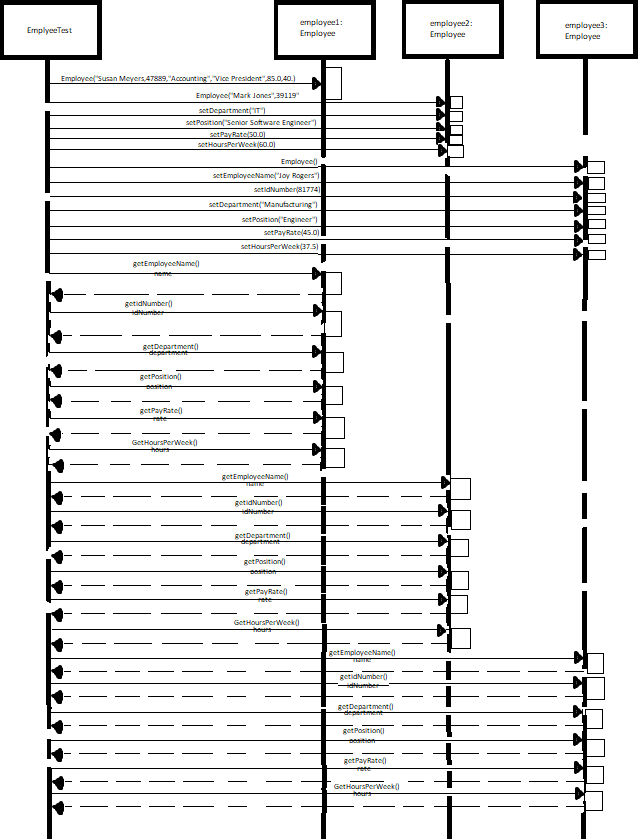
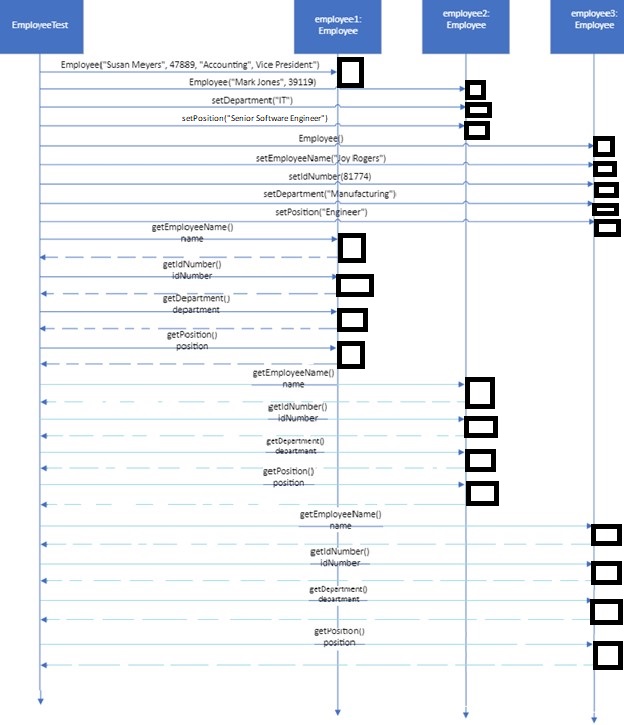
|  |
| --- |
| Employee |
| -employeeName: str  -idNumber: int  -department: str  -position: str |
| +Employee()  +Employee(name: String, id: int)  +Employee(name: String, id: int, dept: String, pos: String)  +setEmployeeName(name: String):void  +setIdNumber(id: int):void  +setDepartment(dept: String):void  +setPosition(pos: String):void  +getEmployeeName(): String  +getIdNumber(): int  +getDepartment(): String  +getPosition(): String |
| Maintaining list of employees.  Adding an employee with 3 different constructors. |

|  |
| --- |
| Employee1: Employee |
| name="Susan Meyers"  idNumber=47899  department="Accounting"  position="Vice President" |

|  |
| --- |
| Employee2: Employee |
| name="Mark Jones"  idNumber=39119  department="IT"  position="Programmer" |

|  |
| --- |
| Employee3: Employee |
| name="Joy Rogers"  idNumber=81774  department="Manufacturing"  position="Engineer" |



a) If you had only the class diagram and no code to work from, give the public and private members of the class and explain how you know which is which.

|  |  |
| --- | --- |
| Public members | Private members |
| Employee()  Employee(name: String, Id: int)  Employee(name: String, Id: int, dept: String, pos: String)  setEmployeeName(name: String): void  setIdNumber(Id: int): void  setDepartment(dept: String): void  setPosition(pos: String): void  getEmployeeName(): String  getIdNumber(): int  getDepartment(): String  getPosition(): String | EmployeeName: String  IdNumber: int  department: String  position: String |

Looking at a class diagram public members are listed with a (+) before the variable while private variables have a (-).

b) Describe the scope of the parameter to setIDNumber() and the scope of idNumber. List the methods in which each is visible.

A variable is only local to its method, but a class wide variable such as idNumber is available to all the methods within the class.

|  |  |
| --- | --- |
| setIdNumber parameter | idNumber |
| Only within setIdNumber. | All class methods:  Employee(),  Employee(name: String, id: int),  Employee(name: String, id: int, dept: String, pos: String)  setEmployeeName(name: String): void,  setIdNumber(id: int): void,  setDepartment(dept: String): void,  setPosition(pos: String): void  getEmployeeName(): String,  getIdNumber(): int,  getDepartment(): String,  getPosition(): String |

c) Are there any overloaded methods? Which? When reading the code, how do you know they're overloaded? How does the compiler know which one to bind to a method call?

Yes, the constructor is overloaded. It is overloaded since the methods are named the same but have different information. The compiler knows which to bind to a method call by comparing the data. The code with the same amount of variables input will get bound to the same type of call.

d) List the methods that have results.

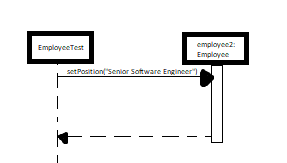
getEmployeeName(): String

getIdNumber(): int

getDepartment(): String

getPosition(): String

e) Show how the object diagram for the Mark Jones Employee object (called employee2 here) looks after the following takes place:



|  |
| --- |
| employee2: Employee |
| name = “Mark Jones”  idNumber = “39119” department = “IT”  position = “Senior Software Engineer” |