**Software Testing and Analysis(CS589)**

**Final Project**

(Professor: Dr. Bogdan Korel)

**Submitted by:**

**Kumar Singh, Dhirendra**

(CWID: **A20331285**)

# Table of Contents

# 1. Introduction ----------------------------------------------------------------------------2

# 2. Model-based testing of GAS PUMP class(Transition Pair)------------------3

# 3. Ghost Transitions of Gas Pump class---------------------------------------------8

# 4. Multiple Condition/Branch Testing of Gas Pump class--------------------17

# 5. Test Suit(TS.txt file) and the results after execution-----------------------30

# 6. Conclusion-----------------------------------------------------------------------------65

# 7. Well documented source code(implemented) of the GasPump class and the test driver----------------------------------------------------------------------66

# 8. Detailed Instruction for executing the code in CD------------------------83

# 1. Introduction

The goal of this project was to test *GasPump* class that exhibits state behavior specified by the EFSM model.

The *GasPump* class is a state-based class that is used to control a simple gas pump. The gas pump disposes two types of gasoline: Regular and Super. The price of each type of gasoline is provided when the gas pump is activated. Users can pay the amount by cash or a credit card.

We need to document various test cases to test the provided source code and all possible scenarios of the GasPump class. Three testing methods are used for this project - Model Based testing(Transition Pair), Default/ghost transition testing and Multiple condition testing.

Source code was provided to us for this GasPump class and we were supposed to implement a testing environment that should contain a class test driver to execute the test cases .

A Test Suit of test cases was asked to design considering all the three methods(Model Based Testing-Transition Pair, Default/Ghost Testing and Multi-Condition / Branch Testing) in a text file(TS.txt)

There are 9 test cases which covers all 2-transition sequence (Model based testing), 3 test cases which covers all ghost transtitions for every state (S0,S1,S2,S3,S4,S5,S6) and 4 test cases for Multilple condition testing, making total of 16 test cases.

There are 11 cases which are not possible and the reasons are provided in detail after every case. Result =0 means it’s a default/ghost transition and Result=1 means transition is as per the flow.

I have used “**JAVA**” as the source code for this project and have used “**Netbeans**”.

# 2. Model-based testing of GAS PUMP class(Transition Pair)

**S0:**

**IN:** T0, T3, T5, T13, T14

**OUT:** T1, T15

TRANSITION PAIRS:

(T0, T1) (T0, T15) (T3, T1) (T3, T15) (T5, T1) (T5, T15) (T13, T1) (T13, T15) (T14, T1) (T14, T15)

**S1:**

**IN:** T1

**OUT:** T2, T6

TRANSITION PAIRS:

(T1, T2) (T1, T6)

**S2:**

**IN:** T2

**OUT:** T3, T4

TRANSITION PAIRS:

(T2, T3) (T2, T4)

**S3**:

**IN:** T4, T6

**OUT:** T5, T7, T8

TRANSITION PAIRS:

(T4, T5) (T4, T7) (T4, T8) (T6, T5) (T6, T7) (T6, T8)

**S4:**

**IN:** T7, T8

**OUT:** T9

TRANSITION PAIRS:

(T7, T9) (T8, T9)

**S5:**

**IN:** T9, T10

**OUT:** T10, T11, T12

TRANSITION PAIRS:

(T9, T10) (T9, T11) (T9, T12) (T10, T10) (T10, T11) (T10, T12)

**S6:**

**IN:** T11, T12

**OUT:** T13, T14

TRANSITION PAIRS:

(T11, T13) (T11, T14) (T12, T13) (T12, T14)

**TEST CASES:**

**Test#1**: Activate(4 ,5) ,Start() ,PayCredit() ,Approved() ,Regular() ,StartPump() ,PumpGallon() ,StopPump(), Receipt() ,Start() ,PayCash(10) ,Super() ,StartPump(),PumpGallon() ,PumpGallon() ,StopPump() ,NoReceipt(), TurnOff()

T0,T1,T2,T4,T7,T9,T10,T11,T13,T1,T6,T8,T9,T10,T10,T11,T14,T15

TRANSITION PAIRS:

(T0,T1) (T1,T2) (T2,T4) (T4,T7) (T7,T9) (T9,T10) (T10,T11) (T11,T13) (T13,T1) (T1,T6) (T6,T8) (T8,T9) (T9,T10) (T10,T10) (T10,T11) (T11,T14) (T14,T15)

**Test#2**: Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Start() ,PayCash(10) ,Cancel() ,Start() ,PayCash(10), Regular() ,StartPump() ,StopPump() ,Receipt() ,TurnOff()

T0,T1,T2,T3,T1,T6,T5,T1,T6,T7,T9,T11,T13,T15

TRANSITION PAIRS:

(T0,T1) (T1,T2) (T2,T3) (T3,T1) (T1,T6) (T6,T5) (T5,T1) (T1,T6) (T6,T7) (T7,T9) (T9,T11) (T11,T13) (T13,T15)

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T2,T3) (T3,T1) (T6,T5) (T5,T1) (T6,T7) (T9,T11) (T13,T15)**

**Test#3**: Activate(4 ,7) ,Start ,PayCredit() ,Approved() ,Cancel() ,TurnOff()

T0,T1,T2,T4,T5,T15

TRANSITION PAIRS:

(T0,T1) (T1,T2) (T2,T4) (T4,T5) (T5,T15)

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T4,T5) (T5,T15)**

**Test#4**: Activate(4 ,7) ,Start() ,PayCredit() ,Approved() ,Super() ,StartPump() ,PumpGallon() ,StopPump(), NoReceipt() ,TurnOff()

T0,T1,T2,T4,T8,T9,T10,T11,T14,T15

TRANSITION PAIRS:

(T0,T1) (T1,T2) (T2,T4) (T4,T8) (T8,T9) (T9,T10) (T10,T11) (T11,T14) (T14,T15)

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T4,T8)**

**Test#5**: Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,TurnOff()

T0,T1,T2,T3,T15

TRANSITION PAIRS:

(T0,T1) (T1,T2) (T2,T3) (T3,T15)

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T3,T15)**

**Test#6**: Activate(4 ,7) ,TurnOff()

T0,T15

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T0,T15)**

**Test#7**: Activate(4 ,7) ,Start() ,PayCash(3) ,Super() ,StartPump() ,PumpGallon() ,NoReceipt() ,TurnOff()

T0,T1,T6,T8,T9,T12,T14,T15

TRANSITION PAIRS:

(T0,T1) (T1,T6) (T6,T8) (T8,T9) (T9,T12) (T12,T14) (T14,T15)

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T9,T12) (T12,T14)**

**Test#8**: Activate(4 ,6) ,Start() ,PayCash(6) ,Regular() ,StartPump() ,PumpGallon() ,PumpGallon() ,Receipt() ,TurnOff()

T0,T1,T6,T7,T9,T10,T12,T13,T15

TRANSITION PAIRS:

(T0,T1) (T1,T6) (T6,T7) (T7,T9) (T9,T10) (T10,T12) (T12,T13) (T13,T15)

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T10,T12) (T12,T13)**

**Test#9**: Activate(4 ,5) ,Start() ,PayCredit() ,Approved() ,Regular() ,StartPump() ,PumpGallon() ,StopPump() ,NoReceipt() ,Start() ,PayCredit() ,Reject() ,TurnOff()

T0,T1,T2,T4,T7,T9,T10,T11,T14,T1,T2,T3,T15

TRANSITION PAIRS:

(T0,T1) (T1,T2) (T2,T4) (T4,T7) (T7,T9) (T9,T10) (T10,T11) (T11,T14) (T14,T1) (T1,T2) (T2,T3) (T3,T15)

**ADDITIONAL TRANSITIONS COVERED ARE:**

**(T14,T1)**

# 

# 3. Ghost Transitions of Gas Pump class

Default/Ghost transitions are the ones which can be executed but the execution of those transactions doesn’t affect the state of the system, the state remains the same.

