```
 1 /*-----------------------------\
 2 |Nick Rebhun                  |
 3 |Project 06                   |
 4 |Computer Science 182         |
 5 |05/31/2011                   |
 6 |                             |
 7 \-----------------------------*/
 8
 9 package project6;
10
11 ////////////////////////////////////////////////////////////////
12 class DataItem
13    {                                // (could have more data)
14    private String iData;               // data item (key)
15 //--------------------------------------------------------------
16    public DataItem(String ii)          // constructor
17       { iData = ii; }
18 //--------------------------------------------------------------
19    public String getKey()
20       { return iData; }
21 //--------------------------------------------------------------
22    }  // end class DataItem
23 ////////////////////////////////////////////////////////////////
24 class HashTable
25    {
26    private DataItem[] hashArray;    // array holds hash table
27    private int arraySize;
28    private DataItem nonItem;        // for deleted items
29 // --------------------------------------------------------------
30    public HashTable(int size)       // constructor
31       {
32       arraySize = size;
33       hashArray = new DataItem[arraySize];
34       nonItem = new DataItem("-1");   // deleted item key is -1
35       }
36 // --------------------------------------------------------------
37    public void displayTable()
38       {
39       System.out.print("Table: ");
40       for(int j=0; j<arraySize; j++)
41          {
42          if(hashArray[j] != null)
43             System.out.print(hashArray[j].getKey() + " ");
44          else
45             System.out.print("** ");
46          }
47       System.out.println("");
48       }
49 // --------------------------------------------------------------
50    public int hashFunc3(String key)
51       {
52        int hashVal = 0;
53        for(int j=0; j<key.length(); j++)    // left to right
54        {
55            int letter = key.charAt(j) - 96; // get char code
56            hashVal = (hashVal * 27 + letter) % arraySize;   //mod
57        }
58        return hashVal;                      // no mod
59       }
60 // --------------------------------------------------------------
61    public void insert(DataItem item) // insert a DataItem
62    // (assumes table not full)
63       {
64       String key = item.getKey();      // extract key
```

```
 65       int hashVal = hashFunc3(key);  // hash the key
 66                                      // until empty cell or -1,
 67       while(hashArray[hashVal] != null &&
 68                       hashArray[hashVal].getKey() != "-1")
 69          {
 70          ++hashVal;                  // go to next cell
 71          hashVal %= arraySize;       // wraparound if necessary
 72          }
 73       hashArray[hashVal] = item;     // insert item
 74       }  // end insert()
 75 // --------------------------------------------------------------
 76    public DataItem delete(String key)  // delete a DataItem
 77       {
 78       int hashVal = hashFunc3(key);  // hash the key
 79
 80       while(hashArray[hashVal] != null)  // until empty cell,
 81          {                               // found the key?
 82          if(hashArray[hashVal].getKey() == key)
 83             {
 84             DataItem temp = hashArray[hashVal]; // save item
 85             hashArray[hashVal] = nonItem;       // delete item
 86             return temp;                        // return item
 87             }
 88          ++hashVal;                  // go to next cell
 89          hashVal %= arraySize;       // wraparound if necessary
 90          }
 91       return null;                   // can't find item
 92       }  // end delete()
 93 // --------------------------------------------------------------
 94    public DataItem find(String key)    // find item with key
 95       {
 96       int hashVal = hashFunc3(key);  // hash the key
 97
 98       while(hashArray[hashVal] != null)  // until empty cell,
 99          {                               // found the key?
100          if(hashArray[hashVal].getKey() == key)
101             return hashArray[hashVal];   // yes, return item
102          ++hashVal;                  // go to next cell
103          hashVal %= arraySize;       // wraparound if necessary
104          }
105       return null;                   // can't find item
106       }
107 // --------------------------------------------------------------
108    }  // end class HashTable
109 ////////////////////////////////////////////////////////////////
```