CS-589

Software Testing and Analysis

PROJECT REPORT

CHETHAN KUMAR APPAJI
[A20355296]
cappaji@hawk.iit.edu

1. Model-based testing of the VendingMachine class (2-transition sequence testing)

State	Incoming Transactions	Outgoing Transactions	Transition Pairs
Idle	T1, T2, T3, T15, T14, T13, T12, T11, T10, T6, T8, T9, T4	T2, T3, T7, T6, T4, T5	(T1,T2) (T1,T3) (T1,T7) (T1,T6) (T1,T4) (T1,T5) (T2,T2) (T2,T3) (T2,T7) (T2,T6) (T2,T4) (T2,T5) (T3,T2) (T3,T3) (T3,T7) (T3,T6) (T3,T4) (T3,T5) (T15,T2) (T15,T3) (T15,T7) (T15,T6) (T15,T4) (T15,T5) (T14,T2) (T14,T3) (T14,T7) (T14,T6) (T14,T4) (T14,T5) (T13,T2) (T13,T3) (T13,T7) (T13,T6) (T13,T4) (T13,T5) (T12,T2) (T12,T3) (T12,T7) (T12,T6) (T12,T4) (T12,T5) (T11,T2) (T11,T3) (T11,T7) (T11,T6) (T11,T4) (T11,T5) (T10,T2) (T10,T3) (T10,T7) (T10,T6) (T10,T4) (T10,T5) (T6,T2) (T6,T3) (T6,T7) (T6,T6) (T6,T4) (T6,T5) (T8,T2) (T8,T3) (T8,T7) (T8,T6) (T8,T4) (T8,T5) (T9,T2) (T9,T3) (T9,T7) (T9,T6) (T9,T4) (T9,T5) (T4,T2) (T4,T3) (T4,T7) (T4,T6) (T4,T4) (T4,T5)
			Non Executable Pairs: (T1,T7) (T1,T6) (T10, T6) (T15,T6) (T14,T6) (T13,T6) (T12, T6) (T11,T6) (T8,T6) (T9,T6) After Execution of tea(), Control never enters coin(t+25<price) Since t+25 will be always greater than price unless setting the new price value
Coins Inserted	T7, T20, T21, T23, T19	T12, T11, T10, T19, T24, T25, T22, T21, T20	(T7,T12) (T7,T11) (T7,T10) (T7,T19) (T7,T24) (T7,T25) (T7,T22) (T7,T21) (T7,T20) (T20,T12) (T20,T11) (T20,T10) (T20,T19) (T20,T24) (T20,T25) (T20,T22) (T20,T21) (T20,T20) (T21,T12) (T21,T11) (T21,T10) (T21,T19) (T21,T24) (T21,T25) (T21,T22) (T21,T21) (T21,T20) (T23,T12) (T23,T11) (T23,T10) (T23,T19) (T23,T24) (T23,T25) (T23,T22) (T23,T21) (T23,T20)

			(T19,T12) (T19,T11) (T19,T10) (T19,T19) (T19,T24) (T19,T25) (T19,T22) (T19,T21) (T19,T20)
			Non Executable Pairs: (T7,T21), (T7,T11), (T21,T12), (T19,T11), (T7,T24), (T7,T25), (T19,T25), (T21,T24) All the above Pairs are Non-Executable since x==0
Sugar	T22, T16, T17, T18	T13, T14, T15, T16, T17, T18, T23, T26, T27	(T22, T13) (T22,T14) (T22,T15) (T22,T16) (T22, T17) (T22, T18) (T22,T23) (T22,T26) (T22,T27) (T16, T13) (T16,T14) (T16,T15) (T16,T16) (T16, T17) (T16, T18) (T16,T23) (T16,T26) (T16,T27) (T17, T13) (T17,T14) (T17,T15) (T17,T16) (T17, T17) (T17, T18) (T17,T23) (T17,T26) (T17,T27) (T18, T13) (T18,T14) (T18,T15) (T18,T16) (T18, T17) (T18, T18) (T18,T23) (T18,T26) (T18,T27)
			Non Executable Pairs: (T17,T13), (T18,T15) (T18,T27) (T17,T26) When s==1, Smallcup would not be executed and when s==2, largecup would not be executed
NoSmallCups	T25, T27, T28	T9, T28	(T25,T9) (T25,T28) (T27,T9) (T27,T28) (T28,T9) (T28,T28)
NoLargeCups	T24, T26, T29	T8, T29	(T24,T8) (T24,T29) (T26,T8) (T26,T29) (T29,T8) (T29,T29)

TestCase	Transition Traversed	New Transition Pairs Covered
Test#1	T1, T2, T3, T4, T7, T21, T22, T15, T5	(T1,T2) (T2,T3) (T3,T4) (T4,T7) (T7,T21) (T21,T22) (T22,T15) (T15,T5)
Test#2	T1, T3, T2, T4, T6, T7, T19, T22, T13, T5	(T1,T3) (T3,T2) (T2,T4) (T4,T6) (T6,T7) (T7,T19) (T19,T22) (T22,T13) (T13,T5)
Test#3	T1, T4, T2, T6, T6, T3, T7, T20, T21, T19, T20, T12, T5	(T1,T4) (T4,T2) (T2,T6) (T6,T6) (T6,T3) (T3,T7) (T7,T20) (T20,T21) (T21,T19) (T19,T20) (T20,T12) (T12,T5)
Test#4	T1, T5	(T1,T5)
Test#5	T1, T4, T2, T2, T7, T22, T16, T18, T13, T7, T19, T22, T23, T12, T7, T10, T5	(T2,T2) (T2,T7) (T7,T22) (T22,T16) (T16,T18) (T18, T13) (T13,T7) (T23,T12) (T12,T7) (T7,T10) (T10,T5)

