Person

- ssn: Double
- firstName: String
- lastname: String
- email: String

+ Person()

+ Person(ssn: Double, firstName: String, lastName:String, email: String)

+ toString(): String

Extends

Transcprit

- semester: Semester
- gpa: ArrayList<Double>
- totalGpa: Double
- creditsPassed: ArrayList<Integer>
- creditsFailed: ArrayList<Integer>
- coursePassed: ArrayList<Course>
- courseFailed: ArrayList<Course>
- coursesNotTaken: ArrayList<Course>
- activeCourses: ArrayList<Course>

+ Transcprit()

+ Transcript(semester: int, gpa: ArrayList<Double>, totalGpa: Double, creditsPassed: ArrayList<Integer>, creditsFailed: ArrayList<Integer>, coursesPassed: ArrayList<Course>, coursesFailed: ArrayList<Course>, coursesNotTaken: ArrayList<Course>, activeCourses: ArrayList<Course>)

+ gpaCalculator(semester: int): void

+ semesterGPA(coursePassed: ArrayList<Course>, courseFailed: ArrayList<Course>): double

+ calculateTakenCredit(): int

+ calculate

Student

- studentID: StudentID
- yearEnrolled: int
- schedule: Schedule
- lastTranscript: Transcript
- nextTranscript: Transcript
- collisionCourses: ArrayList<Course>
- logs: ArrayList<String>

+ Student()

+ Student(ssn: Double, firstName: String, lastName: String, email: String, studentID: StudentID, yearEnrolled: int, schedule: Schedule, lastTranscript: Transcript, nextTranscript: Transcript, collisionCourses:ArrayList<Course>)

+ showTranscript(): void

+ toString(): String

StudentID

- order: int
- id: String
- year: int

+ StudentID(year: int, order: int)

+ generateRandomId(year: int, order: int): String

Course

- courseCode: String
- courseName: String
- givenSemester: Semester
- courseSchedule: Schedule
- students: ArrayList<Student>
- courseCredit: int
- prerequisities: ArrayList<Course>
- section: int
- quota: int
- type: String

+ Course()

+ Course(courseCode: String, courseName: String, fivenSemester: Semester, courseSchedule: Schedule, students: ArrayList<Student>, courseCredit: int, prerequisities: ArrayList<String>, section: int, quota: int, type: String)

+ toString(): String

1

Faculty Member

- presentedCourses: ArrayList<Course>
- schedule: Schedule

+ FacultyMember()

+FacultyMember(ssn: Double, firstName: String, lastName: Stri email: String, presentedCourses: ArrayList<Course>, schedule: Schedule)

+ toString(): String

Extends

Advisor

- students: ArrayList<Student>

+ Advisor()

+ Advisor(ssn: Double, firstName: String, lastName: String, email: String, presentedCourses: ArrayList<Course>, schedule: Schedule, students: ArrayList<Student>)

+ checkCreditsForTechnicalElective(student : Student, course : Course) : boolean

+ checkHourCollision(student : Student) : boolean

+ detectConflict(student : Student, course : Course) : void

1...

Schedule

- courseHour: Time

+ Schedule()

+ Schedule(courseHour: int[][])

+ toString(): String

1..*

Extends

Elective Course

- electiveCourseCode: String
- electiveCourseName: String

+ ElectiveCourse()

+ ElectiveCourse(courseCode: String, courseName: String, fivenSemester: Semester, courseSchedule: Schedule, students: ArrayList<String>, courseCredit: int, prerequisities: ArrayList<String>, section: int, quota: int, type: String, electiveCourseCode: String, electiveCourseName: String)

Semester

- num: int

+ Semester()

+ Semester(num: int, name: String)

+ toString(): String

1