**S0 state** has 14 default/ghost transitions and **S1 state** has also 14 default/ghost transitions.

**S0 state(Default/ghost transitions):**

1.Activate(a,b)

2. PayCredit()

3. Approved()

4. Cancel()

5. PayCash(10)

6. Regular()

7. Super()

8. StartPump()

9. PumpGallon()

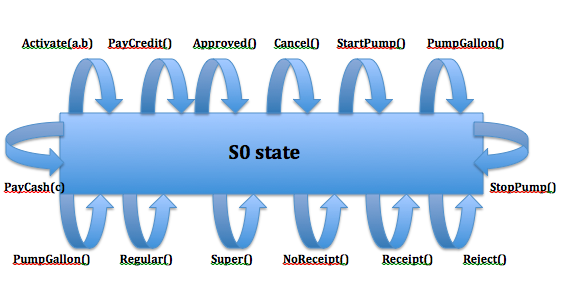
10. PumpGallon()

11. StopPump()

12. NoReceipt()

13. Receipt()

14. Reject()



**S1 state(Default/ghost transitions):**

1.Activate(a,b)

2. Start()

3. Approved()

4. Reject()

5. Cancel()

6. Regular()

7. Super()

8. StartPump()

9. PumpGallon()

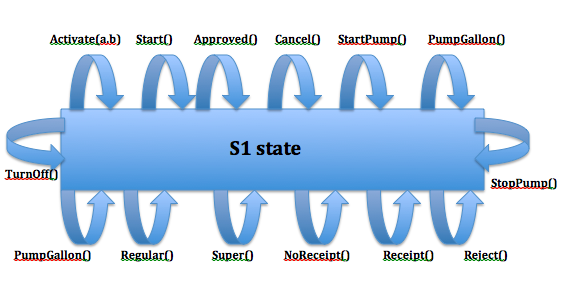
10. PumpGallon()

11. StopPump()

12. Receipt()

13. NoReceipt()

14. TurnOff()



**Combined test cases for both S0 state and S1 state:**

**Test#10**: Activate(4 ,5) ,Activate(4 ,5) ,PayCredit() ,Approved() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,PumpGallon() ,PumpGallon() ,StopPump() ,NoReceipt() ,Receipt() ,Reject() ,Start() ,Activate(4 ,5) ,Start() ,Approved() ,Reject() ,Cancel() ,Regular() ,Super() ,StartPump() ,PumpGallon() ,PumpGallon(), StopPump() ,Receipt() ,NoReceipt() ,TurnOff() ,PayCredit() ,Reject() ,TurnOff()

**S2 state** has 14 default/ghost transitions, **S3 state** has 13 default/ghost transitions and **S4 state** has 15 default/ghost transitions.

**S2 state(Default/ghost transitions):**

1.Activate(a,b)

2. Start()

3. PayCredit()

4. Cancel()

5. PayCash(10)

6. Regular()

7. Super()

8. StartPump()

9. PumpGallon()

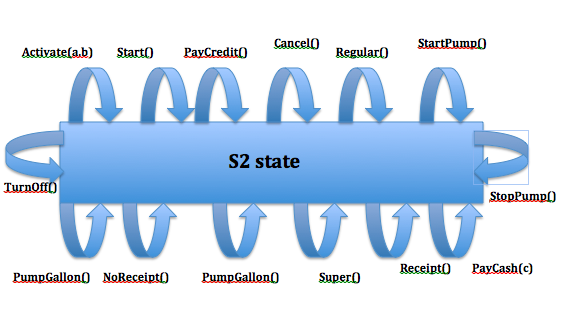
10. StopPump()

11. PumpGallon()

12. Receipt()

13. NoReceipt()

14. TurnOff()



**S3 state(Default/ghost transitions):**

1.Activate(a,b)

2. Start()

3. PayCredit()

4. Reject()

5. Approved()

6. PayCash(10)

7. StartPump()

8. PumpGallon()

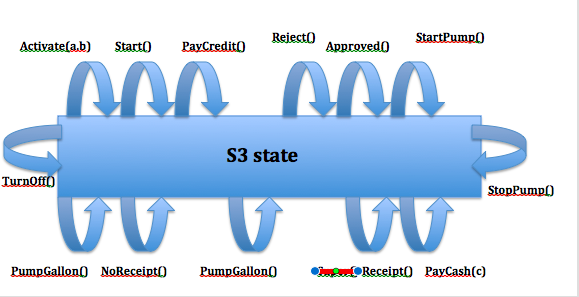
9. StopPump()

10. PumpGallon()

11. Receipt()

12. NoReceipt()

13. TurnOff()



**S4 state(Default/ghost transitions):**

1.Activate(a,b)

2. Start()

3. PayCredit()

4. Reject()

5. Approved()

6. Cancel()

7. PayCash(-10)

8. Regular()

9. Super()

10. PumpGallon()

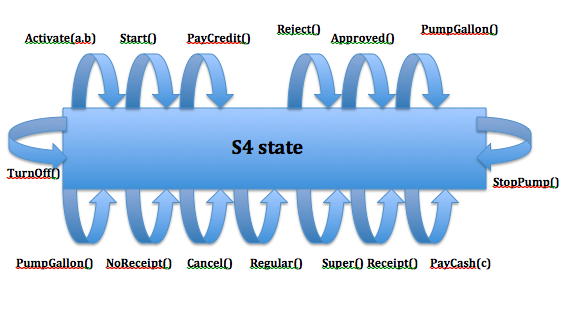
11. StopPump()

12. PumpGallon()

13. Receipt()

14. NoReceipt()

15. TurnOff()



**Combined test cases for S2 state ,S3 state and S4 state:**

**Test#11**: Activate(4 ,7) ,Start() ,PayCredit() ,Activate(4 ,5) ,Start() ,PayCredit() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,PumpGallon() ,StopPump() ,PumpGallon() ,Receipt() ,NoReceipt() ,TurnOff() ,Approved() ,Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Approved() ,PayCash(10) ,StartPump(), PumpGallon() ,StopPump() ,PumpGallon() ,Receipt() ,NoReceipt() ,TurnOff() ,Super() ,Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Approved() ,Cancel() ,PayCash(-10) ,Regular() ,Super() ,PumpGallon() ,StopPump(), PumpGallon() ,Receipt() ,NoReceipt() ,TurnOff() ,StartPump() ,PumpGallon() ,StopPump() ,NoReceipt() ,TurnOff()

