

②

(define m 50)

(define n 100)

(define (hardy))

(display list " In Hardy, n=" n))
(new line)
define (leavel n)

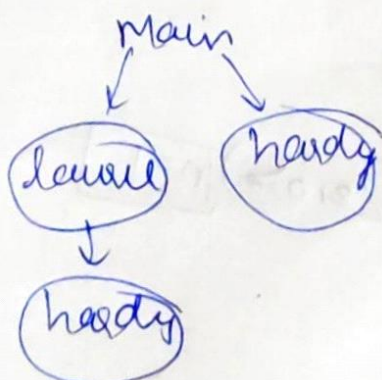
(display list " In Leavel, m=" m))
(new line)

(display (list " In leavel, n=" n))
(new line)
(hardy))

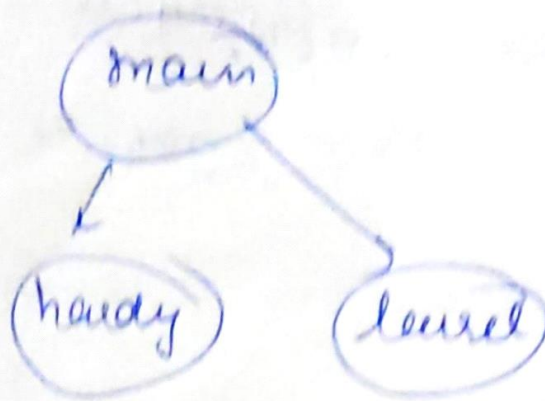
or

(leavel 1)

(hardy)



Call graph



nested relationship

Scope	n	n
main	$\langle 1, 0 \rangle$	$\langle 1, 4 \rangle$
laural	$\langle 1, 0 \rangle$	$\langle 2, 0 \rangle$
hardy	$\langle 1, 0 \rangle$	$\langle 2, 0 \rangle$
hardy	$\langle 1, 0 \rangle$	$\langle 1, 4 \rangle$

dynamic coordinates

$$m\langle 1, 0 \rangle = 50$$

$$n\langle 1, 4 \rangle = 100$$

$$\langle 1, 4 \rangle = 100 \text{ ~~is the~~$$

value of expression is 100 obtained ($n\langle 1, 4 \rangle$)

can laural 1 $n\langle 2, 0 \rangle$ (redeclared value of n)

display in laural $m=50$ ($m\langle 1, 0 \rangle$)

display in laural $n=1$ ($n\langle 2, 0 \rangle$)

calling hardy()

display in hardy ($n\langle 2, 0 \rangle$) $n=1$

return to laural

return to main

calling hardy

display in hardy $n=100$ has saying

$(n(114)) = 100$

return main()

scope	m	n
main	$\angle 100$	$\angle 114$
laural	$\angle 100$	$\angle 100$
hardy	$\angle 100$	$\angle 114$

static coordinate

$n(114) = 100$

value of mp is 100 ($n(114)$)

~~value~~ call laural

display in laural $m=50$ ($m(110)$)

display in laural $n=100$ | $n(210)$
call hardy()

display in hardy $n(114)$

return to laural

return to main

call hardy

display $n < 100$

= 100

Since it is statically scoped
whatever value in laurd is not
known to hardy

① begin

integer global n;

procedure laurd (n: integer);

begin

procedure hardy

begin

print (global);

print n;

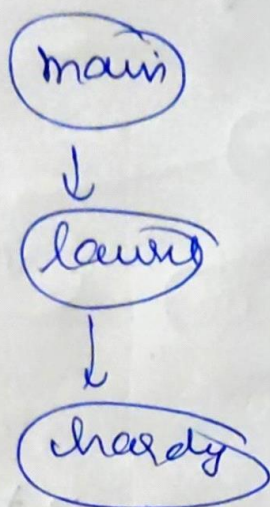
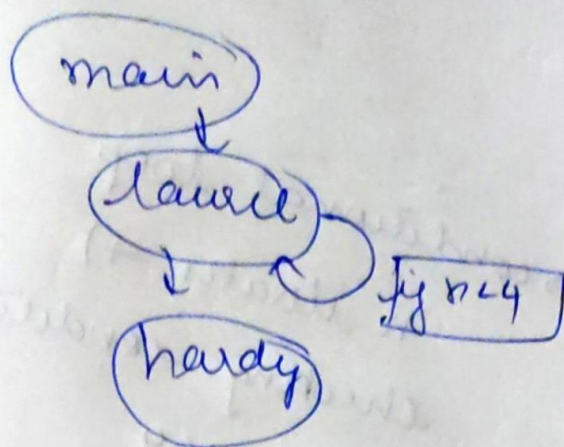
end

