Person

- name: String
- courses: ArrayList<Course>

+ Person(): Person
+ setName(newname: String): void
+ getName(): String
+ getCourses(): ArrayList<Course>
+ detectConflict(aCourse: Course): Boolean
+ printSchedule(): void
+ addCourse(aCourse: Course): abstract void
- getTimeSlots(): int[]

Student

- dept: Department
- unitsCompleted: int
- totalUnitsNeeded: int

+ Student(): Student()
+ setCompletedUnits(units: int): void
+ getUnitsCompleted(): int
+ requiredToGraduate(): int
+ setRequiredUnits(units: int): void
+ getTotalUnits(): int
+ setDepartment(dept: Department): void
+ getDepartment(): Department
+ addCourse(crs: Course): void
+ dropCourse(crs: Course): void

Employee

- payRate: double

+ Employee(): Employee
+ setPayRate(pay: double): void
+ getPayRate(): double
+ raise(percent: double): void
+ earns(): abstract double

Professor

- dept: Department

+ Professor(): Professor
+ setDepartment(dept: Department): void
+ getDepartment(): Department
+ earns(): double
+ addCourse(aCourse: Course): void

Staff

- dept: Department
- hoursWorked: double

+ Staff(): Staff
+ setMonthlyHours(hours: double): void
+ getMonthlyHours(): double
+ setDepartment(dept: Department): void
+ getDepartment(): Department
+ earns(): double
+ addCourse(aCourse: Course): void

Department

- name: String
- courses: ArrayList<Course>
- students: ArrayList<Student>
- profs: ArrayList<Professor>
- staffList: ArrayList<Staff>

+ Department(): Department
+ setDepartmentName(name: String): void
+ getDepartmentName(): String
+ addStaff(Staff stf): void
+ getStaffList(): ArrayList<Staff>
+ addProfessor(Professor prof): void
+ getProfessorList(): ArrayList<Professor>
+ addStudent(Student student): void
+ getStudentList(): ArrayList<Student>
+ addCourse(course: Course): void
+ getCourseList(): ArrayList<Course>
+ printStudentList(): void
+ printProfessorList(): void
+ printStaffList(): void
+ printCourseList(): void

University

+ departmentList: ArrayList<Department>
+ classroomList: ArrayList<Classroom>

+ University(): University
+ printStudentList(): void
+ printStaffList(): void
+ printProfessorList(): void
+ printCourseList(): void

Classroom

- courses: ArrayList<Course>
- room: String

+ Classroom(): Classroom
+ setRoomNumber(newRoom: String): void
+ getRoomNumber(): String
+ addCourse(aCourse: Course): void
+ printSchedule(): void

Course

- Week: String[]
- Slot: String[]
- name: String
- number: int
- schedule: ArrayList<Integer>
- roster: ArrayList<Person>
- dept: Department
- room: Classroom
- prof: Professor

+ Course(): Course
+ getMeetingTime(timeslot: int): String
+ printSchedule(): void
+ printRoster(): void
+ setRoomAssigned(newRoom: Classroom): void
+ getRoomAssigned(): Classroom
+ compareSchedules(other: Course): boolean
+ setName(name: String): void
+ getName(): String
+ setProfessor(prof: Professor): void
+ getProfessor(): Professor
+ setCourseNumber(num: int): void
+ getCourseNumber(): int
+ setSchedule(addNum: int): void
+ getSchedule(): ArrayList<Integer>
+ setDepartment(dept: Department): void
+ getDepartment(): Department
+ addStudent(student: Person): void
+ removeStudent(student: Student): void
+ removeStudent(student: Staff): void
+ getStudentRoster(): ArrayList<Person>
+ getConflictSlots(aCourse: Course): ArrayList<String>