

CLASS ITEM:

FUNCTION INITIALISE

```
Self.name = name  
Self.price = price  
Self.store = store(list)
```

FUNCTION getName(self)

```
Return self.name
```

FUNCTION getPrice(self)

```
Return self.price
```

FUNCTION getStore(self)

```
Return self.store
```

FUNCTION TOSTRING:

```
Printout every data
```

Class ShoppingList

FUNCTION INITIALISE

```
Self.house_num = house_num  
Self.item_list = []  
Self.optimise_item)list = UnorderedList()  
Self.next_delivery = False
```

FUNCTION TOSTRING

```
Print out every data
```

Class ItemQuantity:

FUNCTION INITIALISE

```
Self.item = item(class)  
Self.quantity = quantity
```

FUNCTION getItem(self):

```
Return self.item
```

FUNCTION setItem(self,item):

Self.item = item

FUNCTION TO STRING

print out every data

Class Node:

FUNCTION INITIALISE

Self.data = init_data(class itemQuantity)

Self.next = None

FUNCTION get_date(self)

Return self.data

FUNCTION get_next(self)

Return self.next

FUNCTION get_store(self):

Return self.data.item.store

FUNCTION get_name(self):

Return self.data.name

FUNCTION set_data(self,new_data):

Self.data = new_data

Class UnorderedList:

FUNCTION INITIALISE

Self.head = none

FUNCTION is_empty(self)

Return self.head == None

FUNCTION add(self,item)

Temp = Node(item)

Temp.set_next(self.head)

Self.head = temp

FUNCTION size(self)

```

Current = self.head

Count = 0

WHILE CURRENT NOT EQUAL None
    Count += 1
    Current = current.get_next()

Return count

FUNCTION search(self,item)
    Current = self.head
    Found = False
    WHILE CURRENT NOT EQUAL None AND found NOT EQUAL TRUE
        IF CURRENT NAME EQUAL ITEM
            Found = true
            Return current
        Else:
            Current = current.get_next()

    Return None

FUNCTION getPreNode(self,item)
    Current = self.head
    Previous = none
    Found = false
    WHILE CURRENT NOT EQUAL None AND found NOT EQUAL TRUE
        IF CURRENT NAME EQUAL ITEM
            Found = true
            Return previous
        Else:
            Previous = current
            Current = current.get_next()

    Return none

```

FUNCTION reverse(self)

Current = self.head

Following = self.head

Previous = None

WHILE CURRENT NOT EQUAL NONE

Following = following.get_next()

Current.set_next(previous)

Previous = current

Current = following

Self.head = previous

FUNCTION remove(self,item)

Current = self.head

Previous = None

Found = False

WHILE CURRENT NOT EQUAL None and FOUND NOT EQUAL TRUE

IF CURRENT NAME EQUAL ITEM

Found = True

Else

Previous = current

Current = current.get_next()

IF previous == None:

Self.head = current.get_next()

ELSE

Previous.set_next(current.get_next())