Test#6	T1, T4, T4, T3, T3 T6, T3, T5	(T4,T4) (T4,T3) (T3,T3) (T3,T6) (T3,T5)
Test#7	T1, T4, T6, T2, T6, T4, T5	(T6,T2) (T6,T4) (T4,T5)
Test#8	T1, T4, T7, T10, T2, T7, T10, T3, T7, T10,T7, T10, T4, T6, T5	(T10,T2) (T10,T3) (T10,T7) (T6,T5)
Test#9	T1, T4, T7, T22, T14, T2, T7, T22, T17, T14, T3, T5	(T22,T14) (T14,T2) (T22,T17) (T17, T14) (T14, T3)
Test#10	T1,T4, T3, T7, T22, T18, T14, T7, T22, T14, T4, T7, T22, T14, T5	(T22,T18) (T18,T14) (T14,T7) (T14,T14) (T14,T5)
Test#11	T1, T4, T3, T7, T21, T20, T21, T11, T3, T7, T21, T21, T11, T2, T5	(T21,T20) (T21,T11) (T11,T3) (T21,T21) (T11,T2)
Test#12	T1, T4, T2, T7, T21, T11, T4, T7, T21, T11, T7, T21, T10, T7, T21, T11, T5	(T11,T4) (T11,T7) (T21,T10) (T11,T5)
Test#13	T1, T4, T2, T7, T19, T21, T19, T12, T3, T5	(T19,T21) (T19,T12) (T12,T3)
Test#14	T1, T4, T3, T7, T19, T19, T12, T4, T7, T19, T12, T2, T5	(T19,T19) (T12,T4) (T12,T2) (T2,T5)
Test#15	T1, T4, T2, T7, T22, T18, T16, T17, T18, T13, T3, T5	(T18,T16) (T16,T17) (T17,T18) (T13,T3)
Test#16	T1, T4, T2, T7, T22, T16, T16, T23, T19, T22, T16, T13, T2, T5	(T16,T16) (T16,T23) (T23,T19) (T16,T13) (T13,T2)
Test#17	T1, T4, T2, T7, T20, T19, T20 T20, T19, T20, T20, T22, T17, T17, T23, T22, T18, T18, T23, T22, T13, T4, T5	(T20,T19) (T20,T20) (T20, T22) (T17, T17) (T17, T23) (T23,T22) (T18,T18) (T18,T23) (T13,T4)
Test#18	T1, T4, T3, T7, T21, T22, T15, T2, T7, T22, T17, T15, T3, T5	(T15,T2) (T17,T15) (T15,T3)
Test19	T7, T22, T17, T16, T17, T15, T7, T21, T22, T15, T4	(T17,T16) (T15,T7) (T15,T4)
Test#20	T1,T4, T2, T7, T19, T24, T29, T29, T8, T2, T5	(T19,T24) (T24,T29) (T29,T29) (T29,T8) (T8,T2)
Test#21	T1, T4, T2, T7, T19, T24, T8, T3, T7, T19, T24, T8, T5	(T24,T8) (T8, T3) (T8,T5)
Test#22	T1, T4, T2, T7, T19, T20, T24, T8, T4	(T8, T4) (T20, T24)
Test#23	T1, T4, T2, T7, T19, T22, T23, T24, T8, T7, T10, T5	(T8,T7) (T23,T24)
Test#24	T1, T4, T7, T20, T10,T2, T7, T20, T11, T5	(T20,T10) (T20, T11)
Test#25	T1, T4, T2, T7, T19, T22, T26, T29, T8, T7, T22, T18, T26, T8, T5	(T22,T26) (T26,T29) (T18,T26) (T26,T8)

Test#26	T1, T4, T2, T7, T19, T22, T16, T26, T8, T7, T22, T16, T4, T5	(T16,T26) (T16,T14)
Test#27	T1, T4, T3, T7, T21, T25, T28, T28, T9, T7, T21, T20, T25, T9, T5	(T21,T25) (T25,T28) (T28,T28) (T28,T9) (T20,T25) (T25,T9) (T9,T5)
Test#28	T1, T4, T3, T7, T22, T17, T23, T25, T9, T2, T5	(T23,25) (T9,T2)
Test#29	T1, T4, T3, T7, T22, T23, T21, T22, T23, T20, T25, T9, T3, T7, T22, T23, T10, T5	(T23,T21) (T23,T20) (T9,T3) (T23,T10)
Test#30	T1, T4, T3, T7, T22, T18, T17, T27, T9, T4, T5	(T18,T17) (T17,T27) (T27,T9) (T9,T4)
Test#31	T1, T4, T3, T7, T22, T17, T16, T27, T28, T9, T7, T22, T23, T11, T7, T19, T10, T5	(T16,T27) (T27,T28) (T23,T11) (T19,T10)
Test#32	T1, T4, T3, T7, T21, T22, T27, T9, T5	(T22, T27)

2. Testing default (ghost) transitions of the GasPump class

TestCase	Default Transition	State

Test#33	coin()[price==0]
	small_cup()
	large_cup()
	sugar()
	tea()
	insert_large_cups(int n) [n>=0]
	insert_small_cups(int n) [n>=0]
	set_price(int p) [p<=0]
	cancel()
		IDLE

Test#34,
Test#35,
Test#36
	tea() [k==0 || s==0 || k1==0]
	insert_large_cups(int n)
	insert_small_cups(int n)
	set_price(int p)
	dispose()
		COINSINSERTED

