

**SI SESSION PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| SI Leader: | Alex Iacob | Session Date: | 11/27/22 |
| Week #: | 15 | Session Letter: | A |
| Course & Section: | CSCI 141 Section 7 | Course Instructor: | Steele |
| Planning Date: | 11/27/22 | Planning Time: | 12:45 – 1:00 |

**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:**

* Trees
* Linked structures

|  |  |  |
| --- | --- | --- |
| **Topics to cover** | **Process to use\*:** | **Time** |
| **Opener:** Talk a bit | Nothing special, just try to talk with my students | 2-3 |

|  |  |  |
| --- | --- | --- |
| Trees | The most common tree you will find is a binary search tree, where it is a linked structure that looks something like:  Class Tree:  Value: Any  Left: Union[Tree, None]  Right: Union[Tree, None]  Given this, we can create a binary search tree from a given array of numbers  [6, 8, 4, 5, 2, 1, 3, 7, 9, 10] | 25-30 |
| Hashing sneak peek | Why do we want to hash stuff?  Python uses dictionaries, which are fast, hashing lets us make a key from a given value so we can shove it into a dictionary.  How does this hash function work?  It is dealer’s choice; we can start off simply by just taking the order of the first and last characters of a string and just combine those.  What does this hash table look like?  We have the hash value as a key, then the value as the value (great wording I know) We take the hashed value, then mod it by the size of the table. For example, the word *cat* can produce a hash value of 1, which means we insert it at index 1 in the hash table.  But what happens if two strings give us the same hash value?  We can use something called chaining or open addressing  Chaining is when we treat the value section as an array, then just append the given value.  Open addressing is when we get a repeating hash value, try to insert it, fail, then move to the next open position. | 25-30 |
| **Closer:** |  |  |

*\*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.

* Take care of yourself

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

* Got another 1-on-1 with a top student, so most of the session was him just zooming through everything, so we did the old faithful and just planned for more stuff and talked about things outside of classes.

**Bi-Weekly Question:** If it was week 1 again, what might you do differently?

* I would try to force myself to not be as awkward with new students. I was pretty quiet for the first few weeks and I lost a lot of marketing because of catching Covid week 2. I would try some more aggressive marketing, like talking to more students.