

```

.....
Lab12Status.txt
.....
Problem 1: compiles, runs correctly on all provided input
Problem 2: compiles, runs correctly on all provided input
.....
Lab12Conclusions.txt
.....
I learned how Horner's rule can be used to efficiently create HashTables as Horner
's rule is a technique used for quickly evaluating polynomials. Horner's rule can
be used to compute a hash code for a given key.....
Problem0.txt
.....
To compute the nth power of 32 efficiently using only one multiplication operation
, we can use the property that  $32^n$ 
is equivalent to  $(2^5)^n$ , which simplifies to  $2^{(5n)}$ . Therefore, we can compute th
e nth power of 32 as:

int nthpowerof32 = 1 << (5*n); // using bit shift instead of multiplication

To compute the rank of an upper-case character in alphabetic order, we can subtrac
t the Unicode value of
'A' from the Unicode value of the character and add 1. This works because the Unic
ode values of the
uppercase letters are assigned in consecutive order.

char c = 'D'; // example character
int rank = c - 'A' + 1;
.....
ChainNode.java
.....
/*
 * Purpose: Data Structure and Algorithms Lab 12
 * Status: Complete and thoroughly tested
 * Last update: 4/21/23
 * Submitted: 4/21/23
 * Comment: Test suite and run sample attached.
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.04.21
 */
class ChainNode<K extends Comparable<? super K>, V> {
    private K key;
    private V value;
    private ChainNode<K, V> next;

    public ChainNode(K newKey, V newValue,
                    ChainNode<K, V> nextNode) {
        key = newKey;
        value = newValue;
        next = nextNode;
    } // end constructor

    public V getValue() {
        return value;
    } // end getValue

    public void setValue(V value) {
        this.value = value;
    } // end setValue

```

```

public K getKey() {
    return key;
} // end getKey

public ChainNode<K, V> getNext()
{
    return next;
} // end getNext

public void setNext(ChainNode<K, V> next)
{
    this.next=next;
} // end setNext

public String toString() //use (key,value) format
{
    String str = "{" + getKey() + ", " + getValue() + "}";
    return str;
}
} // end ChainNode::.....
HashTable.java
.....
/*
 * Purpose: Data Structure and Algorithms Lab 12
 * Status: Complete and thoroughly tested
 * Last update: 4/21/23
 * Submitted: 4/21/23
 * Comment: Test suite and run sample attached.
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.04.21
 */

// *****
// Hash table implementation.
// Assumption: A table contains unique keys with possibly non-unique associated va
lues
// *****

public class HashTable<K extends Comparable<? super K>, V> implements HashTableInt
erface<K,V> {
    private ChainNode[] primary; // hash table's primary array
    private int size= 0; // number of entries (number of (key,value) p
airs)

    public HashTable() {
        primary = new ChainNode[3];
    } // end default constructor

    // table operations
    public boolean tableIsEmpty() {
        return size==0;
    } // end tableIsEmpty

    public int tableSize() {
        return size;
    } // end tableSize = number of (key,value) pairs in the HashTable

    //implement the following 5 methods

    // if key is not already in HashTable pair (key,value) is inserted and returns

```

```

true
// if key is already in the HashTable it does not re-insert or overwrite and r
eturns false
public boolean tableInsert(K key, V value)
{
    if(tableRetrieve(key) == null)
    {
        int index = hashCode(key) % primary.length;
        ChainNode newNode = new ChainNode(key, value, primary[index]);
        primary[index] = newNode;
        size++;
        return true;
    }

    else
    {
        return false;
    }
}

// if searchKey is not in the HashTable returns null
// otherwise deletes the searchKey and its association from the HashTable and
// returns the previous value associated with searchKey
public V tableDelete(K searchKey)
{
    V result = null;
    int index = hashCode(searchKey) % primary.length;
    ChainNode curr = primary[index];
    if(curr == null)
    {
        return null;
    }

    if(curr.getKey().equals(searchKey))
    {
        result = (V) curr.getValue();
        primary[index] = curr.getNext();
    }

    else
    {
        while(curr.getNext() != null && !(curr.getNext().getKey().equals(searchKey)))
        {
            curr = curr.getNext();
        }

        if(curr.getNext() != null)
        {
            result = (V) curr.getNext().getValue();
            curr.setNext(curr.getNext().getNext());
        }
    }

    if(result != null)
    {
        size--;
    }

    return result;
}

//returns the value associated with searchKey in the HashTable or null if the
searchKey is not in the HashTable

