Documentation:

=====================================================================

Class :: int foo (param a, ...)

This function is a template function

=====================================================================

Student :: public Student(String name, int id)

* Constructor

Student :: public Student(String name, int id, Faculty major, Faculty minor)

* Constructor

Student :: public Boolean addCourseToCompleted(Course course)

* Adds a course to this student’s completed courses list

Student :: public List<Course> getCompletedCourses()

* Returns the completed courses list

Student :: public void addCourseToShortlist(Course course)

* Adds a course to this student’s shortlist

Student :: public void removeCourseFromShortlist(Course course)

* Removes a course from the shortlist

Student :: public List<Course> getShortlist()

* Returns this student’s shortlist

=====================================================================

Faculty :: public Faculty(String name, int id)

* Constructor

Faculty :: public List<Course> getCoursesOffered()

* Returns the list of courses offered in this faculty

Faculty :: public Boolean addToCourseSet(Course course)

* Adds a course to this faculty’s offered courses. If it already exists it replaces the course with the new value
* Returns true if a course was replaced, false otherwise

Faculty :: public void removeFromCourseSet(Course course)

* Removes the course from the faculty’s offered courses set, if it exists

Faculty :: public List<Student> getStudents()

* Returns a list of students in this faculty

Faculty :: public void addStudentToFaculty(Student student)

* Adds a student to this Faculty’s students list. If it already exists, replaces the student with the new value
* Returns true if a course was replaced, false otherwise

Faculty :: public void removeStudentFromFaculty(Student student)

* Removes the student from this faculty’s students list if it exists

=====================================================================

Course :: public Course(String name, int id, String desc)

* Constructor

Course :: public Course(String name, int id, int sectionid, String desc)

* Constructor

Course :: public Boolean addPreReq(Course course)

* Adds the course to the preReqs list if it is not present. Replaces it with the new value
* Creates the list if it is currently null
* Returns true if a course was replaced, false otherwise

Course :: public void removePreReq(Course course)

* Removes the course from the preReqs list, if it exists

Course :: public List<Course> getPreReqs()

* Returns the preReqs list

=====================================================================

Advisor :: public List<Course> getRemainingPrerequisites(Student student, Course course)

* Returns a list of courses that the student has not yet taken.

Advisor :: public List<Course> getRecommendedCourses(Student student)

* Returns a list of courses that are recommended

Advisor :: private List<Course> getCoursesInYear(List<Course> courses, int year)

* Returns a list of courses that are offered at the given year level (eg. 1st, 2nd, etc year courses)
* Assumes 1st year courses have id’s between 1000 and 1999

=====================================================================