Test#37	tea() [k==0 || s==0 || k1==0]
	insert_large_cups(int n)
	insert_small_cups(int n)
	set_price(int p)
	dispose()
		SUGAR

Test#38	small_cup()
	large_cup()
	sugar()
	tea()
	insert_large_cups(int n) [n>=0]
	insert_small_cups(int n)
	set_price(int p)
	dispose()
	cancel()
		NOLARGECUPS

Test#39	small_cup()
	large_cup()
	sugar()
	tea()
	insert_large_cups(int n)
	insert_small_cups(int n) [n>=0]
	set_price(int p)
	dispose()
	cancel()
		NOSMALLCUPS

3. Multiple-condition testing

Operation: coin()

Condition: (x==1)

x==1	TestCase
T	Test#1
F	Test#3

Operation: coin()

Condition: ((t + 25 >= price) && (price > 0))

t + 25 >= price	price > 0	TestCase
T	T	Test#1
T	F	Test#33
F	T	Test#3
F	F	Not Possible

Operation: coin()

Condition: (t + 25 < price)

t + 25 < price	TestCase
T	Test#3
F	Test#33

Operation: coin()

Condition: ((x > 1) && (x < 6))

x > 1	x < 6	TestCase
T	T	Test#17
T	F	Not Possible
F	T	Not Possible
F	F	Not Possible

Operation: small_cup()

Condition: ((x == 2) || (x == 3))

x == 2	**x == 3**	**TestCase**
T	T	Not Possible
T	F	Test#1
F	T	Test#9
F	F	Test#33

Operation: large_cup()

Condition: ((x == 2) || (x == 3))

x == 2	**x == 3**	**TestCase**
T	T	Not Possible
T	F	Test#2
F	T	Test#10
F	F	Test#33

Operation: sugar()

Condition: ((x == 2) || (x == 3))

x == 2	**x == 3**	**TestCase**
T	T	Not Possible
T	F	Test#32
F	T	Test#17
F	F	Test#33

Operation: sugar()

Condition: ((x == 2)

x == 2	**TestCase**
T	Test#32
F	Test#17

Operation: tea()

Condition: ((x == 2) | | (x == 3))

x == 2	**x == 3**	**TestCase**
T	T	Not Possible
T	F	Test#11
F	T	Test#1
F	F	Test#33

Operation: tea()

Condition: ((x == 2) && (k1 > 1) && (s == 2))

x == 2	**k1 > 1**	**s == 2**	**TestCase**
T	T	T	Test#11
T	T	F	Test#14
T	F	T	Test#35
T	F	F	Test#13
F	T	T	Test#18
F	T	F	Test#2
F	F	T	Test#30
F	F	F	Test#26

Operation: tea()

Condition: ((x == 2) && (k > 1) && (s == 1))

x == 2	**k > 1**	**s == 1**	**TestCase**
T	T	T	Test#3
T	T	F	Test#12
T	F	T	Test#14
T	F	F	Test#35
F	T	T	Test#2
F	T	F	Test#1
F	F	T	Test#25
F	F	F	Test#18

Operation: tea()

Condition: ((x == 2) && (k == 1) && (s == 1))

x == 2	**k == 1**	**s == 1**	**TestCase**
T	T	T	Test#20
T	T	F	Test#35
T	F	T	Not Possible
T	F	F	Test#27
F	T	T	Test#25
F	T	F	Test#30
F	F	T	Test#5
F	F	F	Test#18

Operation: tea()

Condition: ((x == 2) && (k1 == 1) && (s == 2))

x == 2	**K1 == 1**	**s == 2**	**TestCase**
T	T	T	Test#27
T	T	F	Test#36
T	F	T	Test#35
T	F	F	Test#23
F	T	T	Test#30
F	T	F	Test#40
F	F	T	Test#18
F	F	F	Test#5

Operation: tea()

Condition: ((x == 3) && (k1 == 1) && (s == 2))

x == 3	**K1 == 1**	**s == 2**	**TestCase**
T	T	T	Test#30
T	T	F	Test#40
T	F	T	Test#18
T	F	F	Test#5
F	T	T	Not Possible

F	T	F	Test#36
F	F	T	Test#41
F	F	F	Test#23

Operation: tea()

Condition: ((x == 3) && (k == 1) && (s == 1))

x == 3	**K == 1**	**s == 1**	**TestCase**
T	T	T	Test#25
T	T	F	Test#42
T	F	T	Test#5
T	F	F	Test#18
F	T	T	Not Possible
F	T	F	Test#41
F	F	T	Test#36
F	F	F	Test#43

Operation: tea()

Condition: ((x == 3) && (k1 > 1) && (s == 2))

x == 3	**K > 1**	**s == 2**	**TestCase**
T	T	T	Test#18
T	T	F	Test#45
T	F	T	Test#42
T	F	F	Test#5
F	T	T	Not Possible
F	T	F	Test#44
F	F	T	Test#41
F	F	F	Test#36

Operation: tea()

Condition: ((x == 3) && (k > 1) && (s == 1))

x == 3	**K > 1**	**s == 1**	**TestCase**
T	T	T	Test#5
T	T	F	Test#45
T	F	T	Test#40
T	F	F	Test#41
F	T	T	Not Possible
F	T	F	Test#45
F	F	T	Test#36
F	F	F	Test#41

Operation: insert_small_cups(int n)

Condition: ((x == 1) && (n > 0))

x == 1	**n > 0**	**TestCase**
T	T	Test#1
T	F	Test#33
F	T	Test#30
F	F	Test#39

Operation: insert_small_cups(int n)

