Person

- ssn : String
- name : String
- lastname : String
- email : String
- birthDate : int
- age: int

+ Person()
+ Person(ssn : String, name : String, lastname : String, email : String , birthDate : int , age : int)
+ getSsn(): String
+ setSsn(ssn : String): void
+ getName(): String
+ setName(name : String): void
+ getLastname(): String
+ setLastname(lastname : String): void
+ getEmail(): String
+ setEmail(email : String): void
+ getBirthDate(): int
+ setBirthDate(birthDate : int): void
+ getAge(): int
+ setAge(age : int): void
+ toString(): String

Professor

- degree : String
- presentedCourses : ArrayList<Course>
- weeklySchedule : Schedule

+ Instructur()
+ Instructur(ssn : String, name : String, lastname : String, email : String , birthDate : int , age : int,degree : String, presentedCourses : ArrayList<Course>, weeklySchedule : Schedule)
+ getDegree() : String
+ setDegree(degree : String) : void
+ getPresentedCourses() : ArrayList<Course>
+ setPresentedCourses(presentedCourses : ArrayList<Course>) : void
+ getWeeklySchedule() : Schedule
+ setWeeklySchedule(weeklySchedule : Schedule) : void
+ toString() : String

Student

- studentId : StudentID
- yearEnrolled : int
- weeklySchedule : Schedule
- previousTranscript : Transcript
- currentTranscript : Transcript
- collisionCourses : ArrayList<Course>
- logs : ArrayList<String>

+ Student()
+ Student(ssn : String, name : String, lastname : String, email : String, birthDate : int , age : int, studentId : StudentID, yearEnrolled : int, weeklySchedule : Schedule, currentTranscript : Transcript, previousTranscript : Transcript)
+ getLogs() : ArrayList<String>
+ setLogs(logs : ArrayList<String>) : void
+ getStudentId() : StudentID
+ setStudentId(studentId : studentID) : void
+ getYearEnrolled() : int
+ setYearEnrolled(yearEnrolled : int) : void

StudentID

+ order : int
- id : String
- departmentCode : String = "1501"
- year : int

+ StudentID(year : int)
+ getOrder() : int
+ setOrder(order : int) : void
+ getId() : String
+ setId(id : String) : void
+ getDepartmentCode() : String
+ setDepartmentCode(departmentCode : String) : void
+ getYear() : int
+ setYear(year : int) : void
+ generateRandomId(year : int) : String

0..*   1   0..*   0..*   0..*   1

Course

Transcript

Course (partial, top-left)

- CourseCode : String
- CourseTitle : String
- givenSemester : Semester
- courseSchedule : Schedule
- students : ArrayList<Student>
- courseCredit : int
- prerequisites : ArrayList<String>
- section : String
- quota : int
- type : String
- numOfStudents : int
- hourCollisionCounter : int
- prerequisiteFailureCounter : int

+ Course()
+ Course(courseCode : String, courseTitle : String, givenSemester :
  courseSchedule : Schedule, students : ArrayList<Student>, course
  prerequisites : ArrayList<String>, section : String, quota : int,
  numOfStudents : int, type : String)
+ getPrerequisiteFailureCounter() : int
+ setPrerequisiteFailureCounter(prerequisiteFailureCounter : int) : vo
+ incrementPrerequisiteFailiureCounter() : void
+ getCourseCode() : String
+ setCourseCode(courseCode : String) : void
+ getType() : String
+ setType(type : String) : void
+ getCourseTitle() : String
+ setCourseTitle(courseTitle : String) : void
+ getGivenSemester() : Semester
+ setGivenSemester(givenSemester : Semester) : void
+ getCourseSchedule() : Schedule
+ setCourseSchedule(courseSchedule : Schedule) : void
+ getStudents() : ArrayList<Student>
+ setStudents(students : ArrayList<Student>) :void
+ getCourseCredit() : int
+ setCourseCredit(courseCredit : int) : void
+ getPrerequisites() : ArrayList<String>
+ setPrerequisites(prerequisites : ArrayList<String>) : void
+ getSection() : String
+ setSection(section : String) : void
+ getQuota() : int
+ setQuota(quota : int) : void
+ getNumOfStudents() : int
+ setNumOfStudents(numOfStudents : int) : void
+ getHourCollisionCounter() : int
+ incrementHourCollisionCounter() : void
+ toString() : String

(top-center, partial)

+ getWeeklySchedule() : Schedule
+ setWeeklySchedule(weeklySchedule : Schedule) : void
+ getPreviousTranscript() : Transcript
+ getCurrentTranscript() : Transcript
+ getCollisionCourses() : ArrayList<Course>
+ printTranscript() : void

