Liman Chang, Kenneth Christian
Test Design Document

The following test cases are cases used by the test bed main in the ShoppingBag class. The Shopping class was tested using the sample input provided in the project1 description.

Test Case #	Purpose of the test case	Input Data	Expected output
1	Test the add() method by adding GroceryItems to the ShoppingBag. The GroceryItem input is always formatted properly in the Shopping class. This also tests the print() method to display the added item inside the ShoppingBag.	"I", 5.987, false	I: $5.99 : tax free
2	Test the emptyBag() method which nulls all objects in the array and resets size to zero. After adding items to the bag, if print() returns nothing, then the GroceryItem has been removed.	"I", 5.987, false	
3	Test the grow() method, which double the total array size every time the max array size is reached. ● Case 1: Put 3 objects into the shopping bag to grow the size from 2 to 4 ● Case 2: Put more than 4 objects to grow the array to size 8	● Case 1 Instance #1: "I", 5.987, false Instance #2: "I", 5.987, false Instance #3: "I", 5.987, false ● Case 2 Instance #1: "I", 5.987, false Instance #2: "I", 5.987, false Instance #3: "I", 5.987, false Instance #4: "I", 5.987, false Instance #5: "I", 5.987, false Instance #6: "I", 5.987, false	● Case 1 returns I: $5.99 : tax free I: $5.99 : tax free I: $5.99 : tax free ● Case 2 returns I: $5.99 : tax free I: $5.99 : tax free I: $5.99 : tax free I: $5.99 : tax free I: $5.99 : tax free I: $5.99 : tax free

4	Test the find() and remove() methods. The find() method is automatically tested as the remove() method call find() everytime. The method returns false if the item is not found in the array. ● Case 1: Remove an item that is not inside the array ● Case 2: Remove an item that is inside the array ● Case 3: Remove an item with duplicates in the array	● Case 1 Instance #1: “I”, 5.987, false Instance #2: “like”, 4.321, false ● Case 2 Instance #1: “I”, 5.987, false ● Case 3 Instance #1: “I”, 5.987, false Instance #2: “I”, 5.987, false Instance #3: “I”, 5.987, false	● Case 1 returns false ● Case 2 returns true ● Case 3 returns true
5	Test the formatting of the GroceryItems by having inputs with irregular price formatting.	Instance #1: “I”, 5.987, false Instance #2: “like”, 4.321, false Instance #3: “sushi_and_burger”, 1000, false Instance #4: “hello”, 40, true	I: $5.99 : tax free like: $4.32 : tax free sushi_and_burger: $1,000.00 : tax free hello: $40.00 : is taxable
6	Test the salesPrice() method which sums the price of all objects inside the ShoppingBag. The method does not format the price.	Instance #1: “I”, 5.987, false Instance #2: “like”, 4.321, false	10.308
7	Test the salesTax() method which sums the total tax amount of the GroceryItems in the bag. The method does not format the tax.	Instance #1: “hello”, 40, true Instance #2: “world”, 60, true	6.625
8	Test the getSize() method which returns the number of items in the shopping bag. ● Case 1: Put in one item to test the counter ● Case 2: Put more items then the initial max size of the ShoppingBag to show that grow() does not affect size	● Case 1 Instance #1: “I”, 5.987, false ● Case 2 Instance #1: “I”, 5.987, false Instance #2: “I”, 5.987, false Instance #3:	● Case 1 returns 1 ● Case 2 returns 3

		“I”, 5.987, false	
9	Test the isEmpty() method which determines if the size of the ShoppingBag is 0 or not. ● Case 1: Put an object into the array to have it not be empty ● Case 2: Call the emptyBag() method to clear all items in the bag	● Case 1 Instance #1: “I”, 5.987, false ● Case 2	● Case 1 returns false ● Case 2 returns true