Condition: ((x == 4) && (n > 0))

x == 4	**n > 0**	**TestCase**
T	T	Test#30
T	F	Test#39
F	T	Test#38
F	F	Test#37

Operation: insert_large_cups(int n)

Condition: ((x == 1) && (n > 0))

x == 1	n > 0	TestCase
T	T	Test#1
T	F	Test#33
F	T	Test#21
F	F	Test#38

Operation: insert_large_cups(int n)

Condition: ((x == 5) && (n > 0))

x == 5	n > 0	TestCase
T	T	Test#21
T	F	Test#38
F	T	Test#39
F	F	Test#37

Operation: set_price(int p)

Condition: ((x == 1) && (p > 0))

x == 1	p > 0	TestCase
T	T	Test#1
T	F	Test#33
F	T	Test#34
F	F	Test#38

Operation: cancel()

Condition: ((x == 2) || (x == 3))

x == 2	x == 3	TestCase
T	T	Not Possible
T	F	Test#34
F	T	Test#37
F	F	Test#38

Operation: dispose()

Condition: ((x == 1)

x == 1	**TestCase**
T	Test#4
F	Test#34

TS.txt TestCases

Test#1: insert_large_cups 5 insert_small_cups 5 set_price 10 coin small_cup sugar tea dispose

Test#2: insert_small_cups 5 insert_large_cups 5 set_price 30 coin coin large_cup sugar tea dispose

Test#3: set_price 60 insert_large_cups 5 coin coin insert_small_cups 5 coin coin small_cup large_cup coin tea dispose

Test#4: dispose

Test#5: set_price 20 insert_large_cups 2 insert_large_cups 2 coin sugar coin large_cup tea coin large_cup sugar sugar tea coin cancel dispose

Test#6: set_price 26 set_price 4 insert_small_cups 2 insert_small_cups 2 coin insert_small_cups 2 dispose

Test#7: set_price 56 coin insert_large_cups 1 coin set_price 30 dispose

Test#8: set_price 15 coin cancel insert_large_cups 1 coin cancel insert_small_cups 1 coin cancel coin cancel set_price 5 coin dispose

Test#9: set_price 15 coin cancel insert_large_cups 1 coin sugar small_cup cancel insert_small_cups 1 dispose

Test#10: set_price 15 insert_small_cups 5 coin sugar large_cup cancel coin sugar cancel set_price 5 coin sugar cancel dispose

Test#11: set_price 15 insert_small_cups 6 coin small_cup coin small_cup tea insert_small_cups 1 coin small_cup small_cup tea insert_large_cups 2 dispose

Test#12: set_price 15 insert_small_cups 2 insert_large_cups 2 coin small_cup tea set_price 20 coin small_cup tea coin small_cup cancel coin small_cup tea dispose

Test#13: set_price 15 insert_large_cups 5 coin large_cup small_cup large_cup tea insert_small_cups 1 dispose

Test#14: set_price 15 insert_large_cups 5 insert_small_cups 2 coin large_cup large_cup tea set_price 5 coin large_cup tea insert_large_cups 1 dispose

Test#15: set_price 15 insert_large_cups 5 coin sugar large_cup coin small_cup large_cup tea insert_small_cups 2 dispose

Test#16: set_price 15 insert_large_cups 5 coin sugar coin coin sugar large_cup sugar coin tea insert_large_cups 2 dispose

Test#17: set_price 15 insert_large_cups 5 coin coin large_cup coin coin sugar small_cup small_cup sugar sugar large_cup large_cup sugar sugar tea set_price 5 dispose

Test#18: set_price 15 insert_small_cups 5 coin small_cup sugar tea insert_large_cups 2 coin sugar small_cup tea insert_small_cups 1 dispose

Test#19: set_price 15 insert_small_cups 5 coin sugar small_cup coin small_cup tea coin small_cup sugar tea set_price 20 dispose

Test#20: set_price 15 insert_large_cups 1 coin large_cup tea coin coin insert_large_cups 1 insert_large_cups 1 dispose

Test#21: set_price 15 insert_large_cups 1 coin large_cup tea insert_large_cups 1 insert_small_cups 1 coin large_cup tea insert_large_cups 1 dispose

Test#22: set_price 15 insert_large_cups 1 coin large_cup coin tea insert_large_cups 1 set_price 20 dispose

Test#23: set_price 15 insert_large_cups 1 coin large_cup sugar sugar tea insert_large_cups 1 coin cancel dispose

Test#24: set_price 15 coin coin cancel insert_small_cups 2 coin coin small_cup tea dispose

Test#25: set_price 15 insert_large_cups 1 coin large_cup sugar tea coin insert_large_cups 1 coin sugar large_cup tea insert_large_cups 1 dispose

Test#26: set_price 15 insert_large_cups 1 coin large_cup sugar coin tea insert_large_cups 1 coin sugar coin cancel dispose

Test#27: set_price 15 insert_small_cups 1 coin small_cup tea coin coin insert_small_cups 1 coin small_cup coin tea insert_small_cups 1 dispose

Test#28: set_price 15 insert_small_cups 1 coin sugar small_cup sugar tea insert_small_cups 1 insert_large_cups 1 dispose

Test#29: set_price 15 insert_small_cups 1 coin sugar sugar small_cup sugar sugar coin tea insert_small_cups 1 insert_small_cups 1 coin sugar sugar cancel dispose

Test#30: set_price 15 insert_large_cups 1 insert_small_cups 1 coin sugar large_cup small_cup tea insert_small_cups 1 set_price 20 dispose

