

Study HW # 4.

Sorry for the typo in # 3's
code

no declaration $D d$;
should have had $d = new E()$
because F is not subclass of D .

Study Hw #4. Scoping, Call Graph Construction

#1. main

```

a: integer = 1
b: integer = 2
c: integer = 3

proc p():
  b: integer = 5
  proc s():
    print a, b, c ①
  end s
  proc t():
    b: integer = 4
    a = 5
    s()
    print a, b, c ②
  end t
  q()
  t()
end p

proc q():
  print a, b, c ③
end q

p()
print a, b, c ④
end main

```

#1a. with static scoping what's the output?

- ③ ⇒ 1, 2, 3
- ① ⇒ 5, 5, 3
- ② ⇒ 5, 4, 3
- ④ ⇒ 5, 2, 3

#1b. with dynamic scoping what's the output?

- ③ ⇒ 1, 5, 3
- ① ⇒ 5, 4, 3
- ② ⇒ 5, 4, 3
- ④ ⇒ 5, 2, 3

#2. What is the output of the Scheme program shown?

- (a) with Scheme's static scoping?
- (b) assuming dynamic scoping?

```
(define A (lambda (z)
  (let* ((x 2)
        (C (lambda (P) (let ((x 4)) (P 0))))
        (D (lambda (q) x))
        (E (lambda (r)
             (let ((x 3)) (C D))))
        (E 0))))
```

(A 0)

1st we are evaluating A on ϕ (zero).

2nd the let* expression is $(E \phi)$ where E ignores its parameter and evaluates the let - $(let ((x 3)) (C D))$ in its body.

C ~~ignores~~ defines $(x 4)$ and then returns its argument applied to ϕ .

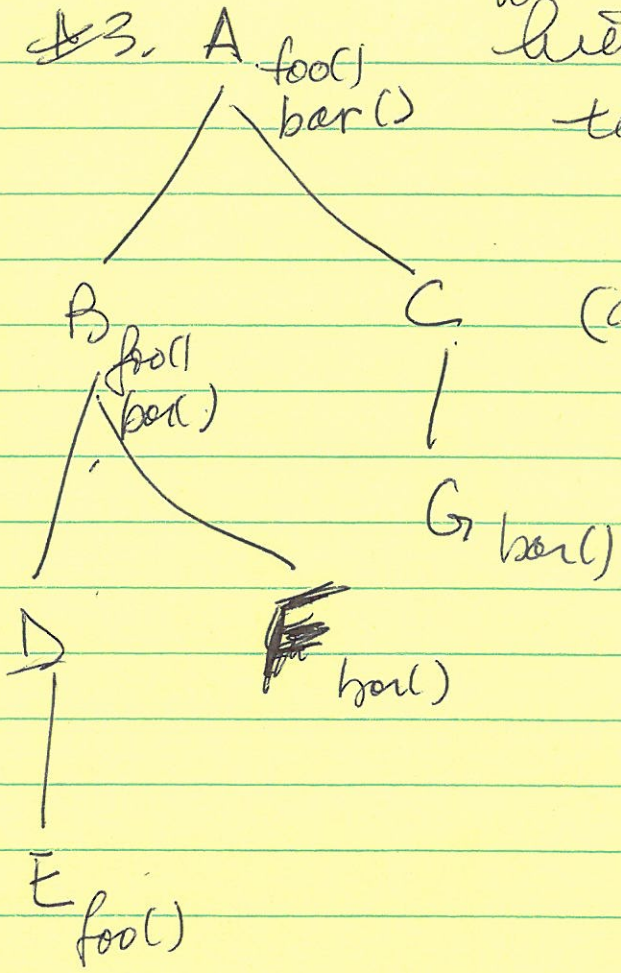
So $(C D)$ evaluates to $(D \phi)$.

D ignores its argument and returns x.

with static scoping this is A's x with value 2

with dynamic scoping, we have the following environments defining new values for x: $A \rightarrow E \rightarrow D$ so it is E's x with value 3.

with the hierarchy given and its functions, tell which functions are called.



(a) using C++ and then (b) RTA resolve the virtual calls below

```

A a1, a2;
B b;
C c; D d;
a1 = new A();
a2 = new A();
c = new G();
d = new F(); typo
      new E();
#1: a1.foo();
#2: a2.bar();
#3: d.foo();
#4: b.bar();
  
```

~~Sorry~~

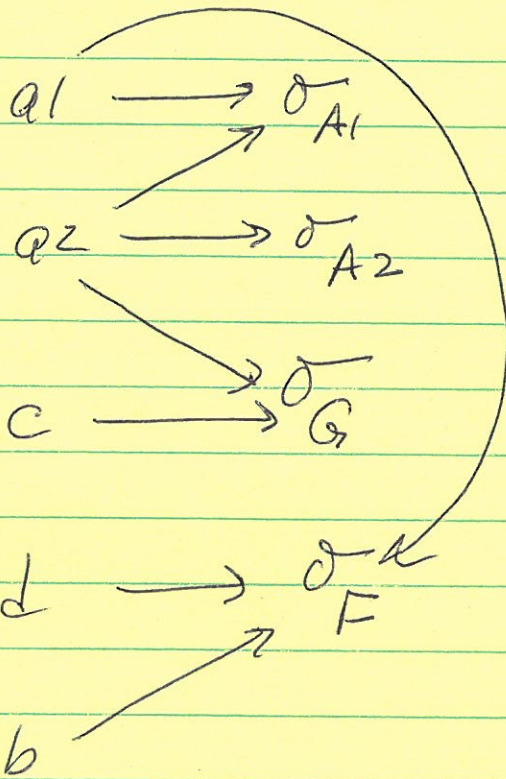
	C++	RTA
#1. a1.foo()	A.foo(), B.foo(), E.foo()	A.foo()
#2. a2.bar()	A.bar(), B.bar(), F.bar(), G.bar()	A.bar(), F.bar(), G.bar()
#3. d.foo()	B.foo(), E.foo()	
#4. b.bar()	B.bar(), F.bar()	

2 ERROR-TYPE

How to fix?
if ~~a2~~ = new E();
then this would have been.

E.foo()
& E.bar() for #4.

#4. Draw points-to graph for program (as given with the types) - assuming no pointer assignment statements in the methods called.



$a2 = a1$ creates

$a2 \rightarrow \sigma_{A1}$ edge

$b = d$ creates

$b \rightarrow \sigma_F$ edge

$a1 = b$ creates

$a1 \rightarrow \sigma_F$ edge

$a2 = c$ creates

$a2 \rightarrow \sigma_G$ edge

final pts-to graph