**S5 state** has 13 default/ghost transitions and **S6 state** has 14 default/ghost transitions.

**S5 state(Default/ghost transitions):**

1.Activate(a,b)

2. Start()

3. PayCredit()

4. Reject()

5. Approved()

6. Cancel()

7. PayCash(10)

8. Regular()

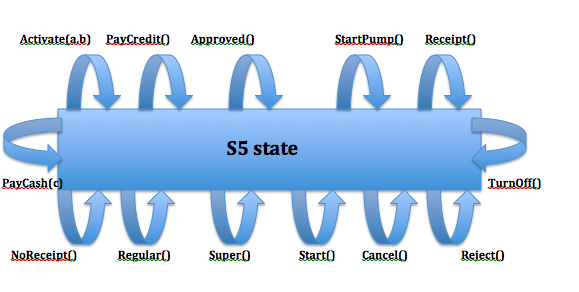
9. Super()

10. StartPump()

11. Receipt()

12. NoReceipt()

13. TurnOff()



**S6 state(Default/ghost transitions):**

1.Activate(a,b)

2. Start()

3. PayCredit()

4. Reject()

5. Approved()

6. Cancel()

7. PayCash(10)

8. Regular()

9. Super()

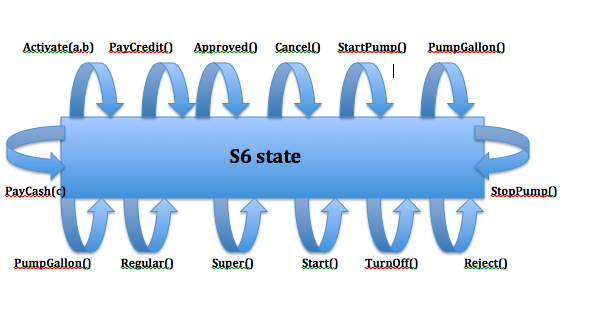
10. StartPump()

11. StopPump()

12. PumpGallon()

13. TurnOff()

14. PumpGallon()

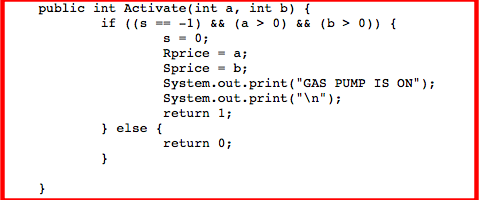


**Combined test cases for both S5 state and S6 state:**

**Test#12**: Activate(4 ,7) ,Start() ,PayCredit() ,Approved(),Regular() ,StartPump() ,Activate(4 ,7) ,Start(), PayCredit() ,Reject() ,Approved() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,Receipt(), NoReceipt() ,TurnOff() , StopPump() ,Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Approved() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,StopPump() ,PumpGallon() ,TurnOff() , PumpGallon() ,NoReceipt() ,TurnOff()

# 4. Multiple Condition/Branch Testing of Gas Pump class

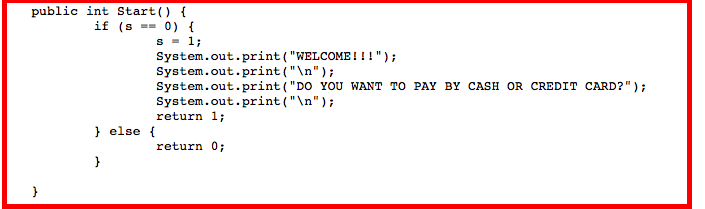
i. Activate(a,b)

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Activate(a,b) [(a>0)&&(b>0)]** | | | |
| **s == -1 &&** | **(a > 0) &&** | **(b > 0)** | **Covered in TestCase#** |
| TRUE | TRUE | TRUE | Test#1 |
| TRUE | TRUE | FALSE | Test#13 |
| TRUE | FALSE | TRUE | Test#13 |
| TRUE | FALSE | FALSE | Test#13 |
| FALSE | TRUE | TRUE | Test#11 |
| FALSE | TRUE | FALSE | Test#13 |
| FALSE | FALSE | TRUE | Test#13 |
| FALSE | FALSE | FALSE | Test#13 |

**Test#13**: Activate(4 ,-4) ,Activate(-3 ,2) ,Activate(-3 ,-2) ,Activate(4 ,5) ,Start() ,PayCredit() ,Activate(4 ,-4) ,Activate(-3 ,5) ,Activate(-3 ,-4) ,Reject() ,TurnOff()

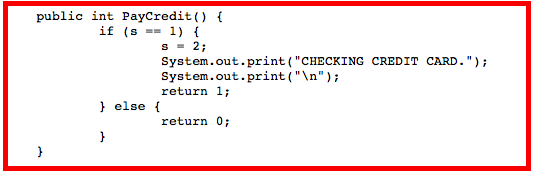
ii. Start()

****

|  |  |
| --- | --- |
| **Start** | |
| **s==0** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

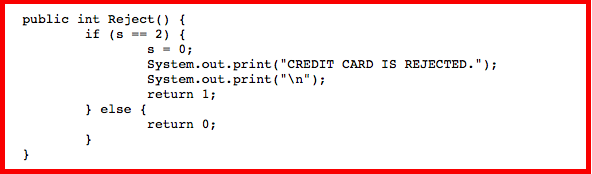
**Test#14**: Activate(4 ,-7) ,Start() ,PayCredit() ,Reject() ,Cancel() ,Approved() ,Regular() ,Super() ,StartPump(), PumpGallon() ,StopPump() ,NoReceipt() ,Receipt() ,TurnOff() ,Activate(4 ,7) ,TurnOff()

iii. PayCredit()



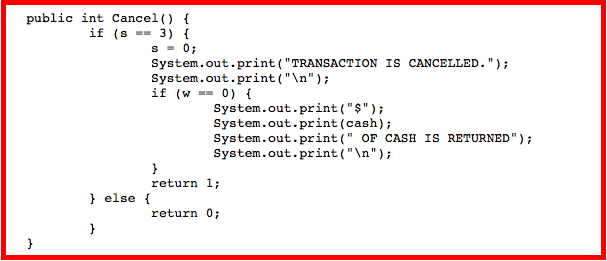
|  |  |
| --- | --- |
| **PayCredit** | |
| **s==1** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

iv. Reject()



|  |  |
| --- | --- |
| **Reject** | |
| **s==2** | **Covered in TestCase#** |
| TRUE | Test#5 |
| FALSE | Test#14 |

v. Cancel()

