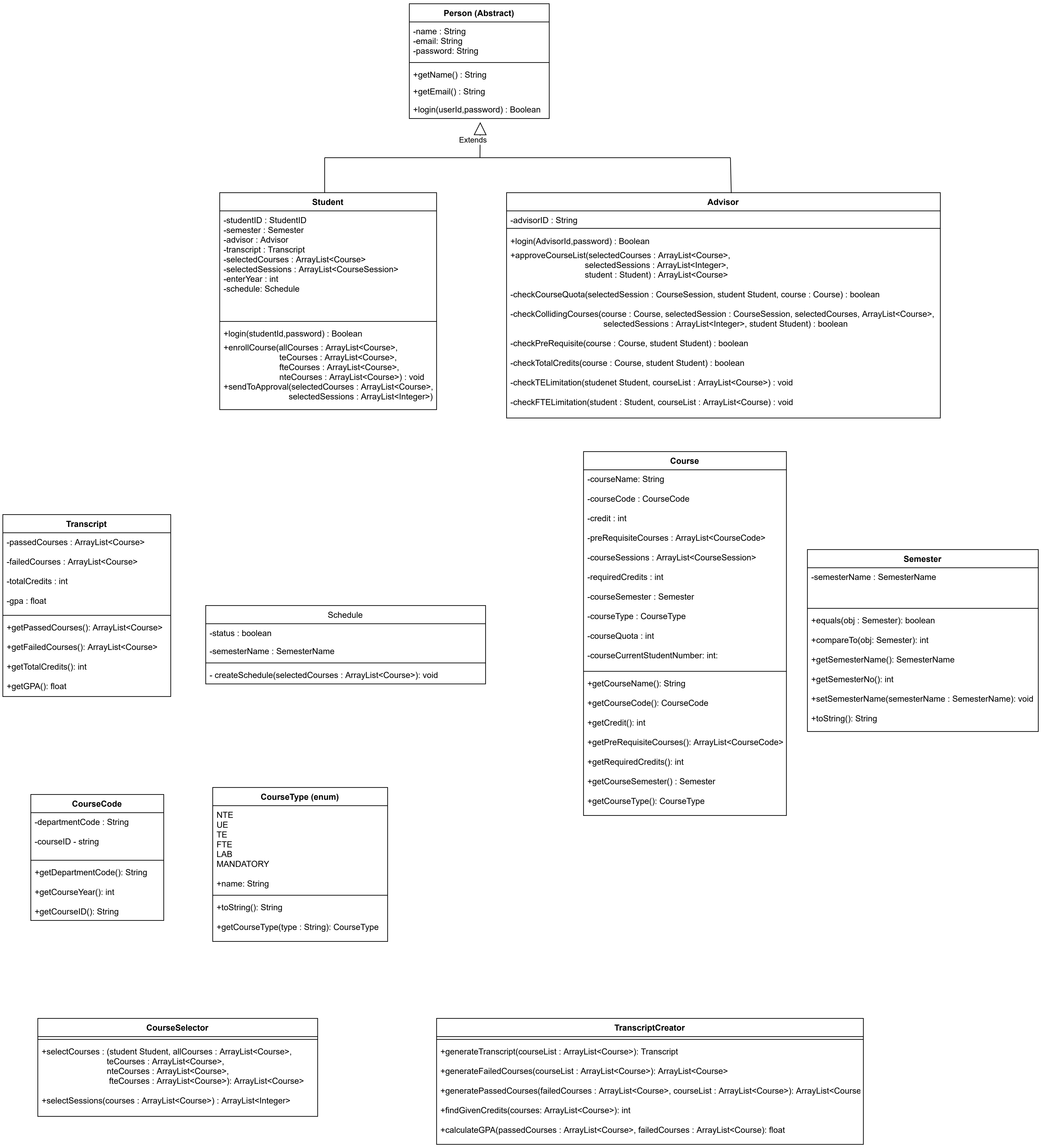

Person (Abstract)

