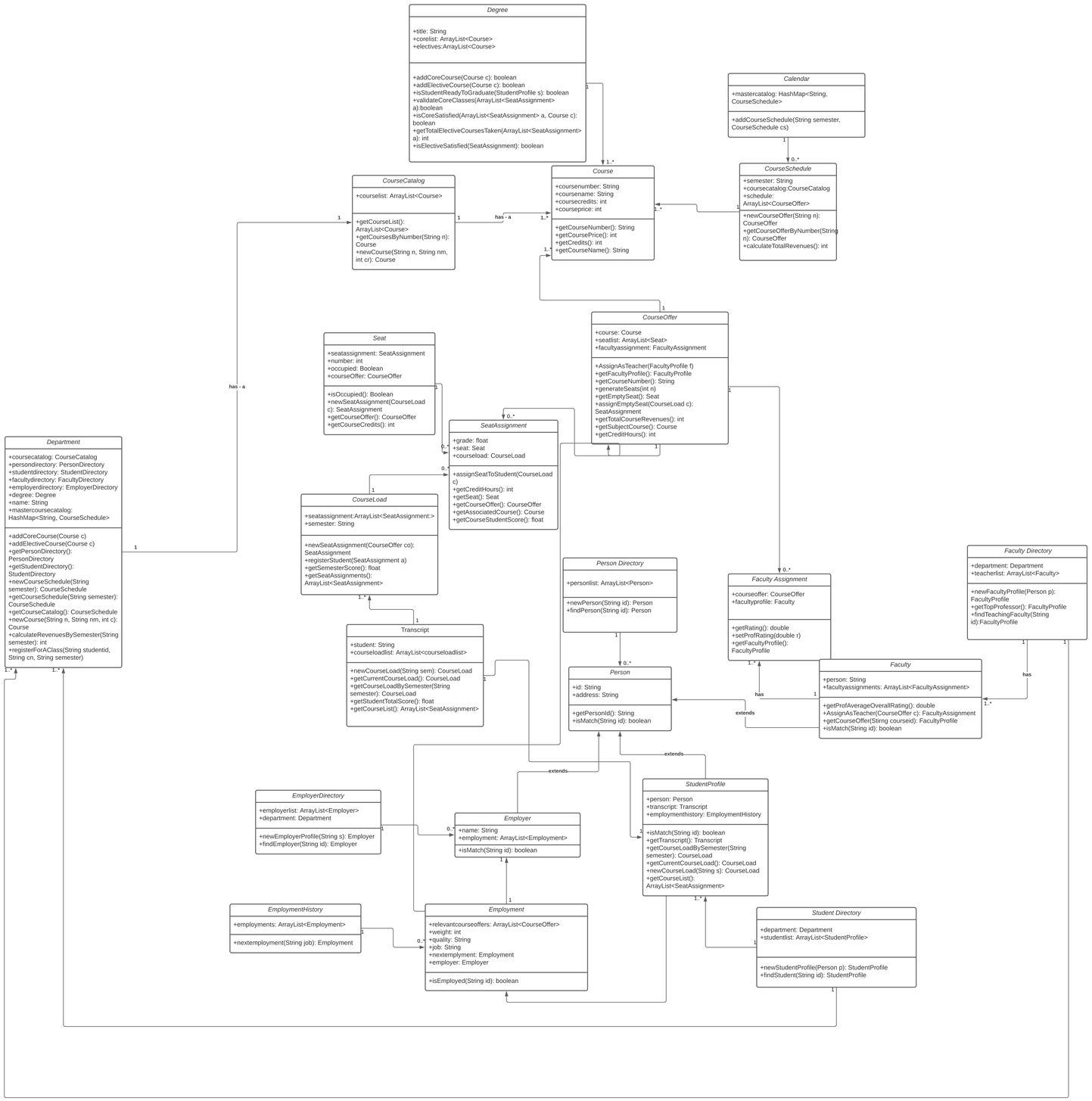
Degree
+title: String
+corelist: ArrayList<Course>
+electives:ArrayList<Course>
+addCoreCourse(Course c): boolean
+addElectiveCourse(Course c): boolean
+isStudentReadyToGraduate(StudentProfile s): boolean
+validateCoreClasses(ArrayList<SeatAssignment> a):boolean
+isCoreSatisfied(ArrayList<SeatAssignment> a, Course c): boolean
+getTotalElectiveCoursesTaken(ArrayList<SeatAssignment> a): int
+isElectiveSatisfied(SeatAssignment): boolean
Calendar
+mastercatalog: HashMap<String, CourseSchedule>
+addCourseSchedule(String semester, CourseSchedule cs)
CourseCatalog
+courselist: ArrayList<Course>
+getCourseList(): ArrayList<Course>
+getCoursesByNumber(String n): Course
+newCourse(String n, String nm, int cr): Course
Course
+coursenumber: String
+coursename: String
+coursecredits: int
+courseprice: int
+getCourseNumber(): String
+getCoursePrice(): int
+getCredits(): int
+getCourseName(): String
CourseSchedule
+semester: String
+coursecatalog:CourseCatalog
+schedule: ArrayList<CourseOffer>
+newCourseOffer(String n): CourseOffer
+getCourseOfferByNumber(String n): CourseOffer
+calculateTotalRevenues(): int
has - a
CourseOffer
+course: Course
+seatlist: ArrayList<Seat>
+facultyassignment: FacultyAssignment
+AssignAsTeacher(FacultyProfile f)
+getFacultyProfile(): FacultyProfile
+getCourseNumber(): String
+generateSeats(int n)
+getEmptySeat(): Seat
+assignEmptySeat(CourseLoad c): SeatAssignment
+getTotalCourseRevenues(): int
+getSubjectCourse(): Course
+getCreditHours(): int
Seat
+seatassignment: SeatAssignment
+number: int
+occupied: Boolean
+courseOffer: CourseOffer
+isOccupied(): Boolean
+newSeatAssignment(CourseLoad c): SeatAssignment
+getCourseOffer(): CourseOffer
+getCourseCredits(): int
SeatAssignment
+grade: float
+seat: Seat
+courseload: CourseLoad
+assignSeatToStudent(CourseLoad c)
+getCreditHours(): int
+getSeat(): Seat
+getCourseOffer(): CourseOffer
+getAssociatedCourse(): Course
+getCourseStudentScore(): float
Department
+coursecatalog: CourseCatalog
+persondirectory: PersonDirectory
+studentdirectory: StudentDirectory
+facultydirectory: FacultyDirectory
+employerdirectory: EmployerDirectory
+degree: Degree
+name: String
+mastercoursecatalog: HashMap<String, CourseSchedule>
+addCoreCourse(Course c)
+addElectiveCourse(Course c)
+getPersonDirectory(): PersonDirectory
+getStudentDirectory(): StudentDirectory
+newCourseSchedule(String semester): CourseSchedule
+getCourseSchedule(String semester): CourseSchedule