0..*   0..*   0..*

                                              1

Advisor

- students : ArrayList<Student>

+ Advisor()

+ Advisor(ssn : String, name : String, lastname : String,
email : String , birthDate : int , age : int,degree : String,
presentedCourses : ArrayList <Course>,
weeklySchedule : Schedule, students : ArrayList<Student>)

+ getStudents() : ArrayList<Student>

+ setStudents(students : ArrayList<Student>) : void

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

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

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

+ checkHourCollision(student : Student) : boolean

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

Schedule

- courseTime : int[][]

+ Schedule()

+ Schedule(courseTime : int [][])

+ getCourseTime() : int[][]

+ setCourseTime(courseTime : int[][]) : void

+ toString() : String

1   0..*

ElectiveCourse

- electiveCourseCode : String
- electiveCourseTitle: String

+ ElectiveCourse()
+ ElectiveCourse(courseCode : String, courseTitle : String,
givenSemester : Semester, courseSchedule : Schedule,
students : ArrayList<Student>, courseCredit : int,
prerequisites : ArrayList<String>, section : String, quota : int,
numOfStudents : int, type : String, electiveCourseTitle : String,

Registration

+ semesterCount : int = 0

+ Registration()

+ register(students : ArrayList<Student>,semester : int) : void

+ checkPrerequisites(checkCourse : Object[],
  globalPassedCourses : ArrayList<ArrayList<Object[]>>) : boolean

*

Transcript (right)

- semester : int
- gpa : ArrayList<Double>
- totalGpa : double
- passedCredits : ArrayList<Integer>
- failedCredits : ArrayList<Integer>
- coursesPassed : ArrayList<ArrayList<Object[]>>
- courseFailed : ArrayList<ArrayList<Object[]>>
- coursesNotTaken : ArrayList<Object[]>
- activeCourses : ArrayList<Object[]>

+ Transcript()

+ Transcript(semester : int, gpa : ArrayList<Double>, totalGpa : double
passedCredits :ArrayList<Integer>, failedCredits : ArrayList<Integer>,
coursesPassed : ArrayList<ArrayList<Object[]>>,
coursesFailed : ArrayList<ArrayList<Object[]>>,
coursesNotTaken : ArrayList<Object[]>,
activeCourses : ArrayList<Object[]>)

+ getCoursesNotTaken() : ArrayList<Object[]>

+ setCoursesNotTaken(coursesNotTaken : ArrayList<Object[]>) : void

+ getTotalGpa() : double

+ setTotalGpa(totalGpa : double) : void

+ getSemester() : int

+ setSemester(semester : int) : void

+ getGpa() : ArrayList<Double>

+ setGpa(gpa : ArrayList<Double>) : void

+ getPassedCredits() : ArrayList<Integer>

+ getFailedCredits() : ArrayList<Integer>

+ getCoursesPassed() : ArrayList<ArrayList<Object[]>>

+ setCoursesPassed(coursesPassed : ArrayList<ArrayList<Object[]>>

+ getCoursesFailed() : ArrayList<ArrayList<Object[]>>

+ setCoursesFailed(coursesFailed : ArrayList<ArrayList<Object[]>>) :

+ getActiveCourses() : ArrayList<Object[]>

+ setActiveCourses(activeCourses : ArrayList<Object[]>) : void

+ calculateGPAs(term : int) : void

+ calculateTermGPA(termPassed : ArrayList<Object[]>,
termFailed : ArrayList<Object[]>) : double

+ calculateTakenCredit() : int

+ calculateCompletedCredit() : int

electiveCourseCode : String)

+ getElectiveCourseCode() : String

+ setElectiveCourseCode(electiveCourseCode : String) : void

+ setElectiveCourseCode(electiveCourseCode : String) : void

+ getElectiveCourseTitle() : String

+ setElectiveCourseTitle(electiveCourseTitle : String) : void

---

+ getAvailableCourses(semester : int, gpa : double,
globalPassedCourses : ArrayList<ArrayList<Object[]>>,
globalFailedCourses : ArrayList<ArrayList<Object[]>>,
notTakenCourses : ArrayList<Object[]>) : void

+ get_gradeEquivalent(grade : String) : double

0..n

0..*

1

*

Semester

- num : int

- name : String

+ Semester()

+ Semester(num : int)

+ getNum() : int

+ setNum(num : int) : void

+ getName() : String

+ setName(name : String) : void

+ toString() : String

OutputGenerator

+ writeStudents(students : ArrayList<Student>) : void

+ evaluateCourseStatistics() : void

+ convertToJson() : String

+ removeDuplicates(list : ArrayList<String>) : ArrayList<String>