

## CS 1044 Design Example: Eating 10 Cashews

---

### The Problem:

Assume you have three bowls, called Bowl 1, Bowl 2 and Bowl 3. Initially, Bowl 1 contains a mixture of three kinds of nuts: cashews, peanuts and walnuts. (At least ten nuts of each kind are present.) You detest peanuts and walnuts, but you love cashews. You must devise an algorithm for eating precisely 10 cashews. Your algorithm must satisfy the following criteria:

- the user will eat exactly 10 cashews, and eat no other kind of nut;
- all the uneaten nuts must be in Bowl 1 when the process terminates;
- the process must be guaranteed to terminate in finite time.

You are allowed to perform only the following actions and must use the specified (functional) notation in your solution:

- pick up a (random) nut (from Bowl N):                      `GetNut ( N )`
- eat the nut you are holding:                                      `EatNut ( )`
- place the nut you are holding in Bowl N:                      `PutNut ( N )`  
(where N is 1, 2 or 3)
- pour all the nuts in Bowl M into Bowl N:                      `PourNuts ( M, N )`  
(where M and N are 1, 2 or 3 and not equal)

You also can tell whether a bowl is empty and what kind of nut you are holding:

- check if Bowl N is empty:                                      `IsEmpty ( N )`  
(equals **false** if Bowl N is not empty and **true** if Bowl N is empty)
- check the type of nut you are holding:                      `IsPeanut ( ) , IsCashew ( ) , IsWalnut ( )`  
(equals **false** if nut is not of the specified type and **true** if nut is of specified type)

You are allowed to hold only one nut at a time.

### Outline:

- I. No cashews have been eaten.
- II. As long as fewer than 10 cashews have been eaten:
  - A. Take a nut from Bowl 1.
  - B. If it's a cashew,
    - 1. Eat it.
    - 2. Remember that another cashew has been eaten.
  - C. If it's not a cashew, put it into Bowl 2.
- III. Pour all the nuts in Bowl 2 back into Bowl 1.

### Pseudocode Solution:

```
NumCashewsEaten <-- 0           // No cashews have been eaten yet

While ( NumCashewsEaten < 10 ) Do // Keep going until 10 have been eaten
    GetNut( 1 )                  // Pick up a nut

    If ( IsCashew( ) ) Then      // See if it's a cashew
        EatNut( )               // If it is, eat it
        Increment NumCashewsEaten // and count it
    Else
        PutNut( 2 )             // If it's not a cashew,
                                // put it in a different bowl
    Endif

Endwhile

PourNuts(2, 1)                  // Put all those non-cashews back
                                // into Bowl 1
```

### Comments:

- If you pick up a non-cashew, you must put it into a different bowl (than Bowl 1); otherwise there's no guarantee you'll ever finish, since you could pick up the same nut over and over.
- IsCashew( ) doesn't get a nut; it just tells you whether the nut you're holding is a cashew.
- Whether Bowls 2 and 3 were initially empty was unspecified. However, when the question was raised in class, I said it was OK to assume so.
- You don't have to test to see if Bowl 1 becomes empty, because it was specified that there were initially at least 10 cashews. That makes it impossible to empty Bowl 1 before you've eaten 10 cashews.
- GetNut( ) selects a random nut — you cannot specify the type of nut to be picked.
- GetNut( ) can be used only to remove a random nut from Bowl 1, not from either of the other bowls.

### What's wrong with this?

```
NumCashewsEaten <-- 0           // No cashews have been eaten yet
While ( NumCashewsEaten < 10 ) Do // Keep going until 10 have been eaten
    GetNut( 1 )                 // Pick up a nut
    If ( IsCashew( ) ) Then      // See if it's a cashew
        EatNut( )               // If it is, eat it
        Increment NumCashewsEaten // and count it
    Else
        PutNut( 1 )             // If it's not a cashew,
                                // put it in a different bowl
    Endif
Endwhile
PourNuts(2, 1)                 // Put all those non-cashews back
                                // into Bowl 1
```