Test#31: set_price 15 insert_small_cups 1 coin sugar small_cup coin tea coin insert_small_cups 3 coin sugar sugar small_cup tea coin large_cup cancel dispose

Test#32: set_price 15 insert_small_cups 1 coin small_cup sugar tea insert_small_cups 1 dispose

Test#33: set_price -5 coin small_cup large_cup sugar tea insert_large_cups -2 insert_small_cups -8 cancel dispose

Test#34: set_price 15 coin tea insert_large_cups 5 insert_small_cups 5 set_price 5 dispose cancel dispose

Test#35: set_price 15 insert_large_cups 1 coin small_cup tea dispose cancel dispose

Test#36: set_price 15 insert_small_cups 1 coin large_cup tea dispose cancel dispose

Test#37: set_price 15 coin sugar tea insert_large_cups 5 insert_small_cups 5 insert_small_cups -5 insert_large_cups -5 set_price 5 dispose cancel dispose

Test#38: set_price 15 insert_large_cups 1 coin large_cup tea small_cup large_cup sugar tea insert_large_cups -5 insert_small_cups 2 set_price 5 set_price -5 cancel dispose insert_large_cups 1 dispose

Test#39: set_price 15 insert_small_cups 1 coin small_cup tea small_cup large_cup sugar tea insert_large_cups 5 insert_small_cups -2 set_price 5 cancel dispose insert_small_cups 1 dispose

Test#40: set_price 15 insert_small_cups 1 coin sugar large_cup tea dispose

Test#41: set_price 15 insert_large_cups 1 coin sugar small_cup sugar tea dispose

Test#42: set_price 15 insert_large_cups 1 coin sugar small_cup tea dispose

Test#43: set_price 15 insert_large_cups 5 coin small_cup tea cancel insert_small_cups 5 coin sugar large_cup tea dispose

Test#44: set_price 15 insert_small_cups 6 coin large_cup tea dispose

Test#45: set_price 15 insert_large_cups 5 coin sugar small_cup tea sugar tea dispose

$$ $$

Test Results:

Test#1:

```
Enter the Choice

2

Insert_Large_Cups(int n)

Enter value of n

5

The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)
```

```
Enter value of n
5
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
10
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
```

```
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
Test #2:
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
30
The return value=1
Enter the Choice
4
Coin
```

```
The return value=1
Enter the Choice
12
Show_State

Coin Inserted state
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#3:

```
Enter the Choice
1
Set_Price(int p)
```

```
Enter value of p
60
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
```

```
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
```

```
The return value=1
```

Result: Test Case Passed

Test #4:

```
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#5:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
20
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
2
```

```
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
2
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
4
Coin
```

```
RETURN COIN
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
```

```
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#6:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
26
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
4
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
2
```

```
The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)

Enter value of n

2

The return value=1

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1
```

Result: Test Case Passed

Test#7:

```
Enter the Choice

1

Set_Price(int p)

Enter value of p

56

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice
```

```
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
30
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Test#8:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
```

```
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
3
```

```
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
5
```

```
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
10
Dispose
The return value=0
```

Result: Test Case Passed

Test#9:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
```

```
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
```

```
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#10:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
```

```
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
```

```
Enter the Choice

1

Set_Price(int p)

Enter value of p

5

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

7

Sugar

The return value=1

Enter the Choice

9

Cancel

RETURN COINS

The return value=1

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1
```

Result: Test Case Passed

Test#11:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
6
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
```

```
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
```

```
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
2
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#12:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
```

```
Insert_Large_Cups(int n)

Enter value of n

2

The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)

Enter value of n

4

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

5

Small_Cup

The return value=1

Enter the Choice

8

Tea

DISPOSE SMALL CUP OF TEA

The return value=1

Enter the Choice

1

Set_Price(int p)

Enter value of p
```

```
20
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
```

```
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#13:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
```

```
15

The return value=1

Enter the Choice

2

Insert_Large_Cups(int n)

Enter value of n

5

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

6

Large_Cup

The return value=1

Enter the Choice

5

Small_Cup

The return value=1

Enter the Choice

6

Large_Cup

The return value=1

Enter the Choice

8

Tea
```

```
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#14:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
```

```
5
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
2
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
1
```

```
Set_Price(int p)
Enter value of p
5
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
```

```
The return value=1
Test#15:
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
```

```
4

RETURN COIN
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
2
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
```

```
The return value=1


Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
```

```
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
```

```
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
2
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
```

Test#17:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=1
```

```
Enter the Choice
4

The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
7
Sugar
The return value=1
```

```
Enter the Choice

5

Small_Cup

The return value=1

Enter the Choice

5

Small_Cup

The return value=1

Enter the Choice

7

Sugar

The return value=1

Enter the Choice

7

Sugar

The return value=1

Enter the Choice

6

Large_Cup

The return value=1

Enter the Choice

6

Large_Cup

The return value=1

Enter the Choice

7

Sugar
```

```
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
5
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#18:

```
Enter the Choice
1
```

```
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
```

```
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
2
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
```

```
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#19:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
```

```
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
```

```
5
Small_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
20
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#20:

```
Enter the Choice
```

```
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
 4
Coin
```

```
RETURN COIN
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#21:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
```

```
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
```

```
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
```

Test#22:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
```

```
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
1
Set_Price(int p)
Enter value of p
20
The return value=1
```

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1

Result: Test Case Passed

Test#23:

Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

2

Insert_Large_Cups(int n)

Enter value of n

1

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

6

