

SessionID.java

Added

- <<constructor>>SessionID()

InstructorID.java

Added

- <<constructor>>InstructorID()
- digitFixer(Integer): String

StudentID.java

Added

- <<constructor>>StudentID()
- + setID(String): void
- digitFixer(Integer): String

Transcript.java

Added

- totalCreditsTakenCalculator(ArrayList<Semester>): Integer
- totalCreditsCompletedCalculator(ArrayList<Semester>): Integer
- pointsCalculator(ArrayList<Semester>): Double

LectureSession.java

Added

- listOfStudents: ArrayList<Student>

Lecture.java

Added

- term: Term
- termYear: TermYear

Removed

- addLectureSession(LectureSession): void

Semester.java

Added

- term: Term
- termYear: TermYear
- creditsTakenCalculator(HashMap<Lecture, LetterGrade>): Integer
- creditsCompletedCalculator(HashMap<Lecture, LetterGrade>): Integer

- pointsCalculator(ArrayList<Semester>): Double
- addLecture(Lecture): void

Removed

- removeLecture(Lecture, LectureGrade): void

LectureRegistrationApplication.java

Removed

- approveApplication(): void

Person.java

Removed

id: UniqueID

- + Person(String, String)

Instructor.java

Added

- schedule: Schedule
- scanner: Scanner
- + showLectures(): void
- + showStudents(): void

Removed

- listOfLectures: ArrayList
- + addLecture(Lecture): void
- + removeLecture(Lecture): void

Advisor.java

Added

- scanner: Scanner
- + showApplications(): void
- + applicationOperations(Integer): void
- + rejectApplications(LectureRegistrationApplication): void

Student.java

Added

- scanner: Scanner
- schedule: Schedule
- + sendForApproval(ArrayList<LectureSession>): void

- + showTranscript(): void
- + showSchedule(): void
- + debtMenu(): void
- + registrationStatusMenu(): void
- + makeRegistrationMenu(): void
- parseSelectionCommand(String, ArrayList<LectureSession>): void
- showChosenLectureSessions(ArrayList<LectureSession>): void
- canTakeLecture(Lecture, Transcript): boolean
- hasPreqLectureTaken(Lecture, ArrayList<Lecture>): boolean
- takenPoint(Lecture, HashMap<Lecture, LetterGrade>): boolean
- + availableLessons(Student): ArrayList<Lecture>

Removed

- listOfLectureSessions: ArrayList

FilterType.java

Added

-LectureSemester

LetterGrade.java

Removed

+ getLetterGradeValue(): Double

JsonOperator.java

Added

- lectureList: ArrayList<LectureJSON>
- studentList: ArrayList<StudentJSON>
- transcriptList: ArrayList<TranscriptJSON>
- advisorList: ArrayList<AdvisorJSON>
- lectureObjectList: ArrayList<Lecture>
- lstudentObjectList: ArrayList<Student>
- advisorObjectList: ArrayList<Advisor>
- objectGenerator: ObjectGenerator
- jsonGenerator: JsonGenerator
- metaData: MetaData
- namePool: NamePool
- + readNamePool(): void

- + readMetaData(): void
- + readLectureJSON(String): void
- + readStudentJSON(String): void
- + readTranscriptJSON(String): void
- + readAdvisorJSON(String): void
- + generateObjects(): void

Removed

- sinleInstance: JsonOperator
- + readJsonFile(File, Class: T
- + writeJsonFile(String,): void

DataManager.java

Added

- jsonOperator: JsonOperator
- + saveObjectsAsJson(): void
- + searchLecture(String, FilterType): ArrayList<Lecture>
- + searchLecture(Term, TermYear): ArrayList<Lecture>
- + searchLecture(Integer): ArrayList<Lecture>
- + searchLectureUntilTerm(Integer): ArrayList<Lecture>
- + searchLectureUntilTerm(Term, TermYear): ArrayList<Lecture>
- + searchStudent(String, FilterType): ArrayList<Student>
- + searchInstructor(String, FilterType): ArrayList<Instructor>
- + searchAdvisor(String, FilterType): ArrayList<Advisor>
- + writeExamples(): void
- + addStudents(ArrayList<Student>): void
- fixCache(): void
- listFiles(String): ArrayList<String>