if $n < 100$ then laurd (n)

in hardy

end

Call graph



nested relationship graph

global :- g

$g[110] = 99$

$m[114] = 100$

} global, n

calling laurel(1) n[210]

Checking condition $n < 4$ laurel n[310]

calling laurel again

Only calling hardy.

In hardy's display $g[110]$

n[210]

After loop completes exit

scope	g	n
main	g(10)	n(14)
laurel	g(10)	n(210)
laurel	g(10)	n(310)
hardy	g(10)	n(210)

conditional loop
n iterates +
checking cond
if $n < 4$

dynamic coordinates

global g : 99 g(10)

n(14) = 100

call laurel n(210) = 1

* checks condition $\text{laurel}(n(10) + 1)$
= n(310)

statically scoped variable check

or in
hardy

In hardy display g(10) and
n(210)

scope	g	n
main	g(10)	n(14)
laurel	g(10)	n(210)
hardy	g(10)	n(210)

static coordinates

② begin

procedure test (n: integer, p: procedure);

begin

procedure run;

begin

print ("in procedure run -- n");

print (n);

end;

print ("in procedure test -- n = ");

print (n);

p;

if n < 10 then

begin

if n = 3 then

test (n+1, run)

test (n+1, run)

run test (n+1)

end

end;

procedure visit;

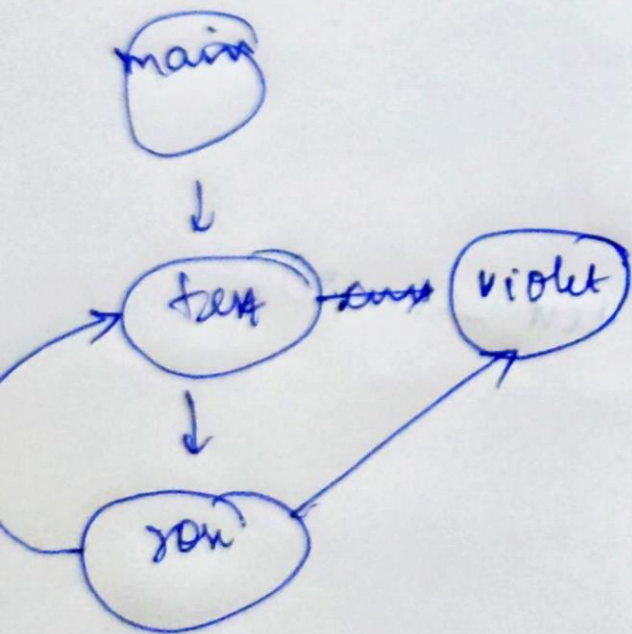
begin

print ("in procedure visit");

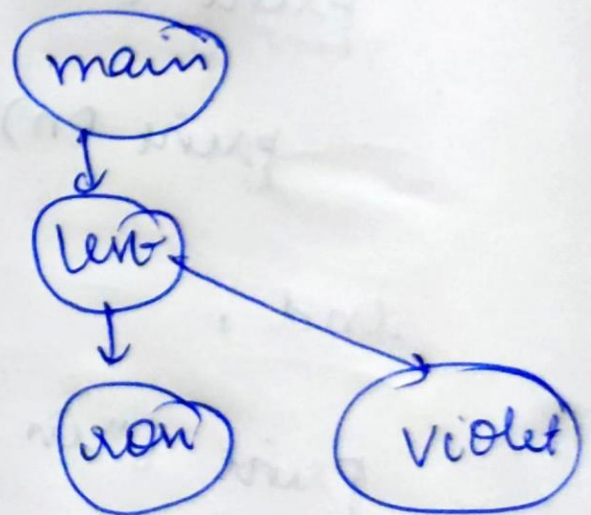
and;

Print Involved);

end.



call graph



nested relationship