**Maintaining Maps**

Here is what the maps look like:

class World {

int sum;

int value;

// value |-> int, sum |-> int

void add(int foo) {

// foo |-> int, value |-> int, sum |-> int

string z;

// z |-> string, foo |-> int, value |-> int, sum |-> int

sum = sum + value;

value = 0;

}

// value |-> int, sum |-> int

void main(string bar) {

// bar |-> string, value |-> int, sum |-> int

int y;

// y |-> int, bar |-> string, value |-> int, sum |-> int

sum = 0;

value = 10;

add();

// y |-> int, bar |-> string, value |-> int, sum |-> int

if (sum % 3 == 1) {

string value;

// value |-> string, y |-> int, bar |-> string, (((value |-> int))), sum |-> int

value = 1;

add();

print("inner value = ", value);

print("sum = ", sum);

}

// y |-> int, bar |-> string, value |-> int, sum |-> int

print("outer value = ", value);

}

}

* Note how the outer declaration of int value is shadowed by the inner declaration of string value
* Note how map becomes bigger as we enter more scopes, but later becomes smaller again
* If we processed statements sequentially imperatively, we need to make maps bigger and later smaller again
  + alternatively we can use functional programming style and keep the old version until it is needed again