Andy Fleischer

Prof. Karla Fant

CS202

2 July 2021

Program 1 UML Writeup

In this UML diagram (submitted separately), a solid line represents an inheritance ("is a") relationship and a dashed line represents a containing ("has a") relationship. I chose to have all three of my event types (gym, restaurant, and party) be their own classes with private data unique to them, and push up the commonalities like a location (address) and a date (event) by having the three event types derive from class Event. I chose to have Event itself derive from Address since the location of an event is pretty important and will likely be accessed often, so a containing relationship would require too much use of wrappers. I made Address its own class to wrap all related data together and manage locations all in one place. One decision I struggled with was whether "date" should be a whole class that event contains or multiple-inherits, but I decided both of those would probably increase complexity, so an Event itself manages date and time.

For the data structures, I went with the design that has nodes derived from what they manage, and a separate data structure class (CLL and LLL_Array) to manage those nodes in the style of the given structure. The CLL uses C_Nodes which are Gyms, the LLL_Array uses Nodes which are Restaurants, and the Party event will be managed with a std::vector. Both of my custom data structures have required functions for insert, remove, display, and remove_all. Although not shown in the diagram, there will be private recursive functions that do the recursive tasks.