-name : String
-email: String
-password: String

+getName() : String
+getEmail() : String
+login(userId,password) : Boolean

Student

-studentID : StudentID
-semester : Semester
-advisor : Advisor
-transcript : Transcript
-selectedCourses : ArrayList<Course>
-selectedSessions : ArrayList<CourseSession>
-enterYear : int
-schedule: Schedule

+login(studentId,password) : Boolean
+enrollCourse(allCourses : ArrayList<Course>,
teCourses : ArrayList<Course>,
fteCourses : ArrayList<Course>,
nteCourses : ArrayList<Course>) : void
+sendToApproval(selectedCourses : ArrayList<Course>,
selectedSessions : ArrayList<Integer>)

Advisor

-advisorID : String

+login(AdvisorId,password) : Boolean
+approveCourseList(selectedCourses : ArrayList<Course>,
selectedSessions : ArrayList<Integer>,
student : Student) : ArrayList<Course>
-checkCourseQuota(selectedSession : CourseSession, student Student, course : Course) : boolean
-checkCollidingCourses(course : Course, selectedSession : CourseSession, selectedCourses, ArrayList<Course>,
selectedSessions : ArrayList<Integer>, student Student) : boolean
-checkPreRequisite(course : Course, student Student) : boolean
-checkTotalCredits(course : Course, student Student) : boolean
-checkTELimitation(studenet Student, courseList : ArrayList<Course>) : void
-checkFTELimitation(student : Student, courseList : ArrayList<Course) : void

Course

-courseName: String
-courseCode : CourseCode
-credit : int
-preRequisiteCourses : ArrayList<CourseCode>
-courseSessions : ArrayList<CourseSession>
-requiredCredits : int
-courseSemester : Semester
-courseType : CourseType
-courseQuota : int
-courseCurrentStudentNumber: int:

+getCourseName(): String
+getCourseCode(): CourseCode
+getCredit(): int
+getPreRequisiteCourses(): ArrayList<CourseCode>
+getRequiredCredits(): int
+getCourseSemester() : Semester
+getCourseType(): CourseType

Transcript

-passedCourses : ArrayList<Course>
-failedCourses : ArrayList<Course>
-totalCredits : int
-gpa : float

+getPassedCourses(): ArrayList<Course>
+getFailedCourses(): ArrayList<Course>
+getTotalCredits(): int
+getGPA(): float

Schedule

-status : boolean
-semesterName : SemesterName

- createSchedule(selectedCourses : ArrayList<Course>): void

Semester

-semesterName : SemesterName

+equals(obj : Semester): boolean
+compareTo(obj: Semester): int
+getSemesterName(): SemesterName
+getSemesterNo(): int
+setSemesterName(semesterName : SemesterName): void
+toString(): String

CourseCode

-departmentCode : String
-courseID - string

+getDepartmentCode(): String
+getCourseYear(): int
+getCourseID(): String

CourseType (enum)

NTE
UE
TE
FTE
LAB
MANDATORY

+name: String

+toString(): String
+getCourseType(type : String): CourseType

CourseSelector

+selectCourses : (student Student, allCourses : ArrayList<Course>,
teCourses : ArrayList<Course>,
nteCourses : ArrayList<Course>,
fteCourses : ArrayList<Course>): ArrayList<Course>
+selectSessions(courses : ArrayList<Course>) : ArrayList<Integer>

TranscriptCreator

+generateTranscript(courseList : ArrayList<Course>): Transcript
+generateFailedCourses(courseList : ArrayList<Course>): ArrayList<Course>
+generatePassedCourses(failedCourses : ArrayList<Course>, courseList : ArrayList<Course>): ArrayList<Course
+findGivenCredits(courses: ArrayList<Course>): int
+calculateGPA(passedCourses : ArrayList<Course>, failedCourses : ArrayList<Course): float