UML Class Diagram

Main
- +main(args : String[]) : void

Advisor
- -teErrorArrayList : ArrayList<String>
- -projectErrorArrayList : ArrayList<String>
- -quotaErrorArrayList : ArrayList<String>
- -collisionErrorArrayList : ArrayList<String>
- -prerequisiteErrorArrayList : ArrayList<String>
- -studentArrayList : Student
- +Advisor(lecturerName : String, department : String, rank : String)
- +addApprovalCourses() : void

Lecturer
- -lecturerName : String
- -department : String
- -rank : String
- +Lecturer(lecturerName : String, department : String, rank : Strin...

Process
- -createNewStudent : boolean
- -semester : String
- -process : Process
- -advisorArrayList : Advisor = new ArrayList<>()
- -studentArrayList : Student = new ArrayList<>()
- -courseArrayList : Course = new ArrayList<>()
- -Process()
- -getCreateNewStudentFromJSON() : boolean
- -getSemesterFromJSON() : String
- -getAdvisorsFromJSON() : ArrayList<Advisor>
- -createCourses() : ArrayList<Course>
- +createElectiveCourses(myCourseArrayList : ArrayList<Course>, courseJsonObject : JsonObject, typeOfElective : String) : ArrayList<Course>
- +getSemesterCourses(semester : int) : ArrayList<Course>
- +getElectiveCourses(typeOfElective : String) : ArrayList<Course>
- -checkForQuota(course : Course) : boolean
- +initializeStudents() : void
- -createTranscript(year : int) : Transcript
- -createCourseOffered(student : Student) : ArrayList<Course>
- -semesterToInt(year : int) : int
- -getNumericGradeFromLetterGrade(letterGrade : String) : double
- +createJSONForAllStudents() : void
- +addApprovalCoursesForAllStudents() : void
- +printErrorLog() : void
- -generateRandomLetterGrade() : String
- -generateRandomName() : String
- -generateStudentNumber(year : int, count : int) : int

Course
- -courseName : String
- -courseCredit : int
- -semester : int
- -quota : int
- -prequisiteName : String
- -electiveType : String
- -courseHour : JsonArray
- -studentList : Student
- -lecturer : Lecturer
- -prequisite : Course
- +Course(courseName : String, credit : int, semester : int, lecturer : Lecturer, quota : int, prequisite : Course, courseHour : JsonArray)
- +Course()
- +enrollStudent(student : Student) : void
- +getEnrolledStudents() : ArrayList<Student>

Student
- -studentNumber : int
- -studentName : String
- -year : int
- -error : ArrayList<String>
- -advisor : Advisor
- -transcriptBefore : Transcript
- -courseOffered : Course
- -transcriptAfter : Transcript
- +Student()
- +addError(error : String) : void
- +createStudenJSON() : void

TranscriptRow
- -letterGrade : String
- -course : Course
- +TranscriptRow(course : Course, letterGrade : String)
- +TranscriptRow(course : Course)

Transcript
- -givenCredit : double
- -completedCredit : double
- -point : double
- -gano : double
- -student : Student
- -transcriptRow : TranscriptRow
- +Transcript(student : Student, transcriptRow : ArrayList<TranscriptRow>, givenCredit : double, completedCredit : double, point : double, gano : dou...
- -Transcript()

Relationships:
- -advisorArrayList *
- -lecturer
- -advisor
- -studentArrayList *
- -process
- -prequisite
- -courseArrayList
- -studentList *
- -courseOffered
- -studentArrayList *
- -course
- -transcriptRow *
- -transcriptAfter
- -transcriptBefore
- -student