

Design Paradigm

We used the **object-oriented design paradigm**. It works well for this project as the structure of a graduation plan naturally forms a hierarchy of semesters containing courses. Data can be grouped together effectively with an object-oriented design. For example, a Course object can contain the course code, credit hours, and semesters offered. A Semester object can contain the season (fall/summer/spring), the year, and a list of Course objects it contains. It also allows functionality to be bundled together with data, such as the ability of a Semester object to determine the total number of credit hours it contains or the ability of a Plan object to validate course prerequisites.

Furthermore, object-oriented design is beneficial for dividing up the core logic of our project into manageable pieces. For example, we can separate out of the logic for rendering the arrows, which involves several internal methods, into its own singleton class with a simple interface. Outside classes (i.e. Executive) then only need to pass in a list of arrows that need to be drawn, and do not need to worry about all of the complexity involved in finding an available path between the two courses on the diagram.

Object-oriented design is also helpful for dividing up work between team members. For example, one person can work on the rendering logic while another can work on the storage of data in Plan/Semester/Course objects. We are free to design the internal structure and logic of the objects we work on in a way that makes sense to us and we only need to agree on a few things (the external interfaces between our classes). Lastly, we know object-oriented design works well for our team as we were already familiar with this design paradigm from our previous experience in Projects 1 and 2.