```
Large_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
9
```

Cancel

RETURN COINS

The return value=1

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1

Result: Test Case Passed

TestCase#24:

Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

4

Coin

RETURN COIN

The return value=1

Enter the Choice

```
9

Cancel

RETURN COINS

The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)

Enter value of n

2

The return value=1Enter the Choice

4

Coin

The return value=1

Enter the Choice

4

Coin

RETURN COIN

The return value=1

Enter the Choice

5

Small_Cup

The return value=1

Enter the Choice

8

Tea

DISPOSE SMALL CUP OF TEA

The return value=1
```

```
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
```

Test#25:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
```

```
Large_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
```

```
7

Sugar

The return value=1

Enter the Choice

6

Large_Cup

The return value=1

Enter the Choice

8

Tea

DISPOSE LARGE CUP OF TEA WITH SUGAR

The return value=1

Enter the Choice

2

Insert_Large_Cups(int n)

Enter value of n

1

The return value=1

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1
```

Result: Test Case Passed

Test#26:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
4
Coin
```

```
RETURN COIN
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
```

```
9

Cancel

RETURN COINS

The return value=1

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1
```

Result: Test Case Passed

Test#27:

```
Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)

Enter value of n

1

The return value=1

Enter the Choice

4
```

```
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
```

```
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
```

SHUT DOWN

The return value=1

Result: Test Case Passed

Test#28:

Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)

Enter value of n

1

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

7

Sugar

The return value=1

Enter the Choice

```
5
Small_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
```

```
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#29:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
```

```
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
3
```

```
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
```

```
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
```

Test#30:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
```

```
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
```

```
Enter the Choice

1

Set_Price(int p)

Enter value of p

20

The return value=1

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1
```

Result: Test Case Passed

Test#31:

```
Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)

Enter value of n

1

The return value=1
```

```
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
4
Coin
RETURN COIN
The return value=1
```

```
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
2
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
```

```
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#32:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
```

```
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
```

```
Enter value of n
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
```

Test#33-Test#39 – Ghost(default) Cases:

Test#33:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
-5
The return value=0
Enter the Choice
4
Coin
The return value=0
Enter the Choice
5
Small_Cup
The return value=0
```

```
Enter the Choice
6
Large_Cup
The return value=0
Enter the Choice
7
Sugar
The return value=0
Enter the Choice
7
Sugar
The return value=0
Enter the Choice
8
Tea
The return value=0
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
-2
The return value=0
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
-8
```

```
The return value=0

Enter the Choice

9

Cancel

The return value=0

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1

Result: Test Case Passed



Test#34

Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

8

Tea
```

```
The return value=0
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=0
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=0
Enter the Choice
1
Set_Price(int p)
Enter value of p
5
The return value=0
Enter the Choice
10
Dispose
The return value=0
Enter the Choice
9
Cancel
RETURN COINS
```

```
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
```

Test#35:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
```

5

Small_Cup

The return value=1

Enter the Choice

8

Tea

The return value=0

Enter the Choice

10

Dispose

The return value=0

Result: Test Case Passed

Test#36:

Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

2

Insert_Small_Cups(int n)

Enter value of n

1

The return value=1

Enter the Choice

```
4
Coin
The return value=1
Enter the Choice
5
Large_Cup
The return value=1
Enter the Choice
8
Tea
The return value=0
Enter the Choice
10
Dispose
The return value=0
```

Result: Test Case Passed

Test#37:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
4
```

```
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
The return value=0
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=0
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=0
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
-5
```

```
The return value=0
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
-5
The return value=0
Enter the Choice
1
Set_Price(int p)
Enter value of p
5
The return value=0
Enter the Choice
10
Dispose
The return value=0
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#38:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
```

```
DISPOSE LARGE CUP OF TEA
The return value=1
Enter the Choice
5
Small_Cup
The return value=0
Enter the Choice
6
Large_Cup
The return value=0
Enter the Choice
7
Sugar
The return value=0
Enter the Choice
8
Tea
The return value=0
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
-5
The return value=0
Enter the Choice
3
Insert_Small_Cups(int n)
```

```
Enter value of n

2

The return value=0

Enter the Choice

1

Set_Price(int p)

Enter value of p

5

The return value=0

Enter the Choice

1

Set_Price(int p)

Enter value of p

-5

The return value=0

Enter the Choice

9

Cancel

The return value=0

Enter the Choice

10

Dispose

The return value=0

Enter the Choice

2

Insert_Large_Cups(int n)

Enter value of n
```

```
1
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
Result: Test Case Passed
```

```
Test#39:
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Small_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
```

```
Enter the Choice
6
Small_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE SMALL CUP OF TEA
The return value=1
Enter the Choice
5
Small_Cup
The return value=0
Enter the Choice
6
Large_Cup
The return value=0
Enter the Choice
7
Sugar
The return value=0
Enter the Choice
8
Tea
The return value=0
Enter the Choice
2
```

```
Insert_Large_Cups(int n)
Enter value of n
5
The return value=0
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
-2
The return value=0
Enter the Choice
1
Set_Price(int p)
Enter value of p
5
The return value=0
Enter the Choice
1
Enter the Choice
9
Cancel
The return value=0
Enter the Choice
10
Dispose
The return value=0
Enter the Choice
```

2

Insert_Small_Cups(int n)

Enter value of n

1

The return value=1

Enter the Choice

10

Dispose

SHUT DOWN

The return value=1

Result: Test Case Passed

Test#40:

Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

3

Insert_Small_Cups(int n)

Enter value of n

1

The return value=1

Enter the Choice

4

Coin

The return value=1

Enter the Choice

7

Sugar

The return value=1

Enter the Choice

6

Large_Cup

The return value=1

Enter the Choice

8

Tea

The return value=0

Enter the Choice

10

Dispose

The return value=0

Result: Test Case Passed

Test#41:

Enter the Choice

1

Set_Price(int p)

Enter value of p

