In the assignment we were asked to create 4 classes; I created them.

Besides the obvious members for each class, it was mentioned that a pokemon should have members that relate to its state eg. Its faint status, its powerup status, and finally which terrain its on status. I created those too. Also it was mentioned that the way poekmons receive damage is the same for all of them so I created that function in pokemon too. Every other function is self-explanatory.

As for relationships between my classes, ElectricPokemon and PsychicPokemon inherit from Pokemon as it can be seen per UML diagram, they are types of pokemon so it makes sense. For the pokemons to fight, they join the Arena which is another class. They are referred to through a pointer in the Arena class. For this reason, there is an aggregation relationship between ElectricPokemon and the Arena likewise from PsychicPokemon to Arena. The multiplicity for that relationship is 1 to 1. As per the universe that I created, one pokemon from a specific type can be present in the arena at a specific time, the other one must be from another type. Its an aggregation relationship as an arena can exist without any pokemons and just wait for them to be added, it isn’t composed from them only. Also, a pokemon can only fight in one arena at a time.

Regarding visibilities there is an attribute visibility between Arena and ElectricPokemon, and Arena and PsychicPokemon. These visibilities occur due to the pointers Electric and Psychic in Arena.

Regarding access control, since I used public inheritance, ElectricPokemon and PsychicPokemon can access all the public methods of the Pokemon class however they cant access the private attributes of the Pokemon class, the only way for them to modify them is to utilize the public methods.