Domain Model Description

- A Company has many Employees. Employee has monthly salary.
- Employee can have one of these Designation – Senior Developer, Developer or Associate.
- The Yearly Bonus structure for the employee depends on the designation
    - For Senior Developer, Annual Bonus is 3 months salary
    - For Developer, Annual Bonus is 2 months salary
    - For Associate, Annual Bonus is 1 months salary
- Each Employee also has one of these project roles – Java Programmer, SQL Programmer, C Programmer.
- Each Employee has a Manager to whom he/she reports.
- A Manager is also employee of the company.
- Manager also has monthly salary. Annual bonus for Manager is 4 months of monthly salary
- A Manager has many employees (and at least 1 employee) reporting to him/her.
- Employee can have at most 3 addresses. The addresses possible are permanent address, current address and office address. Each Address can have as many appropriate fields

TODO:

1. Print all managers in a company
2. Print all employees reporting to each manager
3. Print address slips for each employees
4. Create a UML Class Diagram and
5. Create a UML sequence Diagram for the significant flows

Remember the modus operandi:

1. No need to code elaborate data input mechanisms. Just create data in the main itself.
2. Do not code specific data in the Domain Model Do not put specific employee manager objects etc. in Domain Model
3. Follow `AAA of Code-Test = Arrange, Act, Assert` for creating data and then acting on model to process data
4. Use type safe classes instead of plain enums/typedef (For samples see – Statement.h and Statement.cpp - attached)

Hints:

Use Composite Pattern

Use CRC Techniques

Use inheritance only when required

Model the roles as composition instead of folding back as inheritance into the composing class