

# CSE 102 Programming Assignment 4

Page 1 of 4

## Due:

Friday, 24-May-2024 by 23:59

## Deliverables:

The following Java file should be submitted to Google Classroom by the due date and time specified above. Submissions received after the deadline will be subject to the late policy described in the syllabus.

- Assignment04\_{StudentNumber}.java

## Specifications:

**Overview:** You will continue the program this semester to maintain the information for a university. Do not forget your headers with @author and @since information. You may create any additional classes, methods, or attributes that you think necessary.

**Requirements:** Write and modify the following set of classes (All constructors should take parameters in the order given and initialize empty collections):

1. Department
  - a. No change from Assignment 2
2. Course
  - a. No change from Assignment 2
3. Person - abstract
  - a. No change from Assignment 2
4. Teacher – a child of Person
  - a. Attributes
    - i. No change from Assignment 2
  - b. Methods
    - i. No change from Assignment 2 except getTitle(): String – returns the following based on rank value
      1. Lecturer
      2. Adjunct Instructor
      3. Assistant Professor
      4. Associate Professor
      5. Professor
5. Student – a child of Person
  - a. Attributes
    - i. No change from Assignment 2
  - b. Methods
    - i. addCourse(course: Course, semester: Semester, grade: double): None
      1. Keeps a record of courses the student has taken and the grades the student has earned
      2. If course has already been taken in the passed semester, replaces existing grade
      3. Raises InvalidGradeException if grade is negative or greater than 100

- ii. `getAKTS(): int`  
`getAttemptedAKTS(): int`
    - 1. These should now only count courses once even if the student has taken a course more than once
  - iii. `courseGPAPoints(course: Course): double`  
`courseGradeLetter(course: Course): String`  
`courseResult(course: Course): String`
    - 1. these now return the result for the course based on the highest score the student earned
  - iv. `listGrades(semester: Semester): String`
    - 1. for the semester passed returns a string consisting of Courses and respective grades
    - 2. If no courses can be found for the passed Semester, raises `SemesterNotFoundException`
  - v. `listGrades(course: Course): String`
    - 1. for the Course passed returns a string consisting of Semesters and respective grades
    - 2. If the passed course cannot be found, raises `CourseNotFoundException`
  - vi. `transcript(): String` – returns the courses, letter grades and semester GPA based on semester ordered by Semester. An example is shown below.
6. `GradStudent` – a child of `Student`
- a. Attributes
    - i. No change from Assignment 2
  - b. Methods
    - i. `setTeachingAssistant(course: Course): None`
      - 1. adds the `GradStudent` as the teaching assistant for the passed course
      - 2. A `GradStudent` will only be allowed to be the teaching assistant for at most one course, but they are not required to be teaching assistant at all.
      - 3. If the `GradStudent` has passed the course with at least an 80, they can be added
      - 4. If the `GradStudent` has not yet taken the course or the highest grade they have earned in this course is less than 80, raises a `CourseNotFoundException`
    - ii. `getTeachingAssistant(): Course`
      - 1. returns the course that this `GradStudent` is a teaching assistant for
7. `Semester` – will be immutable (cannot change values once they are set initially)
- a. Attributes
    - i. You will determine any necessary attributes
  - b. Methods

- i. Constructor that takes the two integer parameters for season (can be 1, 2 or 3) and year
  - ii. `getSeason(): String`
    - 1. Fall
    - 2. Spring
    - 3. Summer
  - iii. `getYear(): int` – returns Year
  - iv. `toString(): String` – returns `{getSeason()} + " – " + {Year}`
- 8. Custom Exceptions (All must be instances of `RuntimeException`)
  - a. `SemesterNotFoundException`
    - i. Additional Attributes – student: Student, semester: Semester
    - ii. `toString()` "SemesterNotFoundException: " + {student ID} + " has not taken any courses in " + {semester}

**Design:** Your program does not require a main method. You are only responsible for creating the seven (7) classes and six (6) Exceptions (the 5 from Assignment 2 and the one given here) described above.

**Code:** The file you submit will be named `Assignment04_{StudentNumber}`. You should put all java classes for this assignment inside of this one (1) file as discussed in class.

**Test:** You are responsible for testing your program. It is important to not rely solely on the examples presented in this Assignment description.

```
public static void main(String[] args) {  
  
    Department cse = new Department("CSE", "Computer Engineering");  
    Teacher teacher = new Teacher("Joseph LEDET", "josephledet@akdeniz.edu.tr", 123L, cse, 3);  
    System.out.println(teacher);  
    Student stu = new Student("Assignment 4 STUDENT", "me@somewhere.com", 456L, cse);  
    Semester s1 = new Semester(1, 2020);  
    Course c101 = new Course(cse, 101, "Programming 1", "Introduction", 6, teacher);  
    Semester s2 = new Semester(2, 2021);  
    Course c102 = new Course(cse, 102, "Programming 2", "OOP", 4, teacher);  
    Course c204 = new Course(cse, 204, "Database Systems", "DBMS", 6, teacher);  
  
    stu.addCourse(c101, s1, 80);  
    stu.addCourse(c102, s2, 30);  
    stu.addCourse(c204, s2, 70);  
    System.out.println("List Grades for CSE101:\n" + stu.listGrades(c101));  
    System.out.println("List Grades for Spring 2021:\n" + stu.listGrades(s2));  
    System.out.println("Student Transcript:\n" + stu.transcript());  
  
    GradStudent gs = new GradStudent("Assignment 4 GRADSTUDENT", "me@somewhere.com", 789L, cse, 2, "MDE");  
    gs.addCourse(c101, s1, 85);  
    gs.addCourse(c102, s1, 40);  
    gs.setTeachingAssistant(c101);  
    System.out.println("Teaching Assistant:\n" + gs.getTeachingAssistant());  
    gs.setTeachingAssistant(c102);  
    System.out.println("Teaching Assistant:\n" + gs.getTeachingAssistant());  
}
```

```
Assistant Professor Joseph LEDET (123) - josephledet@akdeniz.edu.tr
List Grades for CSE 101:
Fall - 2020 - BB
List Grades for Spring 2021:
CSE102 - FF
CSE204 - CB
Student Transcript:
Fall - 2020
    CSE101 - BB
GPA: - 3.0

Spring - 2021
    CSE102 - FF
    CSE204 - CB
GPA: - 1.5

Overall GPA: 2.0625
Teaching Assistant:
CSE101 - Programming 1 (6)
Exception in thread "main" CourseNotFoundException: 789 has not yet taken CSE102
    at GradStudent.setTeachingAssistant(Assignment04_123456789.java:423)
    at Assignment04_123456789.main(Assignment04_123456789.java:29)
```

## Grading:

**MS Teams Submission:** If anything is ambiguous, it is your responsibility to ask questions. It is also your responsibility to complete this assignment in a timely manner. Questions regarding this assignment will likely not be answered if received after 17:00 on the due date of the assignment.