all of the examples.
- Structure of a `while` loop.
- Infinite loops.
- `break`
- `continue`

## 3.3 Slide Deck #3: Iteration with `for` Loops

- Understand all of the examples.
- `range()`. "`range()`-based `for` loops".
- When `while` loops are much more convenient than `for` loops.
- `break` and `continue` work the same.
- String traversal (i.e. "non-`range()`-based `for` loops").

### 3.4 Slide Deck #4: Strings

- Understand all of the examples.
- String arithmetic.
- `str()`.
- `len()`.
- Indexing. Negative indices.
- Slicing/substrings/splicing. Step.
- `in` operator.
- Immutability.
- String methods will not be on this particular exam. (Just to be clear, since I kept being asked this the last time I taught this class, `len()` is *not* a method!)

### 3.5 Slide Deck #5: Functions

- Understand all of the examples (that we have done).
- Terminology: caller; passing an argument/parameter or arguments/parameters; returning.
- Defining a function.
- Returning a value.
- Boolean functions. Void functions.
- `None`.

UCDAVIS
COMPUTER SCIENCE