+getCourseCatalog(): CourseSchedule
+newCourse(String n, String nm, int c): Course
+calculateRevenuesBySemester(String semester): int
+registerForAClass(String studentid, String cn, String semester)
CourseLoad
+seatassignment:ArrayList<SeatAssignment:>
+semester: String
+newSeatAssignment(CourseOffer co): SeatAssignment
+registerStudent(SeatAssignment a)
+getSemesterScore(): float
+getSeatAssignments(): ArrayList<SeatAssignment>
Transcript
+student: String
+courseloadlist: ArrayList<courseloadlist>
+newCourseLoad(String sem): CourseLoad
+getCurrentCourseLoad(): CourseLoad
+getCourseLoadBySemester(String semester): CourseLoad
+getStudentTotalScore(): float
+getCourseList(): ArrayList<SeatAssignment>
Person Directory
+personlist: ArrayList<Person>
+newPerson(String id): Person
+findPerson(String id): Person
Faculty Assignment
+courseoffer: CourseOffer
+facultyprofile: Faculty
+getRating(): double
+setProfRating(double r)
+getFacultyProfile(): FacultyProfile
Faculty Directory
+department: Department
+teacherlist: ArrayList<Faculty>
+newFacultyProfile(Person p): FacultyProfile
+getTopProfessor(): FacultyProfile
+findTeachingFaculty(String id):FacultyProfile
Person
+id: String
+address: String
+getPersonId(): String
+isMatch(String id): boolean
Faculty
+person: String
+facultyassignments: ArrayList<FacultyAssignment>
+getProfAverageOverallRating(): double
+AssignAsTeacher(CourseOffer c): FacultyAssignment
+getCourseOffer(String courseid): FacultyProfile
+isMatch(String id): boolean
has
extends
EmployerDirectory
+employerlist: ArrayList<Employer>
+department: Department
+newEmployerProfile(String s): Employer
+findEmployer(String id): Employer
Employer
+name: String
+employment: ArrayList<Employment>
+isMatch(String id): boolean
StudentProfile
+person: Person
+transcript: Transcript
+employmenthistory: EmploymentHistory
+isMatch(String id): boolean
+getTranscript(): Transcript
+getCourseLoadBySemester(String semester): CourseLoad
+getCurrentCourseLoad(): CourseLoad
+newCourseLoad(String s): CourseLoad
+getCourseList(): ArrayList<SeatAssignment>
EmploymentHistory
+employments: ArrayList<Employment>
+nextemployment(String job): Employment
Employment
+relevantcourseoffers: ArrayList<CourseOffer>
+weight: int
+quality: String
+job: String
+nextemplyment: Employment
+employer: Employer
+isEmployed(String id): boolean
Student Directory
+department: Department
+studentlist: ArrayList<StudentProfile>
+newStudentProfile(Person p): StudentProfile
+findStudent(String id): StudentProfile