

ORF 201

COMPUTER METHODS FOR PROBLEM SOLVING

Lecture 16  
Hashing - Continued



© 1996 by Randy Glasbergen. E-mail: randyg@norwich.net

(C) Princeton University

# PUHasher 3

Class **HashTable** developed in **PUHasher2** is fundamental and general - should be enshrined for future use

- Remove classes **ObjLink** and **HashTable** from **PUHasher2.java**.
- Put them into a new file **HashTable.java**.
- Put this file into a *public* directory: **.../public\_html/JAVA/myutil**.
- Add the line:

```
package myutil;
```

telling the compiler where this file is in relation to **codebase** and **\$CLASSPATH**.

- In **PUHasher3**, be sure to import **myutil.HashTable**.

# PUHasher 4

Java already has an easy to use hashtable class.

- Replace **import myutil.HashTable** with **import java.util.Hashtable**
- Note that class name has a lowercase "t" in **Hashtable**.
- Replace calls to **ht.lookup()** with calls to **ht.get()**.
- Replace calls to **ht.add()** with calls to **ht.put()**.

## Class.java. util.Hashtable

See **file:/usr/java/webdocs/api/java.util.Hashtable.html#\_top\_**  
for more info

# Two Dimensional Arrays

Arrays allow one to store a list of data elements:

$$x_1, x_2, \dots, x_n \leftrightarrow \mathbf{x}[1], \mathbf{x}[2], \dots, \mathbf{x}[n].$$

Suppose we wish to store data elements that have two subscripts, that is, a two-dimensional array of data:

$$\begin{array}{cccc} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{array}$$

```
int m=5, n=6;
double[][] a;           // declare
    \
a = new double[m+1][n+1]; // instantiate
    \
for (int i=1; i<=m; i++) {
    for (int j=1; j<=n; j++) {
        a[i][j] = Math.pow(i,j); // initialize
    }
}
    \
for (int i=1; i<=m; i++) {
    for (int j=1; j<= n; j++) {
        System.out.print(a[i][j]); // use
    }
}
```