## **JSONReader** JSONReader(): studentRegistrationMap: Map<Student, Registration> mapper: ObjectMapper - advisorIDCoursesMap: Map<Integer, List<String>> coursePrerequisiteCourseCodesMap: Map<Course, List<String>> lecturerIDCoursesMap: Map<Integer, List<String>> studentIDAdvisorIDMap: Map<Integer, Integer> department: Department - courseSectionCodesMap: Map<Course, List<String>> studentIDDraftMap: Map<Integer, List<String>> - jsonNode: JsonNode readCourses(): void start(Department): void + readTranscript(Student): void readAdvisors(): void + readStudents(): void readRequests(): void readCourseSections(): void readJson(): void - syncObjects(): void readLecturers(): void

## Registration Registration(Student, List<Course>):

**JSONWriter** 

objectMapper: ObjectMapper

department: Department

jsonNode: JsonNode

writeStudents(): void

writeJson(): void

- Grade(String):

- gradeOver4: double

gradeOver100: int

letterGrade: String

setGradeOver4(): void

+ getLetterGrade(): String

- setGradeOver100(): void

+ getGradeOver4(): double

writeTranscripts(): void

start(Department): void

Grade

- writeRequests(): void

JSONWriter():

student: Student courses: List<Course>

getCourses(): List<Course> - approveRequest(): void

rejectRequest(): void - addRequest(Advisor): void

getStudent(): Student

## + main(): + main(String[]): void

main



## + Course(String, String, int, byte): courseCredit: int - preRequisiteCourses: List<Course> courseCode: String · lecturer: Lecturer - courseSections: List<CourseSection> courseName: String - gradeLevel: byte + setLecturer(Lecturer): void + getCourseCode(): String + getCourseName(): String + getCourseCredit(): int + getPreRequisiteCourses(): List<Course> + getGradeLevel(): byte + getCourseSections(): List CourseSection CourseSection(Course, String, String, int): - day: String courseSectionCode: String - hour: int - students: List

- getCourseSectionCode(): String

Course

**Transcript** 

· Transcript(Student)

student: Student

gradeLevel: byte

takenCredits: int

completedCredits: int

studentCourses: List<Course>

+ getStudentCourses(): List<Course>

calculateCompletedCredits(): int

+ courseStatusCheck(): void

+ getCompletedCredits(): int

calculateValues(): void

+ getTakenCredits(): int

· calculateCgpa(): double

calculateTakenCredits(): int

setTakenCredits(int): void

- setStudentCourses(List<Course>): void

getCourseGradeMap(): Map<Course, Grade>

- setCourseGradeMap(Map<Course, Grade>): void

showTranscript(): void

setStudent(Student): void

+ getCgpa(): double

courseGradeMap: Map<Course, Grade>

cgpa: double

Department

advisors: List<Advisor>

getTranscripts(): List<Transcript>

getMaxCourseNumber(): byte