

Problem 3: Company & Departments (composition) 122A Class Diagram sept a selend ber your Company departments: List & Department >1 tadd Department (dep: Department): void 1000 + remove Department (dep: Department): void 1 Problem 2: Sonk and Account Holder 1-employees: List & Employee > 1 Hadd Employee (emp: Employee): void !-1 + remove Employee (emp: Employee): void! temployee ! !

1-name: String!

1-position & string!

Self-Problems & Students with Courses (Association Problem 1. School (and Aggregation) Class Diagram School | + addstudent (student: Student) void | 21 nontrages 1 + remove Student (Student: Student): void! : pholosof 1 + or going (formath : localfo): no Student |-courses: List < Coulse>| 1 tenvoll (course: (ourse): void! 1 + view Courses (): List & Course > Infile 1 1994 Association (many-to-many) 1-students: List < Student >1 1+addstudent (student: Student): void 1 1+view Students (): List < Student > 1

1 noblem 28 University with Faculties & Departments (composition and Aggregation) Class Diagram University. List < Department > 1 to 12> + 3]; Harburg 1-faculty: Ust < faculty >= 112 + + add Department (dept : Department): void liner ; □ Composition Department 1-title: String

Kroblem 3: Hospital, Doctors, and Patients (Association Howbord hon susmater) Hospital I-nance: String |-doctors: List < Doctor> 1 1-partients: List < Partient> t +add Doctor (doc: Doctor): void l'faddlatient (pat: latient): void |- specialty: string! |- patients: List < latient> 1+ Consult (patient: Patient): 2) Association (many to-ma Patient 1 - eloctors: List < Doctor>

E-Commerce Platform with Quiders, Customers and Products Customer 1+placebrder Corder Order |-product: List < Product> 1 + add Product (product : Product): void! 1 + get Total () : floats. product |-product ID: String | |-name: string ! |-price: float

```
Knoblem 5: Univois9ty Management System
   Class Diagram
   Student
  1-Student ID: String,1
   |-name: string |
|-courses: list<(ourse>)
   | +envollaurse (course: course): void |
    + view (ourse (): List < course>
        --- + Association (many-to-many)
    Aggregation
     Profesor |

1 - profesor ID: String |

1 - profesor ID: String |

1 - name: String |

1 - name: List < Course>1
    1 + assign (ourse (course; (ourse): void)
1 + view (ourse (): List < (ourse > )
```