# Contracts for objects -- 1

Black Box Specification

# Information hiding

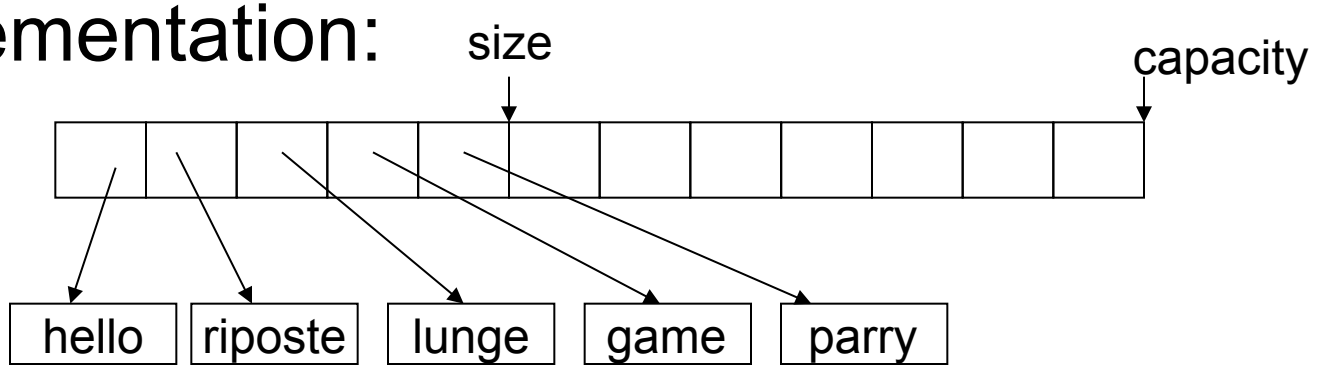
- Recall that so far our class contracts have been in terms of the actual fields of the class.
  - This breaks information hiding. The details of how a class is implemented are typically not something the client coder should worry about, nor should their code depend on these.
  - Interfaces can not be specified by such contracts.

# Model Fields

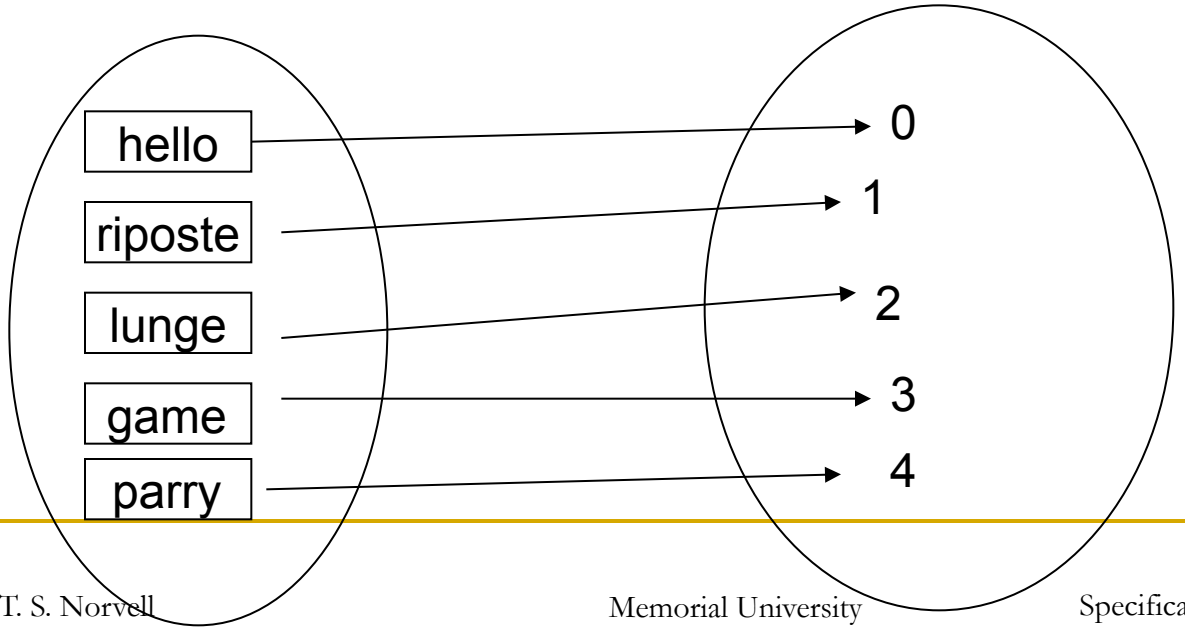
- One solution is to use fictitious fields.
- These are fields that describe the object's state in a client oriented way
- Consider the Dictionary class from the previous slide set.
- What its fields represent is a function from strings to integers.
- We call such field "model fields"

# Dictionary revisited

Actual implementation:




What the client is thinking in terms of:



# Dictionary revisited

```
interface DictionaryI {
    // model field f is a function from a finite set of
    //                  strings to nonnegative integers
    // invariant  f is one-one (i.e. different strings map
    // to different integers)
    // model field int capacity
    // invariant capacity >= the size of domain(f)
```

# Dictionary revisited

```
// ensures result == capacity
int getCapacity() ;

// ensures result == the size of domain(f)
int getSize() ;

// requires str != null
// ensures if str equals any string domain(f)
//                 then result == f(str) else result == -1
int getInt( String str ) ;
```

# Dictionary Revisited

```
// requires str != null and getSize() < getCapacity()
// modifies f
// ensures domain(f') == domain(f) union {str}
//   and f'(str) == result
//   and (for all s in domain(f), f'(s)==f(s) )
int putString( String str ) ;
}
```

# The Dictionary class and DictionaryI

- Everything that DictionaryI promises the Dictionary class delivers
- Let's make a small change to Dictionary

  **class** Dictionary **implements** DictionaryI {
- We can see that Dictionary realizes DictionaryI semantically as well as syntactically.
- I.e. the LSP is respected. Any object of class Dictionary satisfies the expectations of any client who expects an object of type DictionaryI

# Black boxes

- We call DictionaryI a black box specification of Dictionary.
- It (partially) specifies the behaviour of Dictionary objects without revealing their implementation.
- A client that relies only on the guarantees provided by DictionaryI will not be broken if we change the implementation of Dictionary.