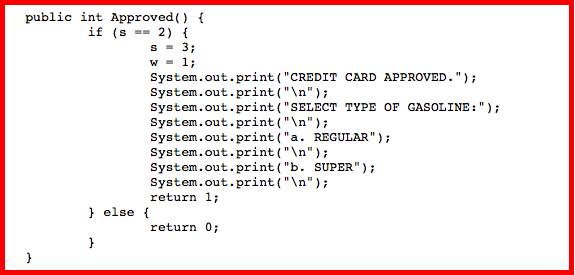


|  |  |
| --- | --- |
| **Cancel** | |
| **s==3** | **Covered in TestCase#** |
| TRUE | Test#3 |
| FALSE | Test#14 |

|  |  |
| --- | --- |
| **Cancel** | |
| **w==0** | **Covered in TestCase#** |
| TRUE | Test#15 |
| FALSE | Test#3 |

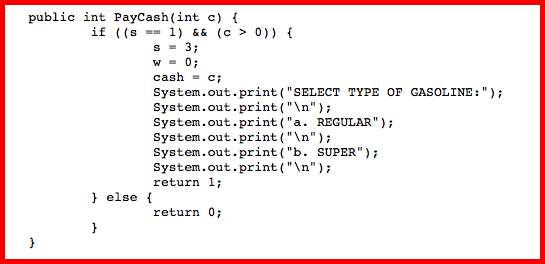
**Test#15**: Activate(4 ,7) ,Start() ,PayCash(-10) ,PayCash(10) ,Cancel() ,TurnOff()

vi. Approved()



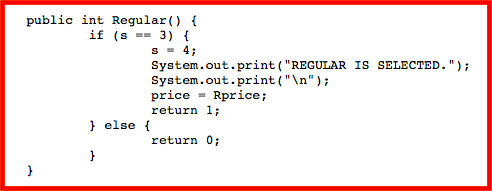
|  |  |
| --- | --- |
| **Approved** | |
| **s==2** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

vii. PayCash(c)

****

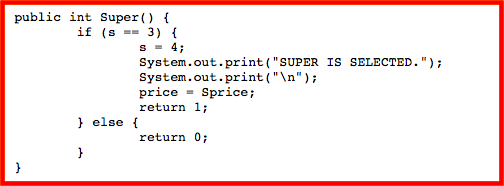
|  |  |  |
| --- | --- | --- |
| **PayCash(c) [c>0]/cash=c** | | |
| **(s == 1) &&** | **(c > 0)** | **Covered in TestCase#** |
| TRUE | TRUE | Test#7 |
| TRUE | FALSE | Test#15 |
| FALSE | TRUE | Test#11 |
| FALSE | FALSE | Test#11 |

viii. Regular()

****

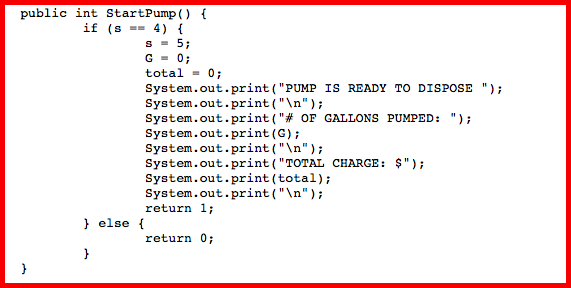
|  |  |
| --- | --- |
| **Regular** | |
| **s==3** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

ix. Super()



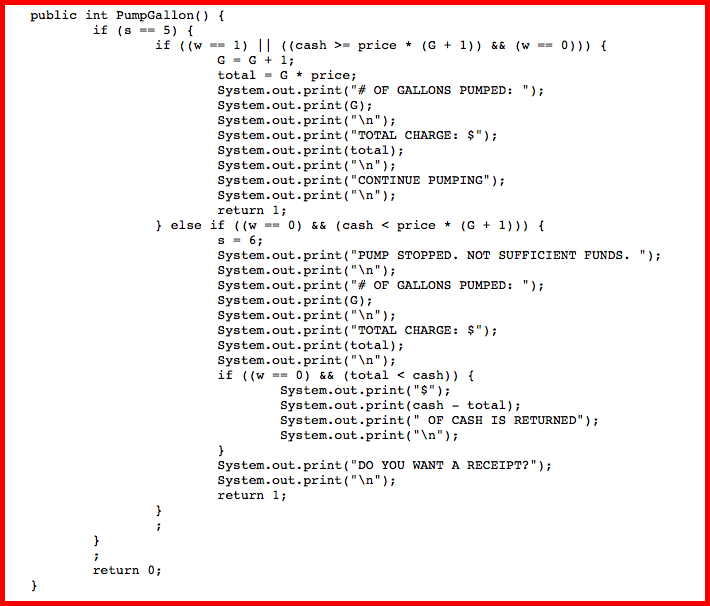
|  |  |
| --- | --- |
| **Super** | |
| **s==3** | **Covered in TestCase#** |
| TRUE | Test#4 |
| FALSE | Test#14 |

x. StartPump()



|  |  |
| --- | --- |
| **StartPump** | |
| **s==4** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

xi. PumpGallon()



**IF Condition:**

|  |  |  |  |
| --- | --- | --- | --- |
| **PumpGallon** | | | |
| **w==1 ||** | **(w==0)&&** | **(cash>=price\*(G+1))** | **Covered in TestCase#** |
| TRUE | TRUE | TRUE | **NOT POSSIBLE** |
| TRUE | TRUE | FALSE | **NOT POSSIBLE** |
| TRUE | FALSE | TRUE | **NOT POSSIBLE** |
| TRUE | FALSE | FALSE | Test#3 |
| FALSE | TRUE | TRUE | Test#1 |
| FALSE | TRUE | FALSE | Test#7 |
| FALSE | FALSE | TRUE | **NOT POSSIBLE** |
| FALSE | FALSE | FALSE | **NOT POSSIBLE** |

**Reason for Not Possible:**

1. **w==1[TRUE]||w==0[TRUE]&&cash>=price\*(G+1) [TRUE**]🡪 All three conditions cannot be true at a time because w==1 means payment to be done by credit card and w==0 means payment to be done by cash and both cannot be true at a time. Either of them can be used at a time.
2. **w==1[TRUE]||w==0[TRUE]&&cash>=price\*(G+1)[FALSE]**🡪 In this statement, w==1 is true and w==0 is also true, so this is also not possible because either of them can be used at a time for the payment. Both cannot be used at a time.
3. **w==1[TRUE]||w==0[FALSE]&&cash>=price\*(G+1)[ TRUE]**🡪 In this statement, if w==1 is true , rest both the conditions has to be false because the payment is done by credit card and not cash.
4. **w==1[FALSE]||w==0[FALSE]&&cash>=price\*(G+1)[TRUE]**🡪 In this statement, w==1 is false and w==0 is also false which is not possible. Either of the two conditions has to be true.
5. **w==1[FALSE]||w==0[FALSE]&&cash>=price\*(G+1)[FALSE]**🡪 In this statement also, both w==1 and w==0 are false which is not possible. Either of the two conditions has to be true.

