It's been a long time since our last CD at the end of September. During that time, we have discussed a lot of "smallish" topics. These include:

- **Object Oriented Programming**
  - Class
  - Object
  - Magic Methods/functions
- **Abstract Data Types (ADTs)**
  - Stack
  - Queue
  Differences in implementation and how this effects Big-Oh notation (runtime performance)

This competency demo is looking at questions that get at a general understanding of these topics. While there may be a few very specific details we would be looking for in questions/answers, most are looking at do you understand the big picture and not, do you understand the nitty-gritty details.

Sample questions could include:

**Object Oriented Programming**

1. What is the purpose of writing/using objects in a programming language?
2. What is the difference between a class and an object?
3. What kind of analogy could you use to explain class vs. object with students?
4. Python has a series of "magic methods." What are these used for? Give an example of one and where/how it is used?
5. Why would programming objects be more difficult without the magic methods?

**Abstract Data Types**

1. What is the main concept of the ABSTRACT Data Type? (emphasis added by me so that you have an understanding of what I am looking for in my answer to this question).
2. When were you using ADTs in the Fundamentals of Programming course without even knowing it at the time?
3. What is a _________? Give an example of where a programmer might use one (that isn't from our AI search discussion) [The blank could contain Stack, Queue, or PriorityQueue]
4. One of the functions for a Stack is __________. What is the expected Big-Oh notation for this function? Why? [Functions that could go in the blank are push, pop, size, isEmpty]
5. One of the functions for a Queue is __________. What is the expected Big-Oh notation for this function? Why? Is your answer implementation dependent? Why? [Functions that could go in the blank are enqueue, dequeue, size, isEmpty]