

Madhuri Pyreddy
12/12/19

CSCI 117 Lab 15

```
1a). fun {New2 Obj} Temp in
  class Temp from Obj      % inheritance
    meth init() skip      % init method
  end
end
{New Obj init()}          % function return
end
```

```
1b). fun {New2 Obj} A Temp in
  A = {NewName}
  class Temp from Obj      % inheritance
    meth !A(init)          % init method
    skip end
  end
  {New Obj A}              % function return
end
```

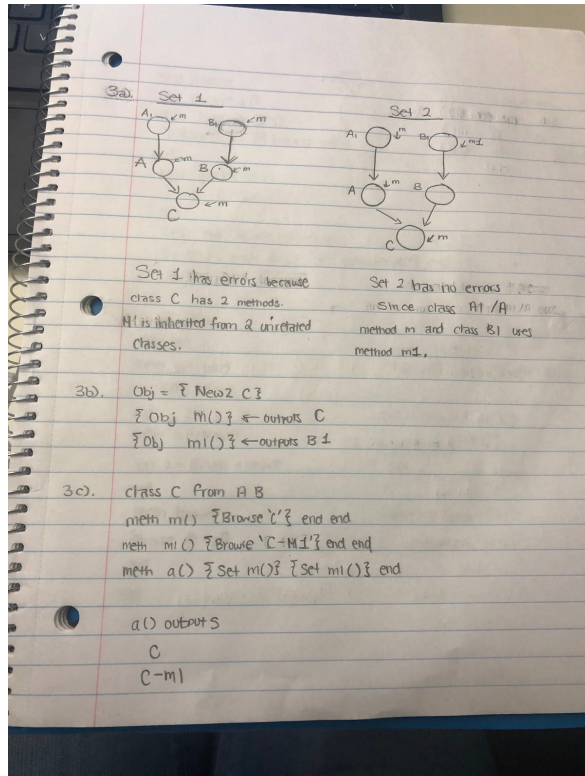
```
2. local
proc {Transfer2 M S}
  transfer2(Amt) = M in
    @(S.balance):= Amt + @S.balance
  end
proc {GetBalance M S}
  getBalance(Bal) = M in
    Bal = @(S.balance)
  end
proc {BatchTransfer M S}
  batchtransfer{AmtList} M in
for A in AmtList do {Transfer transfer(A) S} end
  end
proc {Transfer M S}
  transfer (Amt) = M in
    {LogObj addentry(transfer(Amt))}
    {Transfer2 transfer2(A) S}
  end
in
```

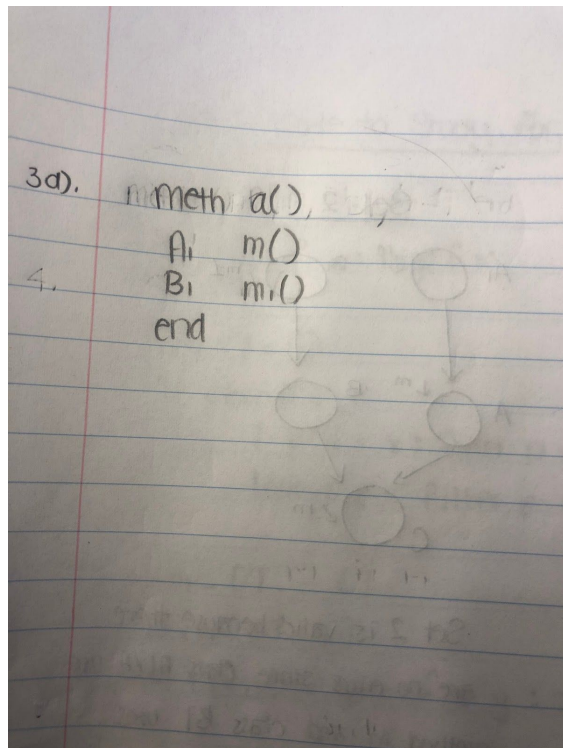
```

LoggedAccount=I(attrs:[balance])
                methods:m(transfer:Transfer getBalance:GetBalance
batchTransfer:BatchTransfer)
end

```

3.





4. The example for forwarding on page 518 shows that the object can forward to any object. The argument *M* is a first-class message that can be passed to another object. The example defines *NewF*, a version of *New* that creates objects that can forward the class. Objects created with *NewF* have a method *setForward(F)* that lets them set dynamically the object to which they will forward messages they do not understand. The example calls *Obj1* and *Obj2*. Each object keeps its own local state, that *Obj2* “inherits” the *inc* and *c* methods from object *Obj1*, and that *Obj2* “overrides” the *browse* method. *ObjX* inherits all its behavior from *Obj2*. It is identical to *Obj2* except that it has a different local state. The delegation hierarchy now has three levels: *ObjX*, *Obj2*, and *Obj1*. By changing *ObjX* delegate to *Obj1*, *ObjX* inherits its behavior from *Obj1* in the new hierarchy. In foreshadowing, *Obj1* and *Obj2* keep their separate identities whereas Delegation allows *Obj2* to self call to call *Obj1*. Forwarding does not imply a common self unlike delegation.