**ELSE IF Condition:**

|  |  |  |
| --- | --- | --- |
| **PumpGallon** | | |
| **(w==0)&&** | **(cash<price\*(G+1))** | **Covered in TestCase#** |
| TRUE | TRUE | Test#7 |
| TRUE | FALSE | **NOT POSSIBLE** |
| FALSE | TRUE | **NOT POSSIBLE** |
| FALSE | FALSE | **NOT POSSIBLE** |

**Reason for Not Possible:**

1. **w==0[TRUE]&&cash<price\*(G+1) [FALSE]**🡪 In this statement, w==0 is true and cash< price\*(G+1) is false but cash can never be less than price\*(G+1).

**Example:** If you take cash=1 and price of the regular fuel is 1, you go on “StartPump” where G=0; So, cash is 1 and price is 1.

Condition is: 1<1\*(0+1) **which can never be true.**

1. **w==0[FALSE]&&cash<price\*(G+1) [TRUE]**🡪 In this statement, w==0 is false, that means the payment is done by Creditcard and not cash, so second condition can never be true.
2. **w==0[FALSE]&&cash<price\*(G+1) [FALSE]**-> In this statement, w==0 is false, that means the payment is done by Creditcard and not cash, so the second condition is false, that means , cash>= price\*(G+1) which is never possible as we are using Creditcard.

**ELSE Condition:**

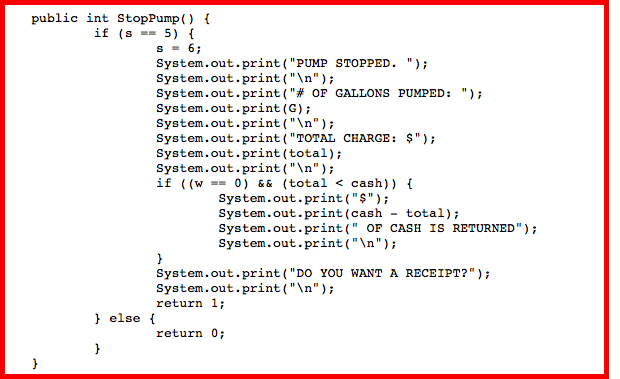
|  |  |  |
| --- | --- | --- |
| **PumpGallon** | | |
| **(w==0)&&** | **total<cash** | **Covered in TestCase#** |
| TRUE | TRUE | Test#8 |
| TRUE | FALSE | Test#16 |
| FALSE | TRUE | **NOT POSSIBLE** |
| FALSE | FALSE | **NOT POSSIBLE** |

**Test#16**: Activate(4 ,5) ,Start() ,PayCash(10) ,Super() ,StartPump() ,PumpGallon() ,PumpGallon(), PumpGallon() ,Receipt() ,TurnOff()

**Reason for Not Possible:**

1. **w==0[FALSE]&&total<cash [TRUE]**🡪 In this statement, w==0 is false, that means the payment is done by Creditcard and not cash, so second condition can never be true.
2. **w==0[FALSE]&& total<cash [FALSE]**🡪 In this statement also, w==0 is false, that means the payment is done by Creditcard and not cash, so second condition can never be true.

xii. StopPump()



**IF Condition:**

|  |  |
| --- | --- |
| **StopPump** | |
| **s==5** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

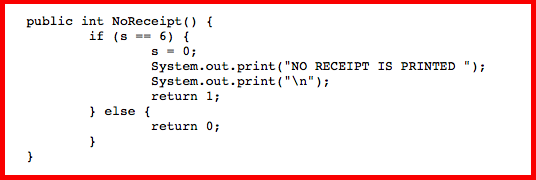
**IF Inside If Condition:**

|  |  |  |
| --- | --- | --- |
| **StopPump(When s==5 is True)** | | |
| **(w==0)&&** | **total<cash** | **Covered in TestCase#** |
| TRUE | TRUE | Test#2 |
| TRUE | FALSE | Test#1 |
| FALSE | TRUE | **NOT POSSIBLE** |
| FALSE | FALSE | Test#9 |

**Reason for Not Possible:**

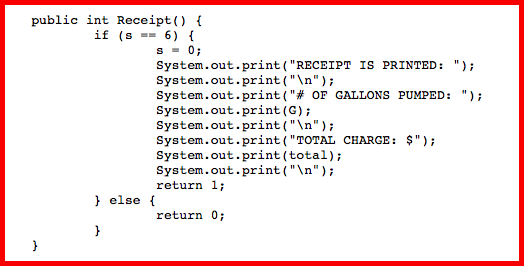
1. **w==0[FALSE]&&total<cash [TRUE]**🡪 In this statement, w==0 is false, that means the payment is done by Creditcard and not cash, so second condition can never be true.

xiii. NoReceipt()



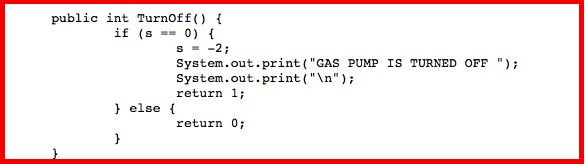
|  |  |
| --- | --- |
| **NoReceipt** | |
| **s==6** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

xiv. Receipt()



|  |  |
| --- | --- |
| **Receipt** | |
| **s==6** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

xv. TurnOff()



|  |  |
| --- | --- |
| **TurnOff** | |
| **s==0** | **Covered in TestCase#** |
| TRUE | Test#1 |
| FALSE | Test#14 |

# 5. Test Suit(TS.txt file) and the results after execution

## Test Suite (TS.txt file):

**Test#1: Activate 4 5 Start PayCredit Approved Regular StartPump PumpGallon StopPump Receipt Start PayCash 10 Super StartPump PumpGallon PumpGallon StopPump NoReceipt TurnOff**

**Test#2: Activate 4 7 Start PayCredit Reject Start PayCash 10 Cancel Start PayCash 10 Regular StartPump StopPump Receipt TurnOff**

**Test#3: Activate 4 7 Start PayCredit Approved Cancel TurnOff**

**Test#4: Activate 4 7 Start PayCredit Approved Super StartPump PumpGallon StopPump NoReceipt TurnOff**

**Test#5: Activate 4 7 Start PayCredit Reject TurnOff**

**Test#6: Activate 4 7 TurnOff**

**Test#7: Activate 4 7 Start PayCash 3 Super StartPump PumpGallon NoReceipt TurnOff**

**Test#8: Activate 4 6 Start PayCash 6 Regular StartPump PumpGallon PumpGallon Receipt TurnOff**

**Test#9: Activate 4 5 Start PayCredit Approved Regular StartPump PumpGallon StopPump NoReceipt Start PayCredit Reject TurnOff**

**Test#10: Activate 4 5 Activate 4 5 PayCredit Approved Cancel PayCash 10 Regular Super StartPump PumpGallon PumpGallon StopPump NoReceipt Receipt Reject Start Activate 4 5 Start Approved Reject Cancel Regular Super StartPump PumpGallon PumpGallon StopPump Receipt NoReceipt TurnOff PayCredit Reject TurnOff**

