Psuedocode:   
  
Define DEFAULT\_SIZE as 179

Define structure Bid: string

bidId/ title / fund

amount

Constructor:

Initialize amount to 0.0

Define class HashTable:

Define nested structure Node:

bid

key (unsigned int)

next (Node pointer)

Constructor:

Initialize key to UINT\_MAX

Initialize next to nullptr

Constructor with Bid:

Initialize bid

Constructor with Bid and key:

Initialize bid with given Bid/key

Public members:

Constructor:

Initialize nodes with tableSize

Constructor with size:

Initialize tableSize with given size

Resize nodes with given size

Destructor:

Erase nodes

Insert(bid):

Calculate key using hash function

If no entry is found for the key:

Assign new Node to the key position

Else if node is not used:

Assign old node key to UINT\_MAX

Set key and bid for old node

Set old node next to nullptr

Else:

Find the next open node

Add new Node to the end

PrintAll():

Remove(bidId):

Calculate key using hash function

If next node is not nullptr:

If bidId matches:

Point tempNode to next node

Copy and delete +1 node

Else:

Find the node with matching bidId

If found:

Delete node

Else:

Print "key not found"

Else:

If bidId matches:

Assign new Node to the current position

Else:

Print "No Bid exists"

Search(bidId):

Calculate key using hash function

Return bid

Move to next node

Return empty bid  
  
Reflection:  
The Hash table effectively implements a hash table with chaining to handle collisions. Ensuring that the table can handle multiple entire with the same value. The Nodes provides a flexible creation of nodes and managed hash table. With the hash function it uses the Zybook modulo to calculate the hash value. Implementing the insertion logic can ensure that new nodes are added to the end of the chain, while maintaining the integrity of the hash table. Printing all method handles both the top and bottom chain nodes, ensuring that all entries are given. Looking through the Removal nodes just handles the separation of nodes. Search method of the code efficiently locates bids by traversing the hash table and chains. Returning if the bid is found or if not found, remain empty. Overall, code demonstrates a simpler understanding of the code and implementation. Effectively handing insertion, removal, and then searching of bids. Continuing the management of the hash table integrity,