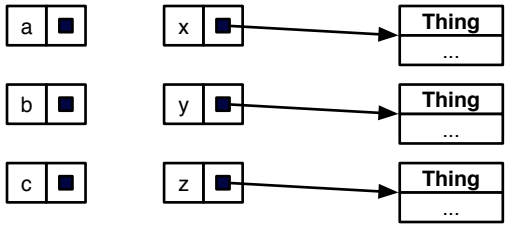
Garbage collection

handout for CS 302 by Will Benton (willb@cs)

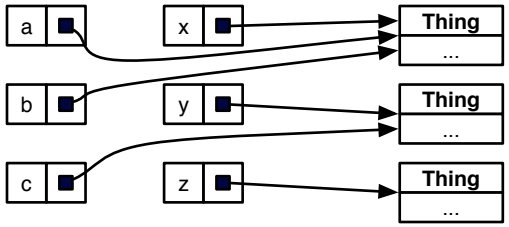
You may have noticed that Java provides a way to create objects -- that is, operator `new` -- but no way to destroy them. The Java language frees programmers from the responsibility to manually destroy objects when they are no longer needed. Instead, the Java language mandates a feature called a garbage collector. The garbage collector runs alongside your program, automatically finds objects that are no longer in use, and reclaims any memory that they were using.

Garbage objects are those that are no longer live. A live object is one that is transitively reachable from a variable currently in scope. (This means that a live object is one that can be referred to by a variable currently in scope, or by a field of any live object.) Once an object becomes garbage, it cannot become live again, and may be collected at any point in the future. (Do not assume that garbage will be collected immediately.)

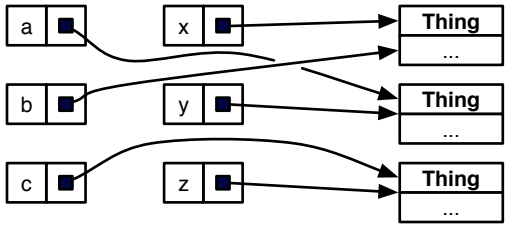
```
Thing a, b, c;
Thing x = new Thing();
Thing y = new Thing();
Thing z = new Thing();
/* point 1 */

a = x;
b = a;
c = y;
/* point 2 */

a = y;
c = z;
/* point 3 */

y = z;
/* point 4 */

a = b;
/* point 5 */

return;
/* point 6 */
```



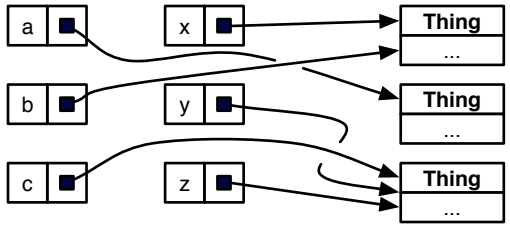
1. At this point, `x`, `y`, and `z` all refer to newly-allocated `Things`. `a`, `b`, and `c` have not received values yet.



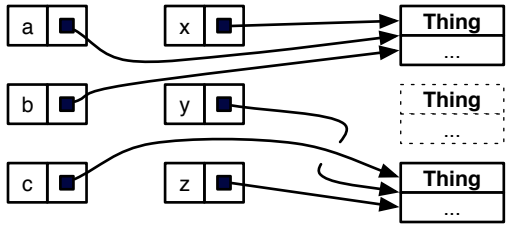
2. After the next three assignments, `a` refers to the same object as `x`, `b` refers to the same object as `a`, and `c` refers to the same object as `y`.



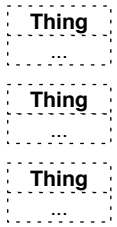
3. Note that assigning a new reference to `a` does not affect the referents of the other variables that had referred to the top `Thing` (as did `a`).



4. `y` no longer refers to the second `Thing`. However, since at least one variable (`a`) still refers to it, the second `Thing` is not garbage.



5. After changing which `Thing` `a` refers to, there are now no active references to the second `Thing`. It may be reclaimed at any point now.



6. When a method returns, all of its local variables go away. With no extant references to any of these `Things`, they are all considered garbage.