**Test#11: Activate 4 7 Start PayCredit Activate 4 5 Start PayCredit Cancel PayCash 10 Regular Super StartPump PumpGallon StopPump PumpGallon Receipt NoReceipt TurnOff Approved Activate 4 7 Start PayCredit Reject Approved PayCash 10 StartPump PumpGallon StopPump PumpGallon Receipt NoReceipt TurnOff Super Activate 4 7 Start PayCredit Reject Approved Cancel PayCash -10 Regular Super PumpGallon StopPump PumpGallon Receipt NoReceipt TurnOff StartPump PumpGallon StopPump NoReceipt TurnOff**

**Test#12: Activate 4 7 Start PayCredit Approved Regular StartPump Activate 4 7 Start PayCredit Reject Approved Cancel PayCash 10 Regular Super StartPump Receipt NoReceipt TurnOff StopPump Activate 4 7 Start PayCredit Reject Approved Cancel PayCash 10 Regular Super StartPump StopPump PumpGallon TurnOff PumpGallon NoReceipt TurnOff**

**Test#13: Activate 4 -4 Activate -3 2 Activate -3 -2 Activate 4 5 Start PayCredit Activate 4 -4 Activate -3 5 Activate -3 -4 Reject TurnOff**

**Test#14: Activate 4 -7 Start PayCredit Reject Cancel Approved Regular Super StartPump PumpGallon StopPump NoReceipt Receipt TurnOff Activate 4 7 TurnOff**

**Test#15: Activate 4 7 Start PayCash -10 PayCash 10 Cancel TurnOff**

**Test#16: Activate 4 5 Start PayCash 10 Super StartPump PumpGallon PumpGallon PumpGallon Receipt TurnOff**

**$$**

## Results :

**Transition Pair(2-Transition)Test Cases**

**Test#1:** Activate(4 ,5) ,Start() ,PayCredit() ,Approved() ,Regular() ,StartPump() ,PumpGallon() ,StopPump(), Receipt() ,Start() ,PayCash(10) ,Super() ,StartPump(),PumpGallon() ,PumpGallon() ,StopPump() ,NoReceipt(), TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

5

CREDIT CARD APPROVED.

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

8

REGULAR IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

CONTINUE PUMPING

Result=1

Enter your Choice:

12

PUMP STOPPED.

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

14

RECEIPT IS PRINTED:

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

6

Please enter the cash amount

10

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

9

SUPER IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $5

CONTINUE PUMPING

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 2

TOTAL CHARGE: $10

CONTINUE PUMPING

Result=1

Enter your Choice:

12

PUMP STOPPED.

# OF GALLONS PUMPED: 2

TOTAL CHARGE: $10

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

13

NO RECEIPT IS PRINTED

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 1 minute 8 seconds)

**TEST CASE PASSED**

**Test#2:** Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Start() ,PayCash(10) ,Cancel() ,Start() ,PayCash(10), Regular() ,StartPump() ,StopPump() ,Receipt() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

4

CREDIT CARD IS REJECTED.

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

6

Please enter the cash amount

10

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

7

TRANSACTION IS CANCELLED.

$10 OF CASH IS RETURNED

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

6

Please enter the cash amount

10

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

8

REGULAR IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

12

PUMP STOPPED.

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

$10 OF CASH IS RETURNED

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

14

RECEIPT IS PRINTED:

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 1 minute 23 seconds)

**TEST CASE PASSED**

**Test#3:** Activate(4 ,7) ,Start ,PayCredit() ,Approved() ,Cancel() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

5

CREDIT CARD APPROVED.

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

7

TRANSACTION IS CANCELLED.

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 25 seconds)

**TEST CASE PASSED**

**Test#4:** Activate(4 ,7) ,Start() ,PayCredit() ,Approved() ,Super() ,StartPump() ,PumpGallon() ,StopPump(), NoReceipt() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

5

CREDIT CARD APPROVED.

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

9

SUPER IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $7

CONTINUE PUMPING

Result=1

Enter your Choice:

12

PUMP STOPPED.

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $7

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

13

NO RECEIPT IS PRINTED

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 36 seconds)

**TEST CASE PASSED**

**Test#5:** Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

4

CREDIT CARD IS REJECTED.

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 13 seconds)

**TEST CASE PASSED**

**Test#6:** Activate(4 ,7) ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 8 seconds)

**TEST CASE PASSED**

**Test#7:** Activate(4 ,7) ,Start() ,PayCash(3) ,Super() ,StartPump() ,PumpGallon() ,NoReceipt() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

6

Please enter the cash amount

3

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

9

SUPER IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

PUMP STOPPED. NOT SUFFICIENT FUNDS.

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

$3 OF CASH IS RETURNED

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

13

NO RECEIPT IS PRINTED

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 31 seconds)

**TEST CASE PASSED**

**Test#8:** Activate(4 ,6) ,Start() ,PayCash(6) ,Regular() ,StartPump() ,PumpGallon() ,PumpGallon() ,Receipt() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

6

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

6

Please enter the cash amount

6

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

8

REGULAR IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

CONTINUE PUMPING

Result=1

Enter your Choice:

11

PUMP STOPPED. NOT SUFFICIENT FUNDS.

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

$2 OF CASH IS RETURNED

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

14

RECEIPT IS PRINTED:

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 22 seconds)

**TEST CASE PASSED**

**Test#9:** Activate(4 ,5) ,Start() ,PayCredit() ,Approved() ,Regular() ,StartPump() ,PumpGallon() ,StopPump() ,NoReceipt() ,Start() ,PayCredit() ,Reject() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

5

CREDIT CARD APPROVED.

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

8

REGULAR IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

CONTINUE PUMPING

Result=1

Enter your Choice:

12

PUMP STOPPED.

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $4

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

13

NO RECEIPT IS PRINTED

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

4

CREDIT CARD IS REJECTED.

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 32 seconds)

**TEST CASE PASSED**

**Ghost Transition Test Cases**

**Test#10:** Activate(4 ,5) ,Activate(4 ,5) ,PayCredit() ,Approved() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,PumpGallon() ,PumpGallon() ,StopPump() ,NoReceipt() ,Receipt() ,Reject() ,Start() ,Activate(4 ,5) ,Start() ,Approved() ,Reject() ,Cancel() ,Regular() ,Super() ,StartPump() ,PumpGallon() ,PumpGallon(), StopPump() ,Receipt() ,NoReceipt() ,TurnOff() ,PayCredit() ,Reject() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

GAS PUMP IS ON

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

Result=0

Enter your Choice:

3

Result=0

Enter your Choice:

5

Result=0

Enter your Choice:

7

Result=0

Enter your Choice:

6

Please enter the cash amount

10

Result=0

Enter your Choice:

8

Result=0

Enter your Choice:

9

Result=0

Enter your Choice:

10

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

12

Result=0

Enter your Choice:

13

Result=0

Enter your Choice:

14

Result=0

Enter your Choice:

4

Result=0

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

Result=0

Enter your Choice:

2

Result=0

Enter your Choice:

5

Result=0

Enter your Choice:

4

Result=0

Enter your Choice:

7

Result=0

Enter your Choice:

8

Result=0

Enter your Choice:

9

Result=0

Enter your Choice:

10

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

12

Result=0

Enter your Choice:

14

Result=0

Enter your Choice:

13

Result=0

Enter your Choice:

15

Result=0

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

4

CREDIT CARD IS REJECTED.

