Whitebox Test: Validate()

Function: Validate()	Data	Expected Result	Description

Test08a	size = 0.25
	weight = 200
	struct point destination = {2,2}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see if valid shipment input with valid shipment size, SMALL_BOX_VOL, returns 1(all inputs are validated)

Test08b	size = 0.50
	weight = 20
	struct point destination = {18,8}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see if valid shipment input with valid shipment size, MED_BOX_VOL, returns 1(all inputs are validated)

Test08c	size = 1
	weight = 20
	struct point destination = {23,11}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see if valid shipment input with valid shipment size, LARGE_BOX_VOL, returns 1(all inputs are validated)

Test09a	size = 0.40
	weight = 1
	struct point destination = {22,22}
		Invalid size	Test to see if invalid shipment size between SMALL_BOX_VOL and MED_BOX_VOL displays error message

Test09b	size = 0.9999999
	weight = 90
	struct point destination = {22,22}
		Invalid size	Test to see if invalid shipment size 0.9999999(closest to valid size LARGE_BOX_VOL) displays error message

Test09c	size = 1.000000001
	weight = 75
	struct point destination = {1,21}
		Invalid size	Test to see if invalid shipment size 1.000000001(closest to valid size LARGE_BOX_VOL) displays error message

Test09d	size = 0.251
	weight = 30
	struct point destination = {1,21}
		Invalid size	Test to see if invalid shipment size 0.251(closest to valid size SMALL_BOX_VOL) displays error message

Test09e	size = 0.500000001
	weight = 90
	struct point destination = {22,22}
		Invalid size	Test to see if invalid shipment size 0.500000001(closest to valid size MED_BOX_VOL) displays error message

Test09f	size = -0.25
	weight = 30
	struct point destination = {1,21}
		Invalid size	Test to see if negative SMALL_BOX_VOL size displays error message

Test09g	size = -50
	weight = 40
	struct point destination = {1,21}
		Invalid size	Test to see if negative MED_BOX_VOL size displays error message

Test10a	size = 0.5
	weight = 999
	struct point destination = {1,7}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see valid shipment weight is validated

Test10b	size = 0.25
	weight = 2
	struct point destination = {1,7}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see valid shipment weight is validated

Test10c	size = 0.50
	weight = 1000.1
	struct point destination = {1,7}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see valid shipment weight (1000.1) is validated

Test11a	size = 1
	weight = -1000
	struct point destination = {1,7}
		Invalid weight (must be 1-1000 Kg.)	Test to see if invalid weight (negative max weight) displays the error message

Test11b	size = 0.25
	weight = 0.9
	struct point destination = {1,21}
		Invalid weight (must be 1-1000 Kg.)	Test to see if invalid weight input (0.9) displays the error message

Test11c	size = 0.25
	weight = ABXZ
	struct point destination = {1,21}
		Invalid weight (must be 1-1000 Kg.)	Test to see if invalid weight input (not numbers) displays the error message

Test11d	size = 1
	weight = -500
	struct point destination = {1,7}
		Invalid weight (must be 1-1000 Kg.)	Test to see if invalid weight input (-500) displays the error message

Test11e	size = 0.25
	weight = 11111111111
	struct point destination = {1,7}
		Invalid weight (must be 1-1000 Kg.)	Test to see if invalid weight input displays the error message

Test12a	size = 0.25
	weight = 30
	struct point destination = {1,4}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see if valid destination input(1st point inside building) returns 1

Test12b	size = 1
	weight = 500
	struct point destination = {24,14}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see if valid destination input(last point inside building) returns 1

Test12c	size = 1
	weight = 90
	struct point destination = {2,10}
		Return 1(all user inputs are validated and valid shipment is sent to the AssignTrucks function)	Test to see if valid destination input(point inside building) returns 1

Test13a	size = 0.50
	weight = 200
	struct point destination = {1,3}
		Invalid destination	Test to see if invalid destination input (not inside building) displays error message

Test13b	size = 1
	weight = 300
	struct point destination = {25,25}
		Invalid destination	Test to see if invalid destination input (max point but not inside building) displays error message

Test13c	size = 1
	weight = 59
	struct point destination = {-14, 20}
		Invalid destination	Test to see if invalid destination column (point inside building but negative) displays error message

Test13d	size = 1
	weight = 99
	struct point destination = {B, 21}
		Invalid destination	Test to see if invalid destination column (not number) displays error message

Test13e	size = 1
	weight = 99
	struct point destination = {14, -20}
		Invalid destination	Test to see if invalid destination row (point inside building but negative) displays error message

Test13f	size = 1
	weight = 99
	struct point destination = {1, @}
		Invalid destination	Test to see if invalid destination row (not number) displays error message

Test13g	size = 1
	weight = 99
	struct point destination = {13,10 }
		Invalid destination	Test to see if invalid destination input (valid point but not inside building) displays error message

Test14	size = 0
	weight = 0
	struct point destination = {0, x}
		Thank you for shipping with Seneca!	Test the exit code (0 0 x). The exit input must display the message (expected result) and terminate the program.