



# CS 1110 Spring 2022, Assignment 6: Wherein we throw Ads into the Mix\*

The files you'll need are in this zip file:<sup>1</sup>

[http://www.cs.cornell.edu/courses/cs1110/2022sp/assignments/assignment6/a6\\_skeleton.zip](http://www.cs.cornell.edu/courses/cs1110/2022sp/assignments/assignment6/a6_skeleton.zip) .

Many online content providers interweave advertisements into what's displayed. In A6, you'll add this functionality to the music-serving system we've been developing this semester. Specifically, you'll implement subclasses that extend your A5 work.<sup>2</sup> While **your code must obey the docstrings** (“contract”) we give in the skeletons, and **you may not alter `a6_classes.py` or `a6_music.py`**, otherwise, **the design of your implementation is mostly up to you**.

Given the open-ended nature of this assignment, we aren't providing testing files; testing thoroughly is your responsibility.<sup>3</sup>

Think of A6 as both an ending — you're pulling everything from the semester together (not only does the A6 code we provide still have some of A4/A5's recursion in it, there's even an optional task where you use *your code from way back in A1* to extract real advertisements) — and a beginning: you're ready to take wing on your real programming flights into the wild blue yonder!

## Contents

<b>1</b>	<b>Rules</b>	<b>1</b>
<b>2</b>	<b>Timeline</b>	<b>2</b>
<b>3</b>	<b>Desired behavior of <code>a6_music.py</code>— a means of “downstream” testing/checking</b>	<b>2</b>
3.1	<code>no_browser</code> mode . . . . .	2
3.2	Ed-discussion Ad mode (the default) . . . . .	2
3.3	CS1110-staff-farewell mode — <b>Don't miss it!</b> . . . . .	3
<b>4</b>	<b>Task: design/implement/test subclasses <code>Ad</code> and <code>Loop</code> in <code>a6_subclasses.py</code></b>	<b>3</b>
<b>5</b>	<b>Grand finale and testing</b>	<b>4</b>
<b>6</b>	<b>OPTIONAL Task: Do this assignment end-to-end, bringing your A1 code to bear</b>	<b>4</b>
<b>7</b>	<b>For further Song processing with python: <code>spotipy</code></b>	<b>4</b>

## 1 Rules

See Sections 1.1-1.3 of [Assignment 1](#)<sup>4</sup>, Sections 3.1-3.2 of [Assignment 2](#)<sup>5</sup>, and Sections 3.1.3 and 3.2 (pre-submission checklist) of [Assignment 3](#)<sup>6</sup>. If you have questions about whether some Python is “legal”, ask on Ed Discussions.

---

\*Authors: Lillian Lee.

<sup>1</sup>If you are using a Windows computer: when you download a zip file, it is represented in File Explorer as a folder with a little zip on it. That is not a real folder, which is why you have may have had trouble saving changes to files in it! Right-click on the zip file and select “Extract All”.

<sup>2</sup>Technically, you'll be extending *our* A5 work, since we're providing a slightly modified version of the A5 solutions in the skeletons. But you students and we staff are all in this together!

<sup>3</sup>In order to reduce grading time, we are not requiring you to submit your testing files.

<sup>4</sup><https://www.cs.cornell.edu/courses/cs1110/2022sp/assignments/assignment1/a1.pdf>

<sup>5</sup><https://www.cs.cornell.edu/courses/cs1110/2022sp/assignments/assignment2/a2.pdf>

<sup>6</sup><https://www.cs.cornell.edu/courses/cs1110/2022sp/assignments/assignment3/a3.pdf>

## 2 Timeline

1. If you are partnering,<sup>7</sup> **form your group on CMS before submitting anything**. This links your submission “portals”.
2. By 2pm Ithaca time on Mon May 9, submit your *partial* progress on CMS.<sup>8</sup> It is OK if you haven’t finished working yet.<sup>9</sup>
3. By **11:59pm (Ithaca time) on Mon May 9**, make your final submission, and do steps 1-3 in the “Updating, verifying, and documenting assignment submission” section of <https://www.cs.cornell.edu/courses/cs1110/2022sp/resources/cms.html>