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 1 minute 58 seconds)

**TEST CASE PASSED**

**Test#11:** Activate(4 ,7) ,Start() ,PayCredit() ,Activate(4 ,5) ,Start() ,PayCredit() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,PumpGallon() ,StopPump() ,PumpGallon() ,Receipt() ,NoReceipt() ,TurnOff() ,Approved() ,Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Approved() ,PayCash(10) ,StartPump(), PumpGallon() ,StopPump() ,PumpGallon() ,Receipt() ,NoReceipt() ,TurnOff() ,Super() ,Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Approved() ,Cancel() ,PayCash(-10) ,Regular() ,Super() ,PumpGallon() ,StopPump(), PumpGallon() ,Receipt() ,NoReceipt() ,TurnOff() ,StartPump() ,PumpGallon() ,StopPump() ,NoReceipt() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

Result=0

Enter your Choice:

2

Result=0

Enter your Choice:

3

Result=0

Enter your Choice:

7

Result=0

Enter your Choice:

6

Please enter the cash amount

10

Result=0

Enter your Choice:

8

Result=0

Enter your Choice:

9

Result=0

Enter your Choice:

10

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

12

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

14

Result=0

Enter your Choice:

13

Result=0

Enter your Choice:

15

Result=0

Enter your Choice:

5

CREDIT CARD APPROVED.

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

Result=0

Enter your Choice:

2

Result=0

Enter your Choice:

3

Result=0

Enter your Choice:

4

Result=0

Enter your Choice:

5

Result=0

Enter your Choice:

6

Please enter the cash amount

10

Result=0

Enter your Choice:

10

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

12

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

14

Result=0

Enter your Choice:

13

Result=0

Enter your Choice:

15

Result=0

Enter your Choice:

9

SUPER IS SELECTED.

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

Result=0

Enter your Choice:

2

Result=0

Enter your Choice:

3

Result=0

Enter your Choice:

4

Result=0

Enter your Choice:

5

Result=0

Enter your Choice:

7

Result=0

Enter your Choice:

6

Please enter the cash amount

-10

Result=0

Enter your Choice:

8

Result=0

Enter your Choice:

9

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

12

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

14

Result=0

Enter your Choice:

13

Result=0

Enter your Choice:

15

Result=0

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $7

CONTINUE PUMPING

Result=1

Enter your Choice:

12

PUMP STOPPED.

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $7

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

13

NO RECEIPT IS PRINTED

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 2 minutes 3 seconds)

**TEST CASE PASSED**

**Test#12:** Activate(4 ,7) ,Start() ,PayCredit() ,Approved(),Regular() ,StartPump() ,Activate(4 ,7) ,Start(), PayCredit() ,Reject() ,Approved() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,Receipt(), NoReceipt() ,TurnOff() , StopPump() ,Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Approved() ,Cancel() ,PayCash(10) ,Regular() ,Super() ,StartPump() ,StopPump() ,PumpGallon() ,TurnOff() , PumpGallon() ,NoReceipt() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

5

CREDIT CARD APPROVED.

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

8

REGULAR IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

Result=0

Enter your Choice:

2

Result=0

Enter your Choice:

3

Result=0

Enter your Choice:

4

Result=0

Enter your Choice:

5

Result=0

Enter your Choice:

7

Result=0

Enter your Choice:

6

Please enter the cash amount

10

Result=0

Enter your Choice:

8

Result=0

Enter your Choice:

9

Result=0

Enter your Choice:

10

Result=0

Enter your Choice:

14

Result=0

Enter your Choice:

13

Result=0

Enter your Choice:

15

Result=0

Enter your Choice:

12

PUMP STOPPED.

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

Result=0

Enter your Choice:

2

Result=0

Enter your Choice:

3

Result=0

Enter your Choice:

4

Result=0

Enter your Choice:

5

Result=0

Enter your Choice:

7

Result=0

Enter your Choice:

6

Please enter the cash amount

10

Result=0

Enter your Choice:

8

Result=0

Enter your Choice:

9

Result=0

Enter your Choice:

10

Result=0

Enter your Choice:

12

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

15

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

13

NO RECEIPT IS PRINTED

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 1 minute 43 seconds)

**TEST CASE PASSED**

**Multiple Condition and Branch Test Cases**

**Test#13:** Activate(4 ,-4) ,Activate(-3 ,2) ,Activate(-3 ,-2) ,Activate(4 ,5) ,Start() ,PayCredit() ,Activate(4 ,-4) ,Activate(-3 ,5) ,Activate(-3 ,-4) ,Reject() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

-4

Result=0

Enter your Choice:

1

Enter the price for Regular fuel:

-3

Enter the price for Super fuel:

2

Result=0

Enter your Choice:

1

Enter the price for Regular fuel:

-3

Enter the price for Super fuel:

-2

Result=0

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

3

CHECKING CREDIT CARD.

Result=1

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

-4

Result=0

Enter your Choice:

1

Enter the price for Regular fuel:

-3

Enter the price for Super fuel:

5

Result=0

Enter your Choice:

1

Enter the price for Regular fuel:

-3

Enter the price for Super fuel:

-4

Result=0

Enter your Choice:

4

CREDIT CARD IS REJECTED.

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 51 seconds)

**TEST CASE PASSED**

**Test#14:** Activate(4 ,-7) ,Start() ,PayCredit() ,Reject() ,Cancel() ,Approved() ,Regular() ,Super() ,StartPump(), PumpGallon() ,StopPump() ,NoReceipt() ,Receipt() ,TurnOff() ,Activate(4 ,7) ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

-7

Result=0

Enter your Choice:

2

Result=0

Enter your Choice:

3

Result=0

Enter your Choice:

4

Result=0

Enter your Choice:

7

Result=0

Enter your Choice:

5

Result=0

Enter your Choice:

8

Result=0

Enter your Choice:

9

Result=0

Enter your Choice:

10

Result=0

Enter your Choice:

11

Result=0

Enter your Choice:

12

Result=0

Enter your Choice:

13

Result=0

Enter your Choice:

14

Result=0

Enter your Choice:

15

Result=0

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 36 seconds)

**TEST CASE PASSED**

**Test#15:** Activate(4 ,7) ,Start() ,PayCash(-10) ,PayCash(10) ,Cancel() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

7

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

6

Please enter the cash amount

-10

Result=0

Enter your Choice:

6

Please enter the cash amount

10

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

7

TRANSACTION IS CANCELLED.

$10 OF CASH IS RETURNED

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 20 seconds)

**TEST CASE PASSED**

**Test#16:** Activate(4 ,5) ,Start() ,PayCash(10) ,Super() ,StartPump() ,PumpGallon() ,PumpGallon(), PumpGallon() ,Receipt() ,TurnOff()

**Result:**

run:

Software testing and Analsysis(CS 589)-Project

----------------------------------------------

Prof: Dr. Bogdan Korel

Test Driver for Gas Pump Implementation...

