Department

```
__laboratorySections: List
__sectionCodeCourseMap: Map
__studentIDStudentMap: Map
__lecturers: List
__courseCodeCourseMap:  Map
__assistants: List
__labSectionCourseMap: Map
__students: List
__advisors: List
__courses: List
__advisorIDAdvisorMap: Map
__departmentName: String
__transcripts: List
__maxCourseNumber: Integer
```
```
__init__(self, departmentName: str):
getDepartmentName(self):
getAdvisors(self):
getTranscripts(self):
getCourses(self):
getStudents(self):
getLaboratorySections(self):
getLecturers(self):
getAssistants(self):
getCourseCodeCourseMap(self):
getSectionCodeCourseMap(self):
getLabSectionCourseMap(self):
getStudentIDStudentMap(self):
getAdvisorIDAdvisorMap(self):
getMaxCourseNumber(self):
```
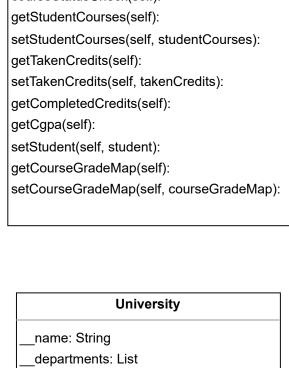
Grade

```
__gradeOver4: Integer
__letterGrade: String
__gradeOver100: Integer
```
```
__init__(self, letterGrade):
setGradeOver100(self):
setGradeOver4(self):
getLetterGrade(self):
getGradeOver4(self):
```

ConsoleColours

```
ANSI_RESET: String
ANSI_WHITE_BRIGHT: String
ANSI_BLACK: String
ANSI_RED_BACKGROUND: String
ANSI_GREEN_BACKGROUND: String
ANSI_BLUE_BACKGROUND: String
ANSI_YELLOW_BACKGROUND: String
ANSI_PURPLE_BACKGROUND: String
ANSI_WHITE_BRIGHT_BACKGROUND: String
```
```
resetColour():
paintBlueMenu():
paintRedMenu():
paintYellowMenu():
paintGreenMenu():
paintPurpleMenu():
paintWhiteMenu():
```

Course

```
__semester: Byte
__capacity: Integer
__courseCode: String
__laboratorySections: List
__courseType: String
__lecturer: Lecturer
__courseName: String
__numberOfStudents: Integer
__day: String
__hour: Integer
__courseCredit: Integer
__preRequisiteCourses: List
```
```
__init__(self, courseName, courseCode, courseType, courseCre
getCourseName(self):
getPreRequisiteCourses(self):
getCourseCode(self):
getCourseType(self):
getCourseCredit(self):
getHour(self):
getDay(self):
getLaboratorySections(self):
getNumberOfStudents(self):
setNumberOfStudents(self, numberOfStudents):
setLecturer(self, lecturer):
hasCapacity(self):
getSemester(self):
```

Notification

```
recipient: Person
message: String
```
```
__init__(self, recipient, message):
sendNotification(self):
getMessage(self):
```

Registration

```
__system:
__courses: List
__student: Student
```
```
__init__(self, student, courses):
approveRequest(self):
rejectRequest(self):
addRequest(self, advisor):
getStudent(self):
getCourses(self):
```

Person

```
__name: String
__password: String
__ID: Integer
__userName: String
__surname: String
__department: String
```
```
__init__(self, ID, name, surname, userName, password, department=None):
login(self, userName, password):
getUserName(self):
setUserName(self, userName):
getPassword(self):
setPassword(self, password):
getDepartment(self):
setDepartment(self, department):
getName(self):
getSurname(self):
getID(self):
```

IDisplayMenu

```
printMenu(self, menuType):
```

JSONReader

```
__assistantIDAssistantMap: Map
__advisorIDCoursesMap: Map
__system:
__coursePrerequisiteCourseCodesMap: Map
__studentIDAdvisorIDMap: Map
__studentRegistrationMap: Map
__department: Department
__university: University
__lecturerIDCoursesMap: Map
__sectionCodeSectionMap: Map
```
```
__init__(self):
readDepartments(self, university):
start(self, department):
readJson(self):
readCourses(self):
readAssistants(self):
readLabSections(self):
readLecturers(self):
readStudents(self):
readTranscript(self, student):
readRequests(self):
readAdvisors(self):
syncObjects(self):
```

Transcript

```
__semester: Integer
__cgpa: Integer
__completedCredits: Integer
__student: Student
__courseGradeMap: Map
__takenCredits: Integer
__studentCourses: List
```
```
__init__(self, student):
calculateValues(self):
calculateCgpa(self):
calculateTakenCredits(self):
calculateCompletedCredits(self):
showTranscript(self):
courseStatusCheck(self):
getStudentCourses(self):
setStudentCourses(self, studentCourses):
getTakenCredits(self):
setTakenCredits(self, takenCredits):
getCompletedCredits(self):
getCgpa(self):
setStudent(self, student):
getCourseGradeMap(self):
setCourseGradeMap(self, courseGradeMap):
```

JSONWriter

```
__department: Department
```
```
__init__(self):
start(self, university):
writeJson(self):
writeTranscripts(self):
writeRequests(self):
writeStudents(self):
```

University

```
__name: String
__departments: List
__userNamePersonMap: Map
```
```
__init__(self, name):
getDepartments(self):
getUserNamePersonMap(self):
```

Assistant

```
__init__(self, ID, name, surname):
login(self, userName, password):
```

Lecturer

```
__lessonsTaught: List
```
```
__init__(self, ID, name, surname, userName, password):
getLessonsTaught(self):
login(self, userName, password):
```

Advisor

```
__system:
__studentsAdvised: List
__requestList: List
__requestNumber: Integer
__notification: Notification
```
```
__init__(self, ID, name, surname, userName, password):
printRequests(self):
replyRequests(self):
printMenu(self, menuType):
login(self, userName, password):
getStudentsAdvised(self):
getRequestList(self):
setNotification(self, notification):
```

Student

```
__draft: List
__transcript: Transcript
__advisor: Advisor
__hasRequest: Boolean
__notification: Notification
__semester: Integer
__system:
__labSections: List
__availableCoursesToDrop: List
__schedule: List
__availableCoursesToAdd: List
```
```
__init__(self, studentID, name, surname, userName, password, se
sendRequest(self):
printMenu(self, menuType):
addCourseToDraft(self):
addMandatoryCourse(self):
chooseLabSection(self, course):
addTechnicalElective(self):
computeAvailableTEAndFTECourses(self, courseType):
addFacultyTechnicalElective(self):
addNonTechnicalElective(self):
computeAvailableNTECourses(self):
computeAvailableMandatoryCourses(self):
addCourseToDrop(self):
checkThePrerequisiteAndCourseThatWasTaken(self, course):
hasCourseOverlap(self, courseToAdd, isLab):
maxCoursesReached(self):
calculateNumberOfCourses(self):
removeCourseFromDraft(self):
showRequestStatus(self):
computeAvailableCoursesToDrop(self):
login(self, userName, password):
getDepartmentName(self):
setPassword(self, password):
setUserName(self, userName):
getAdvisor(self):
setAdvisor(self, advisor):
getSemester(self):
getDraft(self):
setDraft(self, draft):
isHasRequest(self):
setHasRequest(self, hasRequest):
setNotification(self, notification):
getTranscript(self):
setTranscript(self, transcript):
getSchedule(self):
getLabSections(self):
```

CourseRegistrationSystem

```
__choice: Integer
__university: University
__input: Scanner
```
```
__init__(self):
start(self):
mainMenu(self):
loginMenu(self):
studentMenu(self, student):
courseRegistrationMenu(self, student):
advisorMenu(self, advisor):
getInput(self):
exitProgram(self):
printMenu(self, menuType):
addToSchedule(self, course, student):
removeFromSchedule(self, course, student):
```

LaboratorySection

```
__capacity: Integer
__assistant: Assistant
__numberOfStudents: Integer
__day: String
__hour: Integer
__laboratorySectionCode: String
```
```
__init__(self, laboratorySectionCode, capacity, hour, day):
getLaboratorySectionCode(self):
setAssistant(self, assistant):
getDay(self):
getHour(self):
getCapacity(self):
getNumberOfStudents(self):
setNumberOfStudents(self, numberOfStudents):
```