## 3 Desired behavior of a6\_music.py— a means of “downstream” testing/checking

Once you complete a6\_subclasses.py, thus extending the classes Loop and Song already implemented in a6\_classes.py, a6\_music.py will become functional, since:

- lines 79-80 of a6\_music.py create Ads, where subclass Ad extends Song.
- line 112 creates and plays an AdLoop, rather than a Loop as was the case in A5. Subclass AdLoop extends Loop.

a6\_music.py can be run in several modes, controlled by whether you supply extra information at the command line, explained below. You can do some testing by seeing whether your a6\_subclasses.py enables a6\_music.py to reproduce the sample output below.

### 3.1 no\_browser mode

For testing purposes where you don’t want to be slowed down by browser windows constantly popping up, run with the additional *command-line argument* no\_browser, like this:

```
python a6_music.py no_browser
```

Then, “playing” a Song will not open a url, but just print the url out.

### 3.2 Ed-discussion Ad mode (the default)

“Ads” are really public-service announcements drawn from CS1110 SP22’s very own “External Events” Ed Discussion megathread<sup>10</sup>. Sample output below<sup>11</sup>: Ad titles and “artists” are highlighted in magenta.

```
[ljl2@semester's_end] python a6_music.py
Select from the following choices of Mixes
[.....menu omitted.....]
Which number do you want? 0
Starting Loop LCD Soundsystem
Hit Enter/Return to start the next Song, Dance Yrself Clean by LCD Soundsystem, or "q" to quit:
[opens https://www.youtube.com/watch?v=9ZNkPA_zUd4]
Hit Enter/Return to start the next Song, AI-Learners by [Advertiser], or "q" to quit:
[opens https://edstem.org/us/courses/19140/discussion/1050756?comment=3236108]
Hit Enter/Return to start the next Song, Losing My Edge by LCD Soundsystem, or "q" to quit:
```

<sup>7</sup>Reminder: Both parties need to act on CMS in order for the grouping to take effect. See the “How to form a group” instructions at <https://www.cs.cornell.edu/courses/cs1110/2022sp/resources/cms.html> .

<sup>8</sup>And, as usual, perform steps 1-3 in the “Updating, verifying, and documenting assignment submission” section of <https://www.cs.cornell.edu/courses/cs1110/2022sp/resources/cms.html>

<sup>9</sup>The 2pm checkpoints provide you a chance to alert us if any problems arise, and for partners to become aware if something has gone wrong with the CMS grouping process (via us emailing people who haven’t yet submitted — if your partner submitted but you got the warning email, that means CMS doesn’t think you’ve grouped!). Since we’ll send a reminder around 2pm to all those who haven’t submitted yet, do not expect that we will accept work that doesn’t make it onto CMS on time, for whatever reason. There are no so-called “slipdays” and there is no “you get to submit late at the price of a late penalty” policy. Of course, if some special circumstances arise, contact the instructor(s) immediately.

<sup>10</sup><https://edstem.org/us/courses/19140/discussion/1050756>

<sup>11</sup>Our a6\_music.py does not ask the user about shuffling.

```

[opens https://www.youtube.com/watch?v=6xG4oFny2Pk]
Hit Enter/Return to start the next Song, Cornell Fashion Collective Runway Show by [Advertiser], or "q" to quit:
[opens https://edstem.org/us/courses/19140/discussion/1050756?comment=3204940]
Hit Enter/Return to start the next Song, Dance Yrself Clean by LCD Soundsystem, or "q" to quit:
[opens https://www.youtube.com/watch?v=9ZNkPA_zUd4]
Hit Enter/Return to start the next Song, Hearsay A Cappella Concert by [Advertiser], or "q" to quit:
[opens https://edstem.org/us/courses/19140/discussion/1050756?comment=3204481]
[.....some more interactions omitted.....]
Hit Enter/Return to start the next Song, LSC Study Partner Matching by [Advertiser], or "q" to quit:
[would play https://edstem.org/us/courses/19140/discussion/1050756?comment=2393765]
Hit Enter/Return to start the next Song, Dance Yrself Clean by LCD Soundsystem, or "q" to quit:
[opens https://www.youtube.com/watch?v=9ZNkPA_zUd4]
Hit Enter/Return to start the next Song, AI-Learners by [Advertiser], or "q" to quit: q
The AI-Learners ad returned after the last ad in Ad.all_of_em, "LSC Study Partner Matching", was displayed.
Exiting this Loop

