

Due:

Friday, 15-March-2024 by 23:59

Deliverables:

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

- Assignment01_{StudentNumber}.java
- Image files for screenshots of your progress:
 - May be separate image files or all pasted into a .docx or .pdf
 - 1. Assignment01_{StudentNumber}_firstCompile
 - 2. Assignment01_{StudentNumber}_firstCompileNoError
 - 3. Assignment01_{StudentNumber}_firstTestRun
 - 4. Assignment01_{StudentNumber}_firstTestRunNoError (optional)
 - 5. Assignment01_{StudentNumber}_finalCompile
 - 6. Assignment01_{StudentNumber}_finalRun

Specifications:

Overview: You will continue the program this semester to maintain the grades for a student in a course. Do not forget your headers with @author and @since information.

Requirements: Write a set of classes according to the following specifications:

1. Course
 - a. Attributes
 - i. Department Code: String
 1. Must be 3 or 4 characters
 - ii. Course Number: int
 1. Must be in the range 100-999 or 5000-5999 or 7000-7999
 - iii. Title: String
 - iv. Description: String
 - v. AKTS: int
 1. Must be positive
 - b. Methods
 - i. Constructor that takes the department code, number, title, description and AKTS as parameters
 - ii. getDepartmentCode(): String and setDepartmentCode(departmentCode: String)
 - iii. getCourseNumber(): int and setCourseNumber(number: int)
 - iv. getTitle(): String and setTitle(title: String)
 - v. getDescription(): String and setDescription(title: String)
 - vi. getAKTS(): int and setAKTS(AKTS: int)
 - vii. courseCode(): String
 1. returns the department code and course number with no space between (i.e. "CSE102")
 - viii. toString(): String – "{department code}{number} - {title} ({AKTS})"

2. Person

a. Attributes

- i. Name: String
- ii. Email: String
 - 1. Must be of the form {username}@{university name}.{domain}
- iii. ID: long int
- iv. Department Code: String
 - 1. Must be 3 or 4 characters

b. Methods

- i. Constructor that takes the name, email, number, and department code as parameters
- ii. getName(): String and setName(name: String)
- iii. getEmail(): String and setEmail(email: String)
- iv. getID(): long int and setID(ID: long int)
- v. getDepartmentCode(): String and setDepartmentCode(department code: String)
- vi. toString(): String – “{Name} ({ID}) – {Email}”

3. Teacher – a child of Person

a. Attributes

- i. Rank: int

b. Methods

- i. Constructor that takes the name, email, number, department, and rank as parameters
- ii. setRank (rank: int)
 - 1. Must be between 1 and 4 (the possible ranks)
- iii. getTitle(): String – returns the following based on rank value
 - 1. Lecturer
 - 2. Assistant Professor
 - 3. Associate Professor
 - 4. Professor
- iv. promote(): None
 - 1. Increases the status of the Teacher (i.e. increments rank)
 - 2. Make sure that rank remains a valid value
- v. demote(): None
 - 1. Decreases the status of the Teacher (i.e. decrements rank)
 - 2. Make sure that rank remains a valid value
- vi. toString(): String – {title} + toString() of parent

4. Student – a child of Person

a. Attributes

- i. AKTS: int

b. Methods

- i. Constructor that takes the name, email, number, and department as parameters and sets AKTS as 0.

- ii. getAKTS(): int
 - 1. returns AKTS
 - iii. passCourse(course: Course): None
 - 1. Adds the AKTS for the course to the student's total
 - iv. toString(): String – inherits from parent
5. GradStudent – a child of Student
- a. Attributes
 - i. Rank: int
 - ii. Thesis topic: String
 - b. Methods
 - i. Constructor that takes the name, email, number, department, rank, and thesis topic as parameters
 - ii. setRank(rank: int)
 - 1. Must be 1, 2, or 3
 - iii. getLevel(): String – returns the following based on rank value
 - 1. Master's Student
 - 2. Doctoral Student
 - 3. Doctoral Candidate
 - iv. getThesisTopic(): String and setThesisTopic(Thesis Topic: String)
 - v. toString(): String – inherits from parent

Special Note: Any invalid value should result in a Run-time error. An example is shown below.

Design: Your program does not require a main method. You are only responsible for creating the five (5) classes described above. An example of how your program should operate is given below:

```
1 public class Assignment01_123456789 {
2     public static void main(String[] args) {
3         Course c = new Course("CSE", 102, "Programming 2", "Introduction to OOP", 6);
4         System.out.println(c.courseCode() + " - " + c.getTitle());
5         System.out.println(c);
6
7         Teacher t = new Teacher("Joseph LEDET", "josephledet@akdeniz.edu.tr", 123L, "CSE", 1);
8         System.out.println(t);
9
10        Student s = new Student("Test STUDENT", "me@somewhere.com", 456L, "CSE");
11        System.out.println(s);
12
13        s.passCourse(c);
14        System.out.println(s.getAKTS());
15
16        System.out.println("-----");
17
18        Course course = new Course("CSE", 101, "Computer Programming 1", "Introduction to Programming", 6);
19        Student student = new Student("Can DO", "cando@akdeniz.edu.tr", 123L, "CSE");
20        student.passCourse(course);
21        course.setCourseNumber(course.getCourseNumber() + 10);
22        System.out.println(student);
23        System.out.println(course);
24        course = new Course("CSE", 102, "Computer Programming 2", "Introduction to OOP", 4);
25        student.passCourse(course);
26        course.setCourseNumber(course.getCourseNumber() - 10);
27        System.out.println(course);
28        System.out.println(student);
29    }
30 }
```

Initially, the following output and error were encountered. The error is due to trying to set a course number to an invalid value.

```
CSE102 - Programming 2
CSE102 - Programming 2 (6)
Lecturer Joseph LEDET (123) - josephledet@akdeniz.edu.tr
Test STUDENT (456) - me@somewhere.com
6
-----
Can D0 (123) - cando@akdeniz.edu.tr
CSE111 - Computer Programming 1 (6)
Exception in thread "main"
at Course.setCourseNumber(Assignment01_123456789.java:55)
```

Once that line was commented out, this output is produced:

```
CSE102 - Programming 2
CSE102 - Programming 2 (6)
Lecturer Joseph LEDET (123) - josephledet@akdeniz.edu.tr
Test STUDENT (456) - me@somewhere.com
6
-----
Can D0 (123) - cando@akdeniz.edu.tr
CSE111 - Computer Programming 1 (6)
CSE102 - Computer Programming 2 (4)
Can D0 (123) - cando@akdeniz.edu.tr
```

Code: The file you submit will be named Assignment01_{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.

Grading:

MS Teams Submission: If anything is ambiguous, it is your responsibility to ask questions and 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.

Screenshots: For this assignment, you must provide at minimum the six listed screenshots of your progress. An example of the screen shot for first compile is given below. An example of the screen shot for final run is shown above as the example output. These screenshots must include the entire screen (window, task bar, etc.). **Note:** if you do not submit these images, your score on this assignment **will be 0**.

