**Waterfall**

* Requirements Analysis & Definition (week 1)
  + Goal is to recreate LeopardWeb with UIs, databases
  + Test existing system: LeopardWeb
* System & Software Design (week 3)
  + The design of the system is very clearly outlined in the description of the assignments
    - Database of users: the system should work for 100 students, 10 instructors, and 1 admin, however, we will test with fewer.
    - Database of courses: this will contain information such as the CRN, course name, times, and instructor.
    - Three types of users:
      * student – can register, can see available courses and their own schedule.
      * instructor – can see available courses and their own course roster.
      * admin – can see everything, can edit courses/users/schedules.
    - The system should include multiple semesters, print-out of schedule, scheduling preferences.
    - The system as a whole and all components must be tested thoroughly.
    - The base class of the system is user with:
      * Attributes: first name, last name, ID.
      * Methods: set function for each attribute, and a function to print all info for the object.
    - There will be three derived classes:
      * All derived classes must contain any additional attributes and appropriate set/get functions.
      * student – the student class will have functions that allow them to search courses, add/drop courses, print their schedule.
      * instructor – the instructor class will have functions that allow them to print their schedule, print their class list, and search for courses.
      * admin – the admin class will have functions that allow them to add courses to the system, remove courses from the system, add/remove users, add/remove student from a course, search and print rosters and courses.
* Implementation & Unit Testing (week 5)
  + Test components (classes, databases) separately
* Integration & System Testing (week 7)
  + Integrate all components
* Operation & Maintenance (week 9)
  + Have other users (students, instructor) test system
  + Update system based on user requests