

## Iteration 1 – EventManager Class

(Fall 2024)

### Instructions – EventManager:

In the file named EventManager.py, you will find an EventManager class, which will handle the lists of Contact and Event objects for the program. The class will have two methods already defined:

- `def _sort_contacts(self)`
- `def _sort_events(self)`

These methods are used to sort the contact list and event list respectively. Do not modify these methods, unless you need to change attribute names for consistency.

In the EventManager class, add code to handle the following:

1. Create a constructor that accepts no arguments (apart from `self`) but initializes the following attributes:
  - a. A list of Event objects, set as an empty list.
  - b. A list of Contact objects, set as an empty list.
  - c. A list of EventAttendee objects, set as an empty list.
  - d. An Event UID, set to the integer 0. *(This will be incremented to determine the unique ID for each new event.)*
  - e. A Contact UID, set to the integer 0. *(This will be incremented to determine the unique ID for each new contact.)*
2. Using the `@property` and `@[object].setter` decorators, create getters and setters for each of the five attributes defined in the previous step.
3. Define a method called **add\_event** which accepts a dictionary object as an argument. This dictionary represents the new event to be added. This method should:
  - a. Create a new Event object (using the dictionary object as input) and append it to the EventManager's list of events.
  - b. Increment the value of the EventManager's Event UID attribute, so that the next event added will have a different unique ID.
  - c. Sort the EventManager's list of events by calling the pre-existing `_sort_events` method.
4. Define a method called **add\_contact** which accepts a dictionary object as an argument. This dictionary represents the new contact to be added. This method should:

- d. Create a new Contact object (using the dictionary object as input) and append it to the EventManager's list of contacts.
  - e. Increment the value of the EventManager's Contact UID attribute, so that the next contact added will have a different unique ID.
  - f. Sort the EventManager's list of contacts by calling the pre-existing `_sort_contacts` method.
5. Define a method called **is\_attending** which accepts a Contact object and an Event object as arguments. This method should:
  - a. Iterate through the EventManager's list of EventAttendees to determine whether the given contact argument is attending the given event argument.
  - b. If the contact is attending the event, the method should return True.
  - c. If the contact is not attending the event, the method should return False.
6. Define a method called **add\_event\_attendee** which accepts a Contact object and an Event object as arguments. These are the contact and event that will be paired together in the new EventAttendee object. This method should:
  - a. Check to see whether the given contact is already attending the given event.
  - b. If the contact is not already attending the event, create a new EventAttendee object (using the Contact and Event objects as input) and append it to the EventManager's list of EventAttendees.
  - c. If the contact is already attending the event, do nothing.
7. Define a method called **uid\_to\_event** which accepts an integer (representing a unique ID) as an argument. This method should:
  - a. Iterate through the EventManager's list of events to determine whether any event has the given integer as a unique ID.
  - b. If a matching ID is found, the method should return the Event object associated with it.
  - c. If no matching ID is found, the method should return None.
8. Define a method called **uid\_to\_contact** which accepts an integer (representing a unique ID) as an argument. This method should:
  - a. Iterate through the EventManager's list of contacts to determine whether any contact has the given integer as a unique ID.
  - b. If a matching ID is found, the method should return the Contact object associated with it.
  - c. If no matching ID is found, the method should return None.