

**6.00 Handout, Lecture 10**  
**(Not intended to make sense outside of lecture)**

```
numBuckets = 47 #this is ugly. We will see a better way soon
```

```
def create():
    global numBuckets
    hSet = []
    for i in range(numBuckets):
        hSet.append([])
    return hSet

def hashElem(e):
    global numBuckets
    return e%numBuckets

def insert(hSet, i):
    hSet[hashElem(i)].append(i)

def remove(hSet, i):
    newBucket = []
    for j in hSet[hashElem(i)]:
        if j != i:
            newBucket.append(j)
    hSet[hashElem(i)] = newBucket

def member(hSet, i):
    return i in hSet[hashElem(i)]
```

-----

```
def hashElem(e):
    global numBuckets
    if type(e) == int:
        val = e
    if type(e) == str:
        #Convert e to an int
        val = 0
        shift = 0
        for c in e:
            val = val + shift*ord(c)
            shift += 1
    return val%numBuckets
```

```
numBuckets = 47

def test1():
    s = create()
    for i in range(40):
        insert(s, i)
    insert(s, 325)
    insert(s, 325)
    insert(s, 987654321)
    print s
    print member(s, 325)
    remove(s, 325)
    print member(s, 325)
    print member(s, 987654321)
```

```
def test2():
    d = create()
    strs = ['ab', 'ba', '32a',
            'big dog', 'small bird']
    for s in strs:
        insert(d, s)
    for i in range(40):
        insert(d, i)
    print d
    print member(d, 'small bird')
    print member(d, 'big bird')
    remove(d, 'small bird')
    print d
```

```
def readVal(valType, requestMsg, errorMsg):
    numTries = 0
    while numTries < 4:
        val = raw_input(requestMsg)
        try:
            val = valType(val)
            return val
        except ValueError:
            print(errorMsg)
            numTries += 1
    raise TypeError('Num tries exceeded')

print readVal(int, 'Enter int: ', 'Not an int.')

try:
    readVal(int, 'Enter int: ', 'Not an int.')
except TypeError, s:
    print s
```

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.00SC Introduction to Computer Science and Programming  
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.