

**SI SESSION PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| SI Leader: | Alex Iacob | Session Date: | 10/17/22 |
| Week #: | 10 | Session Letter: | B |
| Course & Section: | CSCI 141 Section 7 | Course Instructor: | Steele |
| Planning Date: | 10/17/22 | Planning Time: | 4:50 - 5 |

**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:**

* Stacks
* Queues
* Linked Structures

|  |  |  |
| --- | --- | --- |
| **Topics to cover** | **Process to use\*:** | **Time** |
| **Opener:** Ask everyone how it’s going and announcements | Simple conversation starters since the lecture this week was a miss.  HW due Friday  Lab due Tuesday  Project assigned during week 12  Exam 2 on week 12 | 3-5 |
| Stacks | Stacks are nice  They follow a format that looks like:  @dataclass  Class Stack:  Top: value  Next: Union[“Stack”, None]  *Use cards or markers for this*  The functions we have available are:  Pop() – removes the thing at the top of the stack  Push(item) – pushes the item to the top of the stack  Peek() – returns the thing at the top of the stack  Make\_new\_stack() – makes an empty stack  Stacks are created in a First in Last out fashion. | 25-30 |
| Queues | Queues are easier to visualize  They follow a format that looks like:  @dataclass  Class Queue:  Front: value  Next: Union[“Queue”, None]  *Use cards or markers for this*  The functions we have available are:  Dequeue() – removes the thing at the front of the queue  Enqueue(item) – adds the item at the back of the queue  front() – returns the thing at the front of the queue  Make\_new\_queue() – makes an empty queue  Queues are created in a First in First out fashion. | 25-30 |
| Time complexities and exam questions | Adding or removing anything at the front/top of the linked structure is constant time, while recursing over the entire structure is linear time.  There can and will be questions about this on the exam  I’ll also pull up some questions from the exam and slightly change them. | Remaining time |
| **Closer:** Probably no closer because this content is hard | Probably got going to get to this, but just assuring people that they are doing okay and that this is a difficult concept usually eases stresses. |  |

*\*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 are useful.

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

* I had a few students show up, and they were confused with the whole content since Steele decided to off the rails. I could confidently say that *maybe* 3 people paid attention for more than 15 minutes.
* Me: “Hey guys what do you think should go next to this queue object?”  
  A student: “Barbe-“

**Bi-Weekly Question:** How are you doing at this point in the semester? Is there anything that you’re looking forward to?

* Like a typical CS student, I have a dwindling state of being, but that’s fine, it is what it is, but I must finish strong. The main things I’m looking forward to is the semester to finish and to have that time between semesters to do literally nothing except hang out with friends all day without the stress of classes.