E |- Integer get(int key){...}: Integer

E[key -> int]

E[find return -> Integer]

E[find p1 -> int]

1. E |- HashEntry found = find(key);:void

E[found -> HashEntry]

a. E |- found = find(key) : void

i. E |- found : tau (tau -> HashEntry)

E(found) = HashEntry

ii. E |- find(key) : sigma (sigma -> HashEntry)

(1) E |- key : omega (omega -> int)

E(key) = int

(2) E |-find param1 : phi (phi -> int)

E(find param1) = int

(3) phi := omega (int := int)

(4) E |- find return : sigma (sigma -> HashEntry)

E(find ret) = HashEntry

iii. tau := sigma (HashEntry := HashEntry)

2. E |- if (found == null) {...} else {...};: void

a. E |- found == null : boolean

i. E |- found : tau (tau -> HashEntry)

E(found) = HashEntry

ii. E |- null : sigma (sigma -> null)

E(null) = null

iii. tau :== sigma (HashEntry :== null)

b. E |- return null; : void

i. E |- null : sigma (sigma -> null)

E(null) = null

ii. E |- find return: tau (tau -> Integer)

E(find ret) = Integer

iii. tau := sigma (Integer := null)

c. E |- return found.value; : void

i. E |- found.value : tau (tau -> Integer)

E(found.value) = Integer

ii. E |- find return : sigma (sigma -> Integer)

E(find ret) = Integer

iii. tau := sigma (Integer := Integer)