

**SI SESSION PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| SI Leader: | Alex Iacob | Session Date: | 11/4/21 |
| Week #: | 11 | Session Letter: | B |
| Course & Section: | CSCI 141 Section 3 | Course Instructor: | Steele |
| Planning Date: | 11/4/21 | Planning Time: | 10:30 – 11:15 |

**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:**

* Basic linked list data structures

|  |  |  |
| --- | --- | --- |
| **Topics to cover** | **Process to use\*:** | **Time** |
| **Opener:** Announcements | Exam week 12 on Monday Lab 8 due next Tuesday  HW 11 due Friday (Post session: Extended to Sunday) | 3-5 |
| Remind them about linked list | In python, linked lists don’t have as much of a use, because you can keep on adding things to arrays and whatnot, however in other languages, the array space you get when you declare the list is the space you get. Linked lists allow you to keep on adding things to said list. (Credits to Nicholas for explaining this concept) | 5-15 |
| Find the largest thing in a linked list | First split the class in two and give each the prompt. One group must recurse over the linked list and find the largest value via recursion and the other has to do the same thing iteratively. I predict that they are going to need help with achieving a working solution, so I wrote something up beforehand:  Recursively:  if lnk is None:  return highest  elif lnk.value > highest:  return highest\_value(lnk.rest, lnk.value)  else:  return highest\_value(lnk.rest, highest)  Iteratively:  highest = 0  while lnk is not None:  if lnk.value > highest:  highest = lnk.value  lnk = lnk.rest  return highest | 30-40 |
| **Closer:** | Shout out the SI program and remind them to keep working on the cheat sheet. | 3-5 |

*\*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 closing tips/strategies would you like to emphasize through sessions this week?** Please be specific.

* Make a cheat sheet for the exam

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

* It had a higher turn out rate than most others mainly because linked lists are a difficult concept to wrap your head around.

**Bi-Weekly Question:** Have you completed your peer observation yet? Have you been observed? What have you gained from the feedback/ experience that has helped you in your own sessions?

* I observed one session so far and the main thing that I got from it was wait time. Prachi’s wait time is something that I’m working towards achieving. The main things that I got from being observed is to try to not structure my sessions as another recitation and instead focus on getting them to talk with each other more.
* Post session: I did get observed by Lauren A. today, she mostly said that the things I did were pretty decent and that I’d get an email soon.