```

```

public V tableRetrieve(K searchKey)
{
    V result = null;
    for(int index = 0; index < primary.length; index++)
    {
        ChainNode curr = primary[index];
        while(curr != null)
        {
            if (((String) curr.getKey()).compareTo(((String) searchKey)) == 0)
            {
                return (V) curr.getValue();
            }
            curr = curr.getNext();
        }
    }

    return result;
}

// returns the integer hashCode computed using Horner's rule - assumes K is String
public int hashCode(K key)
{
    ((String)key).toUpperCase();
    int sum = 0;
    for (int index = 0; index < ((String) key).length(); index++)
    {
        char c = ((String) key).charAt(index);
        int rank = c - 'A' + 1;
        int nthpowerof32 = 1 << (((String)key).length() - index - 1) * 5);
        sum += nthpowerof32 * rank;
    }

    return sum;
}

//returns the String representation of the HashTable [all (key,value) pairs in the HashTable]
public String toString()
{
    StringBuilder sb = new StringBuilder();
    for(int index = 0; index < primary.length; index++)
    {
        ChainNode curr = primary[index];
        while(curr != null)
        {
            sb.append(curr.toString()).append(" ");
            curr = curr.getNext();
        }
    }

    return sb.toString();
}

} // end HashTable
:::::::::::::
Lab12P3Driver.java
:::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 12
 * Status: Complete and thoroughly tested
 * Last update: 4/21/23
 * Submitted: 4/21/23
 * Comment: Test suite and run sample attached.

```

```
* Comment: I declare that this is entirely my own work
* @author: Antonio Rosado
* @version: 2023.04.21
*/
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab12P3Driver
{
    private static BufferedReader stdin = new BufferedReader(new InputStreamReader
(System.in));
    public static void main (String[] args) throws IOException
    {
        HashTable table = new HashTable();
        boolean exit = false;
        while (!exit)
        {
            System.out.println("Select from the following menu: \n"
                + "0. Exit the program \n"
                + "1. Insert a [key, associated value] pair in the
table. \n"
                + "2. Delete a pair from the table. \n"
                + "3. Retrieve and display the value associated wit
h a key in the table. \n"
                + "4. Display the hashCode of a key. \n"
                + "5. Display the content of the table. \n");

            System.out.print("Make your menu selection now: " );
            int input = Integer.parseInt(stdin.readLine());
            System.out.println(input);
            // possible cases for initial input
            switch (input)
            {
                case 0:
                    System.out.println("Exiting program... good bye");
                    exit = true;
                    break;

                case 1:
                    insert(table);
                    break;

                case 2:
                    delete(table);
                    break;

                case 3:
                    search(table);
                    break;

                case 4:
                    hashCode(table);
                    break;

                case 5:
                    System.out.println(table.toString());
                    break;

                default:
                    System.out.println("Invalid option, please try again.\n");
                    break;
            }
        }
    }
}
```

```
    }
}

private static void search(HashTable table) throws IOException
{
    System.out.print("Enter the key of the item to search for: ");
    String key = stdin.readLine();
    System.out.println(key);
    Object item = table.tableRetrieve(key);
    if (item == null)
    {
        System.out.println("Item with key ' " + key + "' not found in HashTable
.");
    }
    else
    {
        System.out.println("Item with key ' " + key + "' found in HashTable:\n"
+ item.toString());
    }
}

private static void insert(HashTable table) throws IOException
{
    System.out.print("Enter the key of the item to insert: ");
    String key = stdin.readLine();
    System.out.println(key);

    System.out.print("Enter the value: ");
    String assocstring = stdin.readLine();
    System.out.println(assocstring);

    Object existingItem = table.tableRetrieve(key);
    if (existingItem != null)
    {
        System.out.println("Item with key ' " + key + "' already exists in Hash
Table and cannot be inserted again.");
    }
    else
    {
        table.tableInsert(key, assocstring);
        System.out.println("Item " + assocstring + " inserted into HashTable w
ith key " + key);
    }
}

private static void delete(HashTable table) throws IOException
{
    System.out.print("Enter the key of the item to delete: ");
    String key = stdin.readLine();
    System.out.println(key);

    Object existingItem = table.tableRetrieve(key);
    if (existingItem != null)
    {
        System.out.println("Item " + "" + key + "" + " deleted.");
        table.tableDelete(key);
    }
    else
    {
        System.out.println("Item does not exist, cannot delete a non-existent
item!");
    }
}
```

```
}  
  
private static void hashCode(HashTable table) throws IOException  
{  
    System.out.print("Enter the key you would like the hashCode for: ");  
    String key = stdin.readLine();  
    System.out.println(key);  
    System.out.println("Hashcode : " + table.hashCode(key));  
}  
}
```

```
:::::::::::  
Lab12P3Output.txt  
:::::::::::  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: ATE  
Enter the value: -7  
Item -7 inserted into HashTable with key ATE  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: EAT  
Enter the value: 5  
Item 5 inserted into HashTable with key EAT  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: TEA  
Enter the value: 1000  
Item 1000 inserted into HashTable with key TEA  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: GRADE  
Enter the value: 100
```

```
Item 100 inserted into HashTable with key GRADE  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: DSA  
Enter the value: 10  
Item 10 inserted into HashTable with key DSA  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: CAT  
Enter the value: 67  
Item 67 inserted into HashTable with key CAT  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: DOGGY  
Enter the value: 100  
Item 100 inserted into HashTable with key DOGGY  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: CHAIRS  
Enter the value: -45  
Item -45 inserted into HashTable with key CHAIRS  
Select from the following menu:  
0. Exit the program  
1. Insert a [key, associated value] pair in the table.  
2. Delete a pair from the table.  
3. Retrieve and display the value associated with a key in the table.  
4. Display the hashCode of a key.  
5. Display the content of the table.
```

```
Make your menu selection now: 1  
Enter the key of the item to insert: EAT  
Enter the value: 76  
Item with key 'EAT' already exists in HashTable and cannot be inserted again.  
Select from the following menu:
```