Enter your choice

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

13.NoReceipt

14.Receipt

15.TurnOff

Enter your Choice:

1

Enter the price for Regular fuel:

4

Enter the price for Super fuel:

5

GAS PUMP IS ON

Result=1

Enter your Choice:

2

WELCOME!!!

DO YOU WANT TO PAY BY CASH OR CREDIT CARD?

Result=1

Enter your Choice:

6

Please enter the cash amount

10

SELECT TYPE OF GASOLINE:

a. REGULAR

b. SUPER

Result=1

Enter your Choice:

9

SUPER IS SELECTED.

Result=1

Enter your Choice:

10

PUMP IS READY TO DISPOSE

# OF GALLONS PUMPED: 0

TOTAL CHARGE: $0

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 1

TOTAL CHARGE: $5

CONTINUE PUMPING

Result=1

Enter your Choice:

11

# OF GALLONS PUMPED: 2

TOTAL CHARGE: $10

CONTINUE PUMPING

Result=1

Enter your Choice:

11

PUMP STOPPED. NOT SUFFICIENT FUNDS.

# OF GALLONS PUMPED: 2

TOTAL CHARGE: $10

DO YOU WANT A RECEIPT?

Result=1

Enter your Choice:

14

RECEIPT IS PRINTED:

# OF GALLONS PUMPED: 2

TOTAL CHARGE: $10

Result=1

Enter your Choice:

15

GAS PUMP IS TURNED OFF

Result=1

Enter your Choice:

Exit successfully from the program.

BUILD SUCCESSFUL (total time: 27 seconds)

**TEST CASE PASSED**

# 6. Conclusion

This report fulfills all the requirements and goals. This report contains :

1. All test cases that satisfies 2 transition sequences and the testcase numbers which satisfies that transition .
2. All test cases that are identified and that satisfies default/ghost transition .
3. All test cases that satisfies all multiple condition/branches in the implementation (source-code) of the *GasPump* class.
4. All test cases for the predicates containing only a simple condition(considered as branch testing).
5. All the cases that are not executable and the detailed reason is provided after every such case.
6. The results produced by the *GasPump* class are validated and documented for each and every test case and the results produced are correct.
7. The test suite(TS.txt) file and the results produced after its execution.

**The source code used is “JAVA” and the executable file created is “JAR” file and it has been converted to “.exe”**.

There are total of 16 test cases that covers all methods of testing required for this project. There were 11 cases which are “Not Possible” and the detailed reasons are provided.

The test driver, was written into the source code provided to us to run the test cases, thus checking all the test cases manually. All the test cases were written successfully.

As per my understanding of the project, it took a lot of effort in manually executing each and every test case which was a tedious and time-consuming task. I feel that we can automate the process of executing each test case which can save a lot of time.

Though the execution process is time consuming , this would be the best solution to detect most of the defects in a program.

# 7. Well documented source code(implemented) of the GasPump class and the test driver

import java.util.\*;

import java.io.\*;

public class GasPump {

private int Rprice;

private int Sprice;

private int w;

private int price;

private int G;

private int total;

private int cash;

private int s;

public GasPump() {

Rprice = 0;

Sprice = 0;

w = 0;

price = 0;

G = 0;

total = 0;

cash = 0;

s = -1;

}

**public int Activate(int a, int b)** {

if ((s == -1) && (a > 0) && (b > 0)) {

s = 0;

Rprice = a;

Sprice = b;

System.out.print("GAS PUMP IS ON");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int Start()** {

if (s == 0) {

s = 1;

System.out.print("WELCOME!!!");

System.out.print("\n");

System.out.print("DO YOU WANT TO PAY BY CASH OR CREDIT CARD?");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int PayCredit()** {

if (s == 1) {

s = 2;

System.out.print("CHECKING CREDIT CARD.");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int Reject()** {

if (s == 2) {

s = 0;

System.out.print("CREDIT CARD IS REJECTED.");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int Cancel()** {

if (s == 3) {

s = 0;

System.out.print("TRANSACTION IS CANCELLED.");

System.out.print("\n");

if (w == 0) {

System.out.print("$");

System.out.print(cash);

System.out.print(" OF CASH IS RETURNED");

System.out.print("\n");

}

return 1;

} else {

return 0;

}

}

**public int Approved()** {

if (s == 2) {

s = 3;

w = 1;

System.out.print("CREDIT CARD APPROVED.");

System.out.print("\n");

System.out.print("SELECT TYPE OF GASOLINE:");

System.out.print("\n");

System.out.print("a. REGULAR");

System.out.print("\n");

System.out.print("b. SUPER");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int PayCash(int c)** {

if ((s == 1) && (c > 0)) {

s = 3;

w = 0;

cash = c;

System.out.print("SELECT TYPE OF GASOLINE:");

System.out.print("\n");

System.out.print("a. REGULAR");

System.out.print("\n");

System.out.print("b. SUPER");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int Regular()** {

if (s == 3) {

s = 4;

System.out.print("REGULAR IS SELECTED.");

System.out.print("\n");

price = Rprice;

return 1;

} else {

return 0;

}

}

**public int Super()** {

if (s == 3) {

s = 4;

System.out.print("SUPER IS SELECTED.");

System.out.print("\n");

price = Sprice;

return 1;

} else {

return 0;

}

}

**public int StartPump()** {

if (s == 4) {

s = 5;

G = 0;

total = 0;

System.out.print("PUMP IS READY TO DISPOSE ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int PumpGallon()** {

if (s == 5) {

if ((w == 1) || ((cash >= price \* (G + 1)) && (w == 0))) {

G = G + 1;

total = G \* price;

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

System.out.print("CONTINUE PUMPING");

System.out.print("\n");

return 1;

} else if ((w == 0) && (cash < price \* (G + 1))) {

s = 6;

System.out.print("PUMP STOPPED. NOT SUFFICIENT FUNDS. ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

if ((w == 0) && (total < cash)) {

System.out.print("$");

System.out.print(cash - total);

System.out.print(" OF CASH IS RETURNED");

System.out.print("\n");

}

System.out.print("DO YOU WANT A RECEIPT?");

System.out.print("\n");

return 1;

}

;

}

;

return 0;

}

**public int StopPump()** {

if (s == 5) {

s = 6;

System.out.print("PUMP STOPPED. ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

if ((w == 0) && (total < cash)) {

System.out.print("$");

System.out.print(cash - total);

System.out.print(" OF CASH IS RETURNED");

System.out.print("\n");

}

System.out.print("DO YOU WANT A RECEIPT?");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int NoReceipt()** {

if (s == 6) {

s = 0;

System.out.print("NO RECEIPT IS PRINTED ");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int Receipt()** {

if (s == 6) {

s = 0;

System.out.print("RECEIPT IS PRINTED: ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**public int TurnOff()** {