```

### 3.3 CS1110-staff-farewell mode — Don't miss it!

“Ads” are heartfelt messages from the course staff to you. Run like this:

```
python a6_music.py staff_msgs
```

or this

```
python a6_music.py staff_msgs no_browser
```

## 4 Task: design/implement/test subclasses Ad and Loop in a6\_subclasses.py

Carefully read the docstrings, instruction comments, and code in a6\_subclasses.py, a6\_classes.py, and a6\_music.py. Then complete Ad, a subclass of Song, and AdLoop, a subclass of Loop.

1. AdLoop is a challenge, so start thinking about it early, even if you decide to implement Ad first.

```

class AdLoop(Loop):
    """
    Like a Loop, except that it plays an Ad after every Song. The Ads should
    appear in the order they occur in Ad.all_of_em.

    Example: suppose the source Mix has only two Songs, s1 and s2, and
    Ad.all_of_em has Ads a1, a2, a3, a4, a5.
    Then, the sequence of Songs occurring if one ran the play() method multiple
    times is:
        s1 a1 s2 a2 s1 a3 s2 a4 s1 a5 s2 a1 s1 a2 ... # typo fixed 4/30 (s2->s1)

    Instance attributes:
        Includes those of Loops.

    """

```

Recall that the Loop instance attributes include a list of Songs `slist` and a `next_pos` attribute indicating the next Song to play in the Loop's `slist`.

- (a) One reasonable idea would be to make AdLoops have Ads in their `slists`. This is legal because Ads are Songs via the subclass relation. But if you take this approach, be careful how you construct the `slist` if the number of Songs in the source Mix is different than the number of Ads in `Ad.all_of_em`. Partial credit can be granted for solutions that, by design (that you document), change the problem to a simpler one. The amount of credit will depend on how drastic the simplification is.<sup>12</sup>
- (b) A different idea is to leave `slist` as a set of non-Ad Songs. In this approach, you must figure out when to “display” an Ad vs. a Song, and which Ad to display.

2. Remember, your code must obey and implement the given docstrings.
3. Remember to include the requested JUSTIFICATION comments; they contribute to the assignment grade. We suggest 1-3 sentences per addition/decision. More than 3 per addition/decision is probably excessive.
4. Just so you know: we wrote a while-loop somewhere in our solution.

<sup>12</sup>For example, only including the same Ad every time is more drastic than including only the first few Ads in `Ad.all_of_em`, but preserving the order correctly. Thus, the latter will be looked on more favorably by the graders.

## 5 Grand finale and testing

Run `a6_music.py` and observe whether its behavior is correct, per Section 3.

## 6 OPTIONAL Task: Do this assignment end-to-end, bringing your A1 code to bear

If you want to test yourself as to whether you could have done this A6 project “end-to-end”, including developing your own test code, there are instructions in lines 64-72 so in `a6_music.py`. The main job is to complete `ed22sp_processing.py`, which extracts information from the html behind our Ed Discussions External Events megathread. That’s a string processing task which good ol’ `scoop()` and related functions from A1 are perfect for.<sup>13</sup>

## 7 For further Song processing with python: spotipy

Want to try your python skills on data from Spotify? There’s a python package called, amusingly, *spotipy* that you might be interested in. Check out the [A6 from CS1110 SP19<sup>14</sup>](#), clicking on the "Optional: Clustering Songs" section header, to learn more.

---

<sup>13</sup>See, the whole semester comes together now!

<sup>14</sup><https://www.cs.cornell.edu/courses/cs1110/2019sp/assignments/a6/>