0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: LETTER
Enter the value: 6
Item 6 inserted into HashTable with key LETTER
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: LETTER
Enter the value: 34
Item with key 'LETTER' already exists in HashTable and cannot be inserted again.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: GRADE
Enter the value: 0
Item with key 'GRADE' already exists in HashTable and cannot be inserted again.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: HASH
Enter the value: 88
Item 88 inserted into HashTable with key HASH
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: EAT
Enter the value: 34
Item with key 'EAT' already exists in HashTable and cannot be inserted again.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: JAVA
Enter the value: 50
Item 50 inserted into HashTable with key JAVA
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: HTML
Enter the value: -10
Item -10 inserted into HashTable with key HTML
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: PYTHON
Enter the value: 25
Item 25 inserted into HashTable with key PYTHON
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: PASCAL
Enter the value: 80
Item 80 inserted into HashTable with key PASCAL
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: BETTA
Enter the value: -15
Item -15 inserted into HashTable with key BETTA
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 1
Enter the key of the item to insert: APPLE
Enter the value: 200
Item 200 inserted into HashTable with key APPLE
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 5
{APPLE, 200} {JAVA, 50} {HASH, 88} {LETTER, 6} {GRADE, 100} {EAT, 5} {BETTA, -15}
{PASCAL, 80} {CAT, 67} {DSA, 10} {TEA, 1000} {ATE, -7} {PYTHON, 25} {HTML, -10} {C
HAIRS, -45} {DOGGY, 100}
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 4
Enter the key you would like the hashCode **for**: HTML
HashCode : 283052
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 4
Enter the key you would like the hashCode **for**: JAVA
HashCode : 329409
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 4
Enter the key you would like the hashCode **for**: EAT
HashCode : 5172
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 4
Enter the key you would like the hashCode **for**: HASH

HashCode : 263784
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: ATE
Item 'ATE' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: EAT
Item 'EAT' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: TEA
Item 'TEA' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: GRADE
Item 'GRADE' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: DSA
Item 'DSA' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: CAT

Item 'CAT' deleted.

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: DOGGY

Item 'DOGGY' deleted.

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: CHAIRS

Item 'CHAIRS' deleted.

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: EAT

Item does not exist, cannot delete a non-existent item!

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: LETTER

Item 'LETTER' deleted.

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: LETTER

Item does not exist, cannot delete a non-existent item!

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: GRADE

Item does not exist, cannot delete a non-existent item!

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: HASH

Item 'HASH' deleted.

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: EAT

Item does not exist, cannot delete a non-existent item!

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: JAVA

Item 'JAVA' deleted.

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

Enter the key of the item to delete: HTML

Item 'HTML' deleted.

Select from the following menu:

0. Exit the program

1. Insert a [key, associated value] pair in the table.

2. Delete a pair from the table.

3. Retrieve and display the value associated with a key in the table.

4. Display the hashCode of a key.

5. Display the content of the table.

Make your menu selection now: 2

```
Enter the key of the item to delete: PYTHON
Item 'PYTHON' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: PASCAL
Item 'PASCAL' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: BETTA
Item 'BETTA' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 2
Enter the key of the item to delete: APPLE
Item 'APPLE' deleted.
Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 5

Select from the following menu:
0. Exit the program
1. Insert a [key, associated value] pair in the table.
2. Delete a pair from the table.
3. Retrieve and display the value associated with a key in the table.
4. Display the hashCode of a key.
5. Display the content of the table.

Make your menu selection now: 0
Exiting program... good bye
```