

**SI SESSION PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| SI Leader: | Alex Iacob | Session Date: | 10/16/22 |
| Week #: | 10 | Session Letter: | A |
| Course & Section: | CSCI 141 Section 7 | Course Instructor: | Steele |
| Planning Date: | 10/16/22 | Planning Time: |  |

**Beginning reminders:**

1. Is the room set up in a way conducive to collaborative learning?
2. Is the agenda posted to the board for participants to see?
3. Do you have your attendance sheet up to record your attendance?
4. Do you have any other documents/resources up and ready to go for your session?

If you are all set with the reminders, then go have fun and good luck!

**Main concepts student should feel more comfortable with:**

* Dataclasses
* Dictionaries
* Time complexities

|  |  |  |
| --- | --- | --- |
| **Topics to cover** | **Process to use\*:** | **Time** |
| **Opener:** Spotify game again |  | 3-5 |

|  |  |  |
| --- | --- | --- |
| Dataclasses | Create a recipe dataclass  Should look something like:  From Dataclasses import dataclass  @Dataclass  Class Recipe:  Name: str  Ingredients: list()  Steps: list()  Servings: int  Up for interpretation, but it should look something like that  Make a recipe data objects, for example  instantRamen = Recipe(‘ramen’, [‘instant rame]) | 10-15 |
| Dictionaries | Dictionaries are fast, that’s why we use them. They function under key: value pairs, where a key is related to a value. If we know the key  We can create them in two ways  dictionary = {}  dictionary = dict()  If we ever want to add something to a dictionary, we can:  dictionary[key] = value  But how can we check if something is already in a dictionary  If key not in dictionary:  # do stuff | 10-15 |
| Dataclasses in Dictionaries | How would we make a cookbook?  We can make a bunch of recipe objects and add them to the dictionary | 20-25 |
| **Closer:** Sneakpeek of stacks and queues | Might not get to this, but I can mention the whole linked structure concept with the stacks and queues  Stacks are First in Last out  The available functions are pop(), push(), peek(), make\_new\_stack()  Queues are First in First out  The available functions are enqueue(), dequeue(), front(), and make\_new\_queue() | Remaining time |

*\*Possible processes: puzzles, informal quiz, think-pair-share, paired problem solving, graphic organizers, cheat sheets, collaborative questioning, student summaries, reviewing notes, work at the board, vocabulary…*

**Ending reminders:**

1. Did you check everyone in?
2. Did you remind everyone of the next session and any upcoming tests or quizzes or due dates?

**What is one thing you want to emphasize during this session?** Please be specific.

* Utilize your resources, they’re useful.

**After session thoughts:** How did the session go? Is there anything you would like to keep/drop/change for next time and how?

* Austin came to observe my session, it was pretty nice having him participate a bit. Also I should have guessed, but my one student for my Sunday sessions speeds through everything, so I end up having to do far more material just because he is pretty ahead and comes to sessions for fun.

**Bi-Weekly Question:** Did your class have a test or quiz this week? If so, how did you help students prepare? If not, what should the students be focusing on?

* The material that they must cover is going to be for their next exam during week 12. I’m trying to get a study sheet going for them and try to email it out soon.