```
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
8
Tea
```

```
The return value=0
Enter the Choice
10
Dispose
The return value=0
Result: Test Case Passed
```

Test#42:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
1
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
```

Sugar

The return value=1

Enter the Choice

5

Small_Cup

The return value=1

Enter the Choice

8

Tea

The return value=0

Enter the Choice

10

Dispose

The return value=0

Result: Test Case Passed

Test#43:

Enter the Choice

1

Set_Price(int p)

Enter value of p

15

The return value=1

Enter the Choice

2

Insert_Large_Cups(int n)

```
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
Tea
The return value=0
Enter the Choice
9
Cancel
RETURN COINS
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
5
The return value=1 Enter the Choice
4
```

```
Coin
The return value=1 Enter the Choice
7
Sugar
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
DISPOSE LARGE CUP OF TEA WITH SUGAR
The return value=1
Enter the Choice
10
Dispose
SHUT DOWN
The return value=1
```

Result: Test Case Passed

Test#44:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
```

```
15
The return value=1
Enter the Choice
3
Insert_Small_Cups(int n)
Enter value of n
6
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
6
Large_Cup
The return value=1
Enter the Choice
8
Tea
The return value=0
Enter the Choice
10
Dispose
The return value=0
Result: Test Case Passed
```

Test#45:

```
Enter the Choice
1
Set_Price(int p)
Enter value of p
15
The return value=1
Enter the Choice
2
Insert_Large_Cups(int n)
Enter value of n
5
The return value=1
Enter the Choice
4
Coin
The return value=1
Enter the Choice
7
Sugar
The return value=1
Enter the Choice
5
Small_Cup
The return value=1
Enter the Choice
8
```

```
Tea

The return value=0

Enter the Choice

7

Sugar

The return value=1

Enter the Choice

8

Tea

The return value=0

Enter the Choice

10

Dispose

The return value=0
```

Result: Test Case Passed

Source Code:

VendingMachine.java

```java
public class VendingMachine
{
	private int x;
	private int price;
	private int k;
	private int k1;
	private int t;
	private int s;

	public VendingMachine()
	{
		k1 = 0;
		k = 0;
		t = 0;
		price = 0;
		x = 1;
	}
	public final int coin()
```

```
{
    if (x == 1)
    {
        if ((t + 25 >= price) && (price > 0))
        {
            s = 0;
            t = 0;
            x = 2;
            return 1;
        }
        else if (t + 25 < price)
        {
            t = t + 25;
            return 1;
        }
    }
    else if ((x > 1) && (x < 6))
    {
        System.out.print("RETURN COIN");
        System.out.print("\n");
        return 1;
    }
    return 0;
}
public final int small_cup()
{
    if ((x == 2) || (x == 3))
    {
        s = 2;
        return 1;
    }
    return 0;
}
public final int large_cup()
{
    if ((x == 2) || (x == 3))
    {
        s = 1;
        return 1;
    }
    return 0;
}
public final int sugar()
{
    if ((x == 2) || (x == 3))
    {
        if (x == 2)
        {
            x = 3;
        }
        else
        {
            x = 2;
        }
        return 1;
        }
        return 0;
    }
    public final int tea()
    {
        if ((x == 2) || (x == 3))
        {
            if ((x == 2) && (k1 > 1) && (s == 2))
            {
                System.out.print("DISPOSE SMALL CUP OF TEA");
                System.out.print("\n");
                k1 = k1 - 1;
                x = 1;
                return 1;
            }
            else if ((x == 2) && (k > 1) && (s == 1))
            {
                System.out.print("DISPOSE LARGE CUP OF TEA");
                System.out.print("\n");
                k = k - 1;
                x = 1;
                return 1;
            }
            else if ((x == 2) && (k == 1) && (s == 1))
            {
                System.out.print("DISPOSE LARGE CUP OF TEA");
                System.out.print("\n");
                k = k - 1;
                x = 5;
                return 1;
            }
            else if ((x == 2) && (k1 == 1) && (s == 2))
            {
                System.out.print("DISPOSE SMALL CUP OF TEA");
                System.out.print("\n");
                k1 = k1 - 1;
                x = 4;
                return 1;
            }
            else if ((x == 3) && (k1 == 1) && (s == 2))
            {
                System.out.print("DISPOSE SMALL CUP OF TEA WITH SUGAR");
                System.out.print("\n");
                k1 = k1 - 1;
                x = 4;
                return 1;
            }
            else if ((x == 3) && (k == 1) && (s == 1))
            {
                System.out.print("DISPOSE LARGE CUP OF TEA WITH SUGAR");
                System.out.print("\n");
                k = k - 1;
                x = 5;
                return 1;
            }
```

```
                if ((x == 3) && (k1 > 1) && (s == 2))
                {
                        System.out.print("DISPOSE SMALL CUP OF TEA WITH SUGAR");
                        System.out.print("\n");
                        k1 = k1 - 1;
                        x = 1;
                        return 1;
                }
                else if ((x == 3) && (k > 1) && (s == 1))
                {
                        System.out.print("DISPOSE LARGE CUP OF TEA WITH SUGAR");
                        System.out.print("\n");
                        k = k - 1;
                        x = 1;
                        return 1;
                }
                return 0;
        }
        return 0;
}
public final int insert_large_cups(int n)
{
        if ((x == 1) && (n > 0))
        {
                k = k + n;
                return 1;
        }
        else if ((x == 5) && (n > 0))
        {
                k = n;
                x = 1;
                return 1;
        }
        return 0;
}
public final int insert_small_cups(int n)
{
        if ((x == 1) && (n > 0))
        {
                k1 = k1 + n;
                return 1;
        }
        else if ((x == 4) && (n > 0))
        {
                k1 = n;
                x = 1;
                return 1;
        }
        return 0;
}
public final int set_price(int p)
{
        if ((x == 1) && (p > 0))
        {
                price = p;
```

