### Name: Tradd Schmidt

1. Copy the entire main() function into the box below

|  |
| --- |
| def main():  print("Make a linked list in honor of Donna Schmidt")  mom = LList((8, 21, 1962)) # \_\_init\_\_ was called and self.item and self.link was changed  print("Make a linked list in honor of Jim Carrey")  truman = LList((6, 17, 1962)) # \_\_init\_\_ was called and self.item and self.link was changed   print("Make a linked list for a fictional character")  fictional = LList(((8 + 6) // 2, (21 + 17) // 2, 1962)) # \_\_init\_\_ was called and self.item and self.link was changed    # Printing the Birthdays  # ---------------------------------------------------------------------------------------  print("\n")  print("Printing Donna's birthday")  for item in mom: # \_\_next\_\_ was used  print(str(item) + " ",)  print("\n")  print("Printing Jim Carrey's birthday")  for item in truman: # \_\_next\_\_ was used  print(str(item) + " ",)  print("\n")  print("Printing the fictional character's birthday")  for item in fictional: # \_\_next\_\_ was used  print(str(item) + " ",)   # Changing the year of the fictional character  # ---------------------------------------------------------------------------------------  year = fictional.\_find(len(fictional) - 1) # \_find was used  year.item += 100 # item was changed  print("\n")  print("Printing the fictional character's revised birthday")  for item in fictional: # \_\_next\_\_ was used  print(str(item) + " ", )   # Deleting the date of the fictional character's birthday  # ----------------------------------------------------------------------------------------  fictional.\_\_delitem\_\_(1) # \_\_delitem\_\_ was used  print("\n")  print("Printing the fictional character's revised birthday")  for item in fictional: # \_\_next\_\_ was used  print(str(item) + " ", ) |

For every line the function above, list every method of every class which that line called, and every instance variable that was changed. Include this information as comments in the box above.

1. Write a paragraph reflecting on what you feel that you understand about **interacting classes** at this point, and what questions you still have about them.

|  |
| --- |
| I feel like I have a good grasp on interacting classes. They are simple in the fact that one is usually just obtaining or manipulating data from another. However, I do not know if this is the only thing that interacting classes do. A question I still have is why you separate some classes into different modules and why you keep some classes in the same module. |

1. Write a paragraph reflecting on what you feel that you understand about **interacting linked lists** at this point, and what questions you still have about them.

|  |
| --- |
| I believe I have a good grasp on these as well. The biggest thing that one has to remember is that when an element of a linked list is deleted, the link of the previous element needs to be updated to point to a new data point, or it will be deleted. My question is can an element of a linked list link with something other than another list node? |

1. Write a paragraph reflecting on what you feel that you understand about **interators** at this point, and what questions you still have about them.

|  |
| --- |
| Iterators are pretty straightforward for linked lists. There has essentially a long string connecting all of the listnodes. The trick is you have to follow the links until you have reached the specific element of the list that you were trying to reach. I do not really have any lingering questions about iterators. |