Hash Table ADT		
Hash = {Key = <K>, Value=<V>}		
{Inv: k ∈ N}		
Create HashTable		HASH
add:	K X V	
remove:	HASH X K	BOOLEAN
search:	HASH X K	V
getSize:	HASH X HASH	INTEGER
isEmpty:		BOOLEAN
hashFunction:	HASH X K	INTEGER

Constructor Operations

HashTable():
Constructs a new empty hash table.

{pre: }
{pos: Creates a new empty hash table.}

Modifying Operations

add(key: K, value: V):
Inserts a key-value pair into the hash table.

{pre: key is a valid key }

{post: the pair is inserted in the hash table. If there was another value associated with this key before, it is replaced by the new value.}

remove(key: K):
Removes the key-value pair associated with the given key from the hash table.

{pre: key is valid}
{pos: Removes the key-value pair associated with the given key from the hash table.}

search(key: K):
Retrieves the value associated with the given key from the hash table.

{pre: key is valid}
{pos: Retrieves and returns the value associated with the given key from the hash table.}

getSize():
Returns the number of key-value pairs in the hash table.

{pre: the hash table must exist}
{pos: Returns the number of key-value pairs in the hash table.}

isEmpty():
Checks if the hash table is empty.

{pre: the hash table must exist}
{pos: Returns true if the hash table is empty, false otherwise.}

Analysing Operations

hashFunction(key: K):
Computes the index in the array for a given key.

{pre: key is a valid key}

{pos: Computes and returns the index in the array for a given key.}

Note: This ADT uses a custom HashNode class for handling linked lists within buckets.

This ADT represents a node used within the hash table for chaining collisions. It contains a key-value pair, as well as references to the next and previous nodes in the linked list.

Hash Node ADT		
Hash Node= {Key = <K>, Value=<V>, Next = <HashNode>, Previous = <HashNode>}		
{Inv: k ∈ N}		
HashNode	K X V	HASH NODE
add	K X V	HASH NODE
incrementSize		INTEGER
incrementSize	HASH NODE X INTEGER	INTEGER
removeLast		HASH NODE
getSize		INTEGER

Constructor Operations

HashNode(key: K, value: V): Constructs a new hash node. {pre: } {pos: Constructs a new hash node with the specified key-value pair.}

Modifying Operations

add(added: HashNode<K, V>): Adds a new node to the end of the linked list.

{pre: }
{pos: Adds the provided node to the end of the linked list.}

incrementSize(): Increments the size of the node by a specified amount.

{pre: }
{pos: Increments the size of the node by 1.}

incrementSize(size: int): Increments the size of the node by a specified amount.

{pre: size is a valid integer}
{pos: Increments the size of the node by the specified amount.}

removeLast(): Removes the last node in the linked list.

{pre: }
{pos: Removes the last node in the linked list.}

getSize(): Retrieves the size of the node.

{pre: }
{pos: Retrieves and returns the size of the node.}