

# Extra Credit

*To be considered for extra credit, you must submit these assignments  
by Sunday March 17 at 11:59 pm*

<b>Overview</b>	<b>1</b>
<b>Part I: Open Book Quiz</b>	<b>1</b>
<b>Part II: Recursion and Dictionaries</b>	<b>2</b>
Recursion	2
Dictionaries	2

## Overview

Instead of offering an automatic curve, we've decided to offer you an opportunity to earn your grade. To receive the curved score, you must complete this assignment. You can choose to complete only part of the assignment, but you will only receive only part of the reward! Your final score will be equal to  $10 * \sqrt{\text{score}}$  times the Extra Credit grade. For example, if you received a grade of 50 and only complete half of the extra credit, your final score will equal 60.4 instead of the full 70.7.

The purpose of this assignment is not to penalize you or make you do more work because of a bad quiz grade, but rather to reinforce concepts that you should know at this point in the course. Regardless of the amount of points you earn toward quiz 1, we encourage all of you to take the time to complete this assignment!

## Part I: Open Book Quiz

*To complete this portion of the extra credit, please email the HTAs so that we can re-invite you to the quiz!*

Once you have access to your quiz, we'll give you unlimited time to go through it with whatever notes or resources you'd like to use. All you have to do is edit your solutions to get them working!

To receive credit for this portion, you must submit:

- Correct solutions for all coding questions on the quiz
- Thoroughly commented code that includes 1) your *original* submitted code in a comment block, 2) explains *why* your original solution was insufficient or incorrect, and 3) *how* you fixed it.

- ❑ Comments about your debugging process. What tools did you use to figure out the problems with your code? (Some examples: Hand simulation, print statements, test-cases, commenting out code, pythontutor.com's stack frame, etc). Please be specific!

## Part II: Recursion and Dictionaries

*Through our review of the quizzes, we found that a lot of people struggled to write good recursive code or struggled to use dictionaries correctly. As such, we thought we'd give you some extra practice*

*To complete this portion of the extra credit, please run `cs4_install extracredit` in your terminal. This will create an `extracredit` folder in your `course/cs0040/projects` folder. Submit by running `cs4_handin extracredit` from this directory.*

### Recursion

In hw 04, we had you implement `map` using recursion. Do the same for `filter` and `reduce` in the file called `recursion.py`

### Dictionaries

In its explanation of dictionaries, the [CS4All textbook uses an example](#) of creating a music recommender program. The goal of the program is to tell a user what kind of music to listen to based off of what they like. For example, if Griffin is a huge fan of the Jonas Brothers, Miley Cyrus, and One Direction, the program would look at its database and find someone with similar preferences. Let's say the program finds Hersh, who likes the Jonas Brothers, Miley Cyrus, and Justin Bieber. Because Hersh likes similar music to Griffin, the program would recommend Justin Bieber to Griffin.

The textbook tries to implement parts of this in different ways, but ultimately uses a dictionary for its database.

[Please read the textbook section on dictionaries](#). It's short and is (hint hint) really helpful! Once you've read it, help us out! We need a program that will tell us which artists should perform in concert together. To do so, fill in `makeConcertLineup.py` following these steps:

- ❑ In `create_listener_map()`, add our staff's music preferences to a dictionary called `listenerMap` and return it (see our favorites on the last page of this assignment!)
  - ❑ Also add yourself to the same dictionary with your music preferences (this is really just for our own fun)
- ❑ In `get_fans()`, use the map returned by `create_listener_map()` to create a dictionary called `fanMap` which maps an artist to all the people who listen to them. Return `fanMap`

- ❑ In `make_lineup()`, use the map returned by `get_fans()` to return a list of all the artists who should perform with `currArtist`
- ❑ Testing is **optional but highly recommended** -- we expect your code to work, and tests are the best way for you to make sure they do!

## Our Music Preferences (Kinda not really)

<b>Hersh</b>	Lauv, Twice, Kendrick Lamar
<b>Griffin</b>	Frank Ocean, J.Cole, Jon Bellion, Mumford and Sons
<b>Joy</b>	Jonas Brothers, Kendrick Lamar, Frank Ocean, John Mayer
<b>Annie</b>	One Direction, Shawn Mendes, Chainsmokers
<b>Professor G</b>	Revivalists, Stevie Ray Vaughan, Massive Attack
<b>Aryan</b>	Coldplay, Ed Sheeran, John Mayer
<b>Alex</b>	Taylor Swift, Ed Sheeran
<b>Ellen</b>	Lauv, Chainsmokers
<b>Irene</b>	Jonas Brothers, Lauv
<b>Jarrett</b>	Led Zeppelin, Imagine Dragons, Mumford and Sons
<b>Tiffany</b>	Ed Sheeran, Post Malone
<b>Milla</b>	One Direction, Ed Sheeran, Andy Grammer
<b>Solomon</b>	Taylor Swift
<b>Pedro</b>	Led Zeppelin, Pink Floyd
<b>Joseph</b>	Lauv, Post Malone, Khalid