```
            return 1;
        }
        return 0;
    }
    public final int cancel()
    {
        if ((x == 2) || (x == 3))
        {
            System.out.print("RETURN COINS");
            System.out.print("\n");
            x = 1;
            return 1;
        }
        return 0;
    }
    public final int dispose()
    {
        if ((x == 1))
        {
            System.out.print("SHUT DOWN");
            System.out.print("\n");
            x = 6;
            return 1;
        }
        return 0;
    }



//  #Testing Oriented Methods

    void Show_State(){
        System.out.println("");
        if(x==1){
            System.out.println("Idle state");
        }else if(x==2){
            System.out.println("Coin Inserted state");
        }else if(x==3){
            System.out.println("Sugar state");
        }else if(x==4){
            System.out.println("No_Large_Cups state");
        }else if(x==5){
            System.out.println("No_Small_Cups state");
        }
        System.out.println("");
    }

    void Show_All_Values(){
        System.out.println("Price="+price);
        System.out.println("No_of_Large_Cups="+k);
        System.out.println("No_of_small_Cups="+k1);
        System.out.println("T="+t);
        System.out.println("CurrentState="+x);
    }
```

```
}
```

VendingMachineDriver.java

```
import java.io.*;

import java.util.Scanner;



public class VendingMachineDriver

{

  public static void main(String [] args)

  {


           int userChoice=0;

           int retr= 0;

           int n;


          VendingMachine obj=new VendingMachine();

          Scanner in = new Scanner(System.in);



         while(userChoice!=11){


                 System.out.println("Please choose from below options");

                 System.out.println("1.Set_Price");

                 System.out.println("2.Insert_Large_Cups");

                 System.out.println("3.Insert_Small_Cups");

                 System.out.println("4.Coin");
                 System.out.println("5.Small_Cup");
                 System.out.println("6.Large_Cup");
                 System.out.println("7.Sugar");
                 System.out.println("8.Tea");
                 System.out.println("9.Cancel");
                 System.out.println("10.Dispose");
                 System.out.println("");
                 System.out.println("");
                 System.out.println("11.Quit");
                 System.out.println("");
                 System.out.println("");
                 System.out.println("Testing Oriented Methods");
                 System.out.println("12.Show_state");
                 System.out.println("13.Show_All_Values");
                 System.out.println("");
                 System.out.println("Enter the Choice");
                 userChoice=in.nextInt();

                 switch(userChoice){
                 case 1:
                         int price;
                         System.out.println("Set_Price(int p)");
                         System.out.println("Enter value of p");
                         price=in.nextInt();

                         retr=obj.set_price(price);
                         System.out.println("The return value="+retr);
```

```
            break;
case 2:
            System.out.println("Insert_Large_Cups(int n)");
            System.out.println("Enter value of n");
            n=in.nextInt();

            retr=obj.insert_large_cups(n);
            System.out.println("The return value="+retr);
            break;
case 3:

            System.out.println("Insert_Small_Cups(int n)");
            System.out.println("Enter value of n");
            n=in.nextInt();

            retr=obj.insert_small_cups(n);
            System.out.println("The return value="+retr);
            break;
case 4:
            System.out.println("Coin");
            retr=obj.coin();
            System.out.println("The return value="+retr);
            break;
case 5:
            System.out.println("Small_Cup");
            retr=obj.small_cup();
            System.out.println("The return value="+retr);
```

```
        break;
case 6:
        System.out.println("Large_Cup");
        retr=obj.large_cup();
        System.out.println("The return value="+retr);
        break;
case 7:
        System.out.println("Sugar");
        retr=obj.sugar();
        System.out.println("The return value="+retr);
        break;
case 8:
        System.out.println("Tea");
        retr=obj.tea();
        System.out.println("The return value="+retr);
        break;
case 9:
        System.out.println("Cancel");
        retr=obj.cancel();
        System.out.println("The return value="+retr);
        break;
case 10:
        System.out.println("Dispose");
        retr=obj.dispose();
        System.out.println("The return value="+retr);
        break;
case 12:
```

```
                        System.out.println("Show_State");
                        obj.Show_State();
                        System.out.println("");
                        break;
                case 13:
                        System.out.println("Show_All_Values");
                        obj.Show_All_Values();
                        System.out.println("");
                        break;
                default:
                        if(userChoice!=11){
                        System.out.println("Please choose only from the given options!!");
                        System.out.println("Else Enter 11 to quit");
                        }
                                break;
                }
                System.out.println("");
                System.out.println("");

        }

        System.out.println("VendingMachine Driver is Stopped");
        System.exit(0);
  }
}
```

Conclusion:

Class and Model based testing demands complete understanding of the logic of operations/methods/actions in the given class or EFSM. Creating our own test driver helped us to reduce the human errors. Thus, a testing environment increases the reliability. Testing process as a whole needs a lot of time, effort and concentration.

All the test cases have been executed and it was found that there are no errors in the source code. While performing multiple condition testing, it was found that some of the conditions were such that there are no test cases for that condition.