Removed

- + findListOfFilesInDirectory(): File[]
- + loadLectures(): void
- + loadAdvisors(): void
- + loadInstructors(): void
- + loadStudents(): void
- + findPerson(FilterType, String): ArrayList

+ contentsOfFiles(File[]): ArrayList

StudentRegistrationSystem.java

Removed

+ transcriptMenu(Student): void

+ scheduleMenu(Student): void

+ makeRegistrationMenu(Student): void

+ registrationStatusMenu(Student): void

+ debtMenu(Student): void

Added Schedule.java

-added - person: Person

-added - listOfLectureSessions: ArrayList<LectureSession>

-added - term: Term

-added - termYear: TermYear

-added + <<constructer>>Shedule

-added + showSchedule(): void

Added JsonReader.java

-added - file: File

-added + JsonReader(String)

-added + JsonReader(File)

-added + readJsonFiles(Class<T>): <T>

Added JsonWriter.java

-added - file: File

-added + JsonWriter(String)

-added + JsonWriter(File)

-added + writeJsonFiles(T): <T>

Added ObjectGenerator.java

-added - lectureList: ArrayList<LectureJSON>

-added - lectureList: ArrayList<LectureJSON>

-added - transcriptList: ArrayList<TranscriptJSON>

-added - advisorList: ArrayList<AdvisorJSON>

-added - lectureObjectList: ArrayList<Lecture>

-added - studentObjectList: ArrayList<Student>

-added - transcriptObjectList: ArrayList<Transcript>

-added - advisorObjectList: ArrayList<Advisor>

```

-added + <<constructor>>ObjectGenerator()
    -added +generateObjects: void
    -added + pairObjects(): void
    -added - pairForLecture(): void
    -added - pairForStudent(): void
-added - pairForTranscript(Student): void
    -added - pairForAdvisor(): void
    -added - generateLectures(): void
    -added - generateStudents(): void
    -added - generateAdvisors(): void
-added - findLecture(String): Optional<Lecture>
-added - findStudent(String): Optional<Student>
-added - findAdvisor(String): Optional<Advisor>
-added -generateLectureSessions(ArrayList<LectureSessionJSON> : List<LectureSessionJSON>
    -added - stringToCalendar(String): Optional<Calendar>
    -added - stringToLectureType(String): LectureType
    -added - stringToSessionType(String): SessionType
    -added - stringToInstructorType(String): InstructorType
    -added - stringToLetterGrade(String): LetterGrade
    -added - stringToTerm(String): Term
    -added - stringToTermYear(String): TermYear
-added - intToLectureHours(Integer [] []): LectureHour [] []
Added TranscriptGenerator.java
-added - termAndYear: HashMap<String, Integer>
    -added <<constructor>>TranscriptGenerator()
    -added + generate(Student, Schedule): Transcript
        -added - build(): void
    -added - randomLetterGrade(): LetterGrade
-added - semesterGenerator(Student, Schedule, Integer): Semester
-added - setAllSemester(Student, Schedule): ArrayList<Semester>
Added StudentIDGenerator.java
    -added - departmentCode: int
        -added - yearCode: int
    -added - orderOfPlacement: int

```

-added <<constructor>>StudentIDGenerator(Student, Student)

Added **StudentGenerator.java**

-added- namePool: NamePool

-added - transcriptGenerator: TranscriptGenerator

-added <<constructor>>StudentGenerator()

-added + generate(Integer, Integer): Student

-added - studentIdGenerator(Integer, Integer): StudentID

-added - studentDebtGenerator(): Debt

-added - LRAGenerator(Student): LectureRegistrationApplication