

Evidence 1

```
class BookRec:
    def __init__(self, BookID, Title, Pointer):
        self.__BookID = BookID
        self.__Title = Title
        self.__Pointer = Pointer
    def GetBookID(self):
        return self.__BookID
    def GetTitle(self):
        return self.__Title
    def GetPointer(self):
        return self.__Pointer
    def SetBookID(self, BookID):
        self.__BookID = BookID
    def SetTitle(self, Title):
        self.__Title = Title
    def SetPointer(self, Pointer):
        self.__Pointer = Pointer
```

Evidence 2

```
class LinkedList:                                # dynamic linked list
    def __init__(self):
        self.__Start = None                    # null

    def IsEmpty(self):
        return (self.__Start == None)

    def DisplayLinkedList(self):
        curr = self.__Start                    # current node
        if curr == None:
            #empty linked list
            print("Empty linked list")
        else:
            print()
            print("-" * 52)
            print("| {0:^15} | {1:^30} |".format("BookID", "Title"))
            print("-" * 52)
            while curr != None:
                print("| {0:^15} | {1:^30} |".format(curr.GetBookID(),
curr.GetTitle()))
                curr = curr.GetPointer()
            print("-" * 52)

    def AddNode(self, BookID, Title):
        NewNode = BookRec(BookID, Title, None)
        if self.IsEmpty():
            self.__Start = NewNode
        else:
            curr = self.__Start
            while curr.GetPointer() != None:
                curr = curr.GetPointer()
            curr.SetPointer(NewNode)
```

```

def SearchNode(self, BookID):
    curr = self.__Start
    while curr != None and curr.GetBookID() != BookID:
        curr = curr.GetPointer()
    if curr == None:
        return False           # is within linked list
    else:
        return True            # is not in linked list
    # return (curr == None)

def DeleteNode(self, BookID):
    if self.SearchNode(BookID):
        prev = self.__Start
        curr = self.__Start
        while curr != None and curr.GetBookID() != BookID:
            prev = curr
            curr = curr.GetPointer()
        prev.SetPointer(curr.GetPointer())

```

Evidence 3

```

class HashTable:
    def __init__(self, Size = 17):           # Initialise()
        self.__Size = Size
        self.__Slots = [None] + [LinkedList() for i in range(Size)]
        # array of empty linked lists.
        # self.__Slots[0] = None as hash table is 1-based.

    def Hash(self, BookID):
        ASCIIsum = 0
        for character in BookID:
            ASCIIsum += ord(character)        # add ASCII value to ASCII sum
        Address = (ASCIIsum % self.__Size) + 1 # address = remainder + 1
        return Address

    def Display(self):
        for i in range(self.__Size):
            Address = i + 1
            CurrentLinkedList = self.__Slots[Address]
            print("[{0}] in hash table: ".format(Address), end='')
            CurrentLinkedList.DisplayLinkedList()
            print()

    def Put(self, BookID, Title):
        Address = self.Hash(BookID)
        self.__Slots[Address].AddNode(BookID, Title)

    def Remove(self, BookID):
        Address = self.Hash(BookID)
        CurrentLinkedList = self.__Slots[Address]
        if CurrentLinkedList.SearchNode(BookID):
            CurrentLinkedList.DeleteNode(BookID)
            print("{0} has been removed from the hash table.".format(BookID))
        else:
            print("{0} cannot be removed as it is not found in the hash
table.".format(BookID))

```

```

def Search(self, BookID):
    Address = self.Hash(BookID)
    CurrentLinkedList = self.__Slots[Address]
    if CurrentLinkedList.SearchNode(BookID):
        return True
    else:
        return False
    # alternatively: return CurrentLinkedList.SearchNode(BookID)

```

Evidence 4

Screenshots

```

RESTART: C:\Users\USER\Google Drive\codes\2019 Programming Exercise\PC8 Hashtable Chaining - Hash Table\pc8.py
[1] in hash table: Empty linked list

[2] in hash table:
+-----+-----+
| BookID | Title |
+-----+-----+
| CS733  | Basic algorithms |
+-----+-----+

[3] in hash table: Empty linked list

[4] in hash table:
+-----+-----+
| BookID | Title |
+-----+-----+
| AB944  | Master Computing |
| BK232  | Programming exercises |
+-----+-----+

[5] in hash table: Empty linked list
[6] in hash table: Empty linked list
[7] in hash table: Empty linked list
[8] in hash table: Empty linked list
[9] in hash table: Empty linked list
[10] in hash table: Empty linked list
[11] in hash table:
+-----+-----+
| BookID | Title |
+-----+-----+
| KS293  | Data structures |
+-----+-----+

[12] in hash table: Empty linked list
[13] in hash table:
+-----+-----+
| BookID | Title |
+-----+-----+
| PK199  | Testing Python |
+-----+-----+

[14] in hash table: Empty linked list
[15] in hash table: Empty linked list
[16] in hash table: Empty linked list
[17] in hash table: Empty linked list

```

AB944 has been removed from the hash table.

[1] in hash table: Empty linked list

[2] in hash table:

BookID	Title
CS733	Basic algorithms

[3] in hash table: Empty linked list

[4] in hash table:

BookID	Title
AB944	Master Computing
BK232	Programming exercises

[5] in hash table: Empty linked list

[6] in hash table: Empty linked list

[7] in hash table: Empty linked list

[8] in hash table: Empty linked list

[9] in hash table: Empty linked list

[10] in hash table: Empty linked list

[11] in hash table:

BookID	Title
KS293	Data structures

[12] in hash table: Empty linked list

[13] in hash table:

BookID	Title
PK199	Testing Python

[14] in hash table: Empty linked list

[15] in hash table: Empty linked list

[16] in hash table: Empty linked list

[17] in hash table: Empty linked list

True

Program code

```
Books = HashTable(17)
```

```
Books.Put("CS733", "Basic algorithms")
```

```
Books.Put("AB944", "Master Computing")
Books.Put("KS293", "Data structures")
Books.Put("BK232", "Programming exercises")
Books.Put("PK199", "Testing Python")
Books.Display()
Books.Remove("AB944")
Books.Display()
print(Books.Search("PK199"))
```