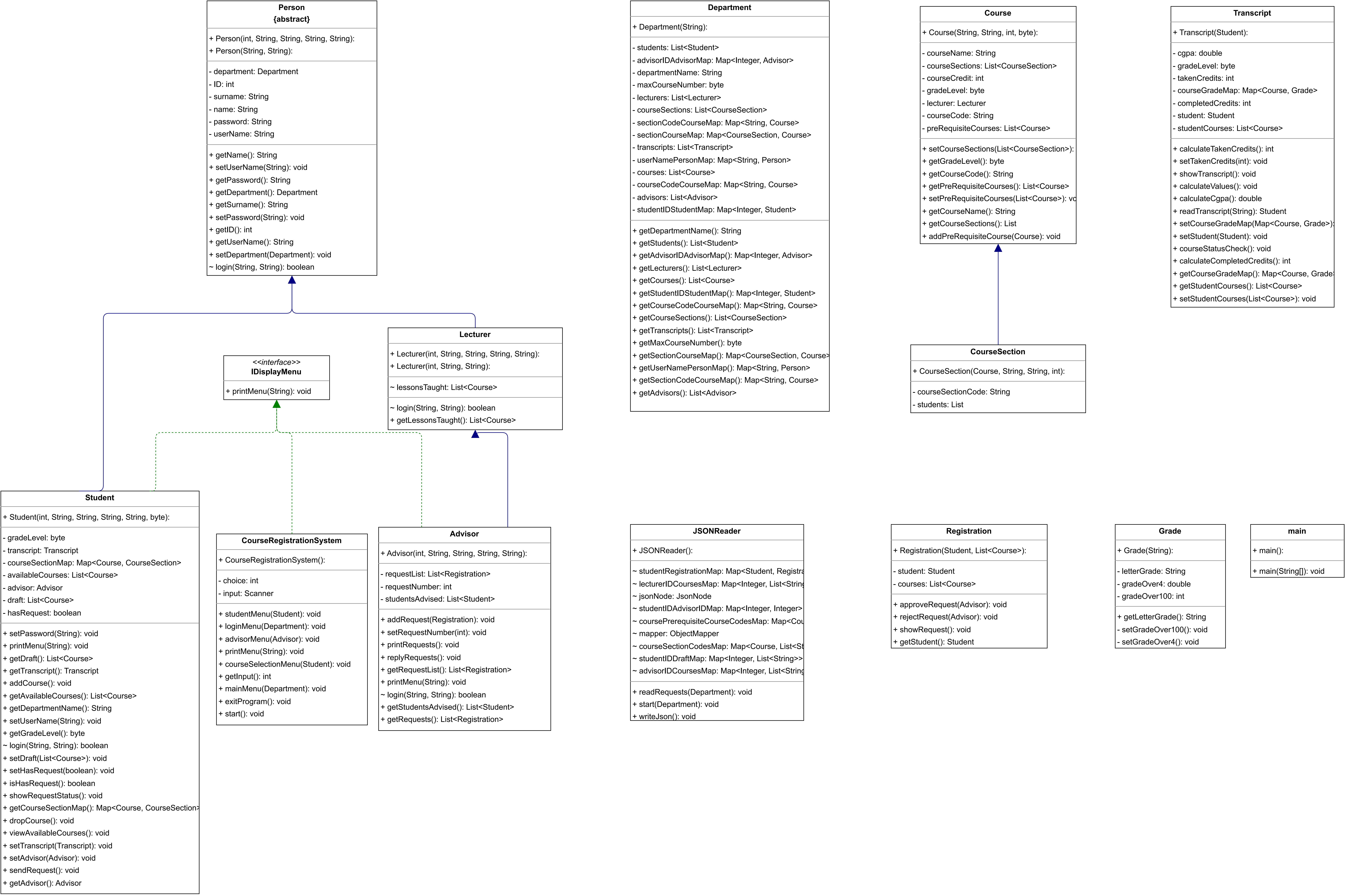

Person
{abstract}

```
+ Person(int, String, String, String, String):
+ Person(String, String):
- department: Department
- ID: int
- surname: String
- name: String
- password: String
- userName: String
+ getName(): String
+ setUserName(String): void
+ getPassword(): String
+ getDepartment(): Department
+ getSurname(): String
+ setPassword(String): void
+ getID(): int
+ getUserName(): String
+ setDepartment(Department): void
~ login(String, String): boolean
```

Department

```
+ Department(String):
- students: List<Student>
- advisorIDAdvisorMap: Map<Integer, Advisor>
- departmentName: String
- maxCourseNumber: byte
- lecturers: List<Lecturer>
- courseSections: List<CourseSection>
- sectionCodeCourseMap: Map<String, Course>
- sectionCourseMap: Map<CourseSection, Course>
- transcripts: List<Transcript>
- userNamePersonMap: Map<String, Person>
- courses: List<Course>
- courseCodeCourseMap: Map<String, Course>
- advisors: List<Advisor>
- studentIDStudentMap: Map<Integer, Student>
+ getDepartmentName(): String
+ getStudents(): List<Student>
+ getAdvisorIDAdvisorMap(): Map<Integer, Advisor>
+ getLecturers(): List<Lecturer>
+ getCourses(): List<Course>
+ getStudentIDStudentMap(): Map<Integer, Student>
+ getCourseCodeCourseMap(): Map<String, Course>
+ getCourseSections(): List<CourseSection>
+ getTranscripts(): List<Transcript>
+ getMaxCourseNumber(): byte
+ getSectionCourseMap(): Map<CourseSection, Course>
+ getUserNamePersonMap(): Map<String, Person>
+ getSectionCodeCourseMap(): Map<String, Course>
+ getAdvisors(): List<Advisor>
```

