Aaron Jones

3/18/19

CS 241 Final Exam

Provide a written answer in your own words for each question using proper grammar and spelling. Each answer should be limited to one or two paragraphs. You must type your response. Include the exam question before each of your responses. Your name should be at the top of the document. You may use online resources and your class notes. You may discuss the questions with other students in the class, but you may not write responses together or share responses. You must submit a PDF copy of your response into I-Learn by midnight on the last day of class.

1. What is the difference between a class and an object?

The difference between a class and an object is that a class has a blueprint that can be defined and used to make up objects whereas an object is an instance of a class and can be used to set the variable name for the class provided. For example, if you were to have a class named student. Then you would set the object name to s and assign the s variable to the student class. You can also use the reference of the variable s to call multiple functions inside of the student class by using the dot operator (ex. s.class_name()).

Additionally, a class has many attributes that can initialize objects and can hold different values. A class has an initializer that can set an objects name to a variable and can contain multiple values that can be passed to specific functions inside of the same class.

2. Give an example of an IS-A (inheritance) relationship and a separate example of a HAS-A (composition) relationship within the fields of either mathematics, biology, or engineering. The example should be different from one given in class. Explain why these are appropriate examples.

An example of an 'IS-A' relationship in terms of biology could be the genetic connection between the parent and child. The parent has many similarities to the child, and the traits that are passed down through the generation are considered common genetic traits. Whereas the child may have some different qualities that are unique to the child, for example, this could be the child's hair color or the child's likes and dislikes that are unique to him and him alone.

An example of a 'HAS -A' relationship in terms of biology could be a person having a virus, and that virus could take on many traits from the person, but these would not be unique to the virus they would be considered initial state values. When the person takes medications, then these values could be overridden, and the person could move away from having a virus.

3. Define inheritance and explain how it was used in the Skeet project.

Inheritance is the use of one class that can obtain specific characteristics from another class. This method was used in the Skeet Project with five different classes and created a hierarchy to form with the flying object's class being the top. The flying object's class took in values from the velocity and the point class and created multiple abstract methods to be used in the bullet and target classes. The target class was another abstract class that held initial state values and passed them on to the different types of targets that had their unique characteristics.

4. Define polymorphism and explain how it was used in the Asteroid project.

Polymorphism in the literal sense means to take many different forms. In the Asteroids project, we used polymorphism for the different types of asteroids. The asteroid class had the function hit which is redefined in each of the asteroid-derived classes — creating multiple sub-classes that take on the same function names from there parent class. This is called method overriding and is used when the parent class functionality for the specific function does not fit the situation that is needed for the derived child class.

5. Explain the benefit of using a Python property in a class and how this supports the principle of encapsulation?

Using a python property in a class gives the user the ability to edit the class and add a value constraint to that same class without changing the code. This way the program can now be backward compatible. This property supports the principles of encapsulation by preventing any user from accessing the code because in a sense the program is now in a private state. Granted the program can still have the ability to become public, but that would mean you would have to change the variable name. In a large program, this would be extremely difficult because you would have to change the variable name a countless number of times so that it could work.

6. Compare and contrast a Stack and a Queue. In your response, give an example of when a Stack should be used and an example of when a Queue should be used.

The differences between a stack and a queue are that a stack uses the last in first out method (LIFO) and a queue uses the first in first out method (FIFO) both of which are used to access and add data elements. A queue has both ends that are open for enqueuing and dequeuing data elements whereas stacks only have one end that is open for pushing and popping data.

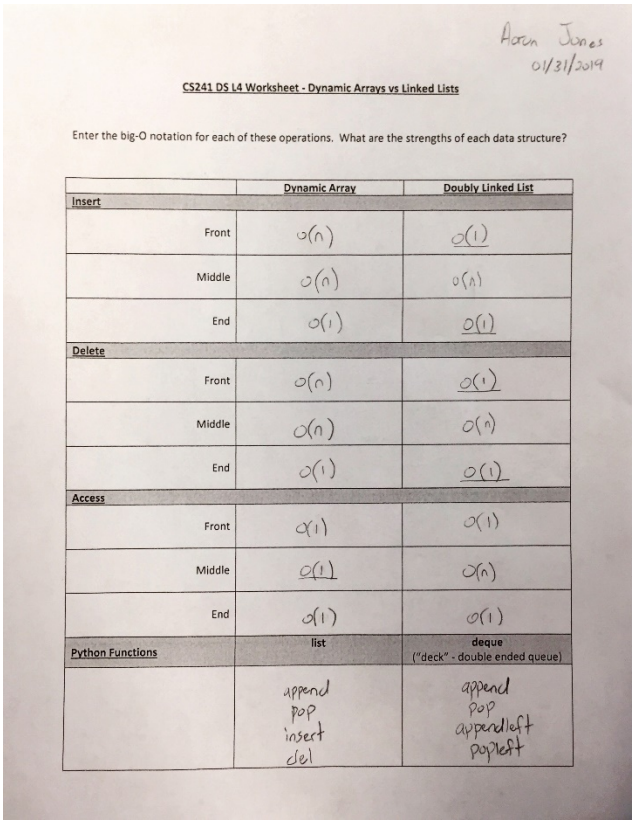
An example of a stack would be a stack of waffles. In this example, you would have the first waffle at the bottom of the stack, and as you make more waffles and add them to the top of the stack, they will then be taken from top to be eaten. As for the last waffle, it will be the last one consumed from the stack.

An example of a queue would be a telephone service center. In this example, you would have a user call in, and their phone call would be answered first, and every customer that calls after while that same call is being answered will have to wait because of the first in first out rule.

7. Compare and contrast a Dynamic Array and a Linked List. In your response, compare and contrast the performance for accessing, inserting, and removing items from either data structure.

The differences between a Linked List and a Dynamic Array is a Linked List is a collection of unorganized elements that are linked together also considered to be nodes. A dynamic array is a better-structured collection of similar data elements. When the user searches through a linked list, they must look through the entire list to find the exact point to access O(n).

Whereas in a dynamic array you can search through an index and assign a variable name with the location within square brackets O(1). When you want to insert an item into a linked list the user can append a new element to the end or the beginning of the linked chain O(1). In a dynamic array, you can only add to the end of the list and not the beginning O(n). Alternatively, if you wanted to delete from the end or beginning of a linked list, then it would be O(1). The user cannot do this as effectively with a dynamic array because you can only pop or delete items from the end of the array O(n). Overall it is better to use a Linked list because you can add and remove items more efficiently than a dynamic array, but in a like manner it is better to access elements inside of a dynamic array then it is to a linked list.



Aaron Jones
01/31/2019

CS241 DS L4 Worksheet - Dynamic Arrays vs Linked Lists

Enter the big-O notation for each of these operations. What are the strengths of each data structure?

	Dynamic Array	Doubly Linked List
Insert		
Front	O(n)	O(1)
Middle	O(n)	O(n)
End	O(1)	O(1)
Delete		
Front	O(n)	O(1)
Middle	O(n)	O(n)
End	O(1)	O(1)
Access		
Front	O(1)	O(1)
Middle	O(1)	O(n)
End	O(1)	O(1)
Python Functions	**list**	**deque** ("deck" - double ended queue)
	append pop insert del	append pop appendleft popleft

8. Define recursion and explain why it is used in the Merge Sort algorithm.

Recursion is a function that operates on a terminating case and then works on a smaller subdivided input. This input is inside an array and starts at the left side. When the function returns, it will continue to keep working on smaller parts and potentially can work on the right half or move up to the beginning and recurse on the more substantial portion of the input. The merge sort algorithm uses recursion through continually splitting a list into halves. If the list has more than one item, the algorithm will cut the list and use recursion to invoke a merge sort on both halves. When both halves have been sorted, then the merge will be performed. Merge Sort splits the array in half and starts by sorting the left half. When the sorting is finished, the algorithm will then continue to sort the right half. In conclusion, there will be a merge between the right and left halves of the list.

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

9. Explain why the worst-case performance of Selection Sort is $O(n^2)$.

Selection sort selects the smallest unsorted item and swaps the value to the next position. Additionally, while this is occurring it will keep checking the remaining unsorted array for potential values to be moved. In other words, the first ‘for’ loop must check each element of the array O(i) by starting at 0 and continue moving towards n. In a like manner, the second ‘for’ loop will perform a similar process O(i). Although these processes are similar in performance the difference is that the ‘if’ statement inside of the

second ‘for’ loop will be O(1) because of the comparison to the smallest element in the array.

$$\sum_{i=0}^{i=n} O(i) = 1 + 2 + 3 \ldots + (n-1) = O(n^2)$$

10. Describe the differences between a binary tree, a binary search tree, and a balanced binary search tree.

A binary tree can have only two child nodes. Inside of each node and there is no order in the way that each node is organized throughout the tree. Additionally, when a node does not have a child node, it is considered a leaf node.

A binary search tree is derived from a binary tree because it still follows the rule of a certain number of child nodes, but the difference between a binary search tree and a binary tree is that there is a particular ordering on how the nodes are organized. The nodes to the left have values that are less than the values of the nodes to the right which have greater values.

A balanced binary search tree or AVL is necessarily the same thing a binary search tree, but the difference between the two is the performance and the fact that a balanced binary search tree keeps track of the balance for each node in the tree. This is done by checking the heights for the left and right subtrees for each node.