Course

```
+ Course(String, String, int, byte):
- courseName: String
- courseSections: List<CourseSection>
- courseCredit: int
- gradeLevel: byte
- lecturer: Lecturer
- courseCode: String
- preRequisiteCourses: List<Course>
+ setCourseSections(List<CourseSection>):
+ getGradeLevel(): byte
+ getCourseCode(): String
+ getPreRequisiteCourses(): List<Course>
+ setPreRequisiteCourses(List<Course>): vo
+ getCourseName(): String
+ getCourseSections(): List
+ addPreRequisiteCourse(Course): void
```

Transcript

```
+ Transcript(Student):
- cgpa: double
- gradeLevel: byte
- takenCredits: int
- courseGradeMap: Map<Course, Grade>
- completedCredits: int
- student: Student
- studentCourses: List<Course>
+ calculateTakenCredits(): int
+ setTakenCredits(int): void
+ showTranscript(): void
+ calculateValues(): void
+ calculateCgpa(): double
+ readTranscript(String): Student
+ setCourseGradeMap(Map<Course, Grade>):
+ setStudent(Student): void
+ courseStatusCheck(): void
+ calculateCompletedCredits(): int
+ getCourseGradeMap(): Map<Course, Grade
+ getStudentCourses(): List<Course>
+ setStudentCourses(List<Course>): void
```

Lecturer

```
+ Lecturer(int, String, String, String, String):
+ Lecturer(int, String, String):
~ lessonsTaught: List<Course>
~ login(String, String): boolean
+ getLessonsTaught(): List<Course>
```

<<interface>>
IDisplayMenu

```
+ printMenu(String): void
```

CourseSection

```
+ CourseSection(Course, String, String, int):
- courseSectionCode: String
- students: List
```

Student

```
+ Student(int, String, String, String, String, byte):
- gradeLevel: byte
- transcript: Transcript
- courseSectionMap: Map<Course, CourseSection>
- availableCourses: List<Course>
- advisor: Advisor
- draft: List<Course>
- hasRequest: boolean
+ setPassword(String): void
+ printMenu(String): void
+ getDraft(): List<Course>
+ getTranscript(): Transcript
+ addCourse(): void
+ getAvailableCourses(): List<Course>
+ getDepartmentName(): String
+ setUserName(String): void
+ getGradeLevel(): byte
~ login(String, String): boolean
+ setDraft(List<Course>): void
+ setHasRequest(boolean): void
+ isHasRequest(): boolean
+ showRequestStatus(): void
+ getCourseSectionMap(): Map<Course, CourseSection>
+ dropCourse(): void
+ viewAvailableCourses(): void
+ setTranscript(Transcript): void
+ setAdvisor(Advisor): void
+ sendRequest(): void
+ getAdvisor(): Advisor
```

CourseRegistrationSystem

```
+ CourseRegistrationSystem():
- choice: int
- input: Scanner
+ studentMenu(Student): void
+ loginMenu(Department): void
+ advisorMenu(Advisor): void
+ printMenu(String): void
+ courseSelectionMenu(Student): void
+ getInput(): int
+ mainMenu(Department): void
+ exitProgram(): void
+ start(): void
```

Advisor

```
+ Advisor(int, String, String, String, String):
- requestList: List<Registration>
- requestNumber: int
- studentsAdvised: List<Student>
+ addRequest(Registration): void
+ setRequestNumber(int): void
+ printRequests(): void
+ replyRequests(): void
+ getRequestList(): List<Registration>
+ printMenu(String): void
~ login(String, String): boolean
+ getStudentsAdvised(): List<Student>
+ getRequests(): List<Registration>
```

JSONReader

```
+ JSONReader():
~ studentRegistrationMap: Map<Student, Registra
~ lecturerIDCoursesMap: Map<Integer, List<String
~ jsonNode: JsonNode
~ studentIDAdvisorIDMap: Map<Integer, Integer>
~ coursePrerequisiteCourseCodesMap: Map<Cou
~ mapper: ObjectMapper
~ courseSectionCodesMap: Map<Course, List<St
~ studentIDDraftMap: Map<Integer, List<String>>
~ advisorIDCoursesMap: Map<Integer, List<String
+ readRequests(Department): void
+ start(Department): void
+ writeJson(): void
```

Registration

```
+ Registration(Student, List<Course>):
- student: Student
- courses: List<Course>
+ approveRequest(Advisor): void
+ rejectRequest(Advisor): void
+ showRequest(): void
+ getStudent(): Student
```

Grade

```
+ Grade(String):
- letterGrade: String
- gradeOver4: double
- gradeOver100: int
+ getLetterGrade(): String
- setGradeOver100(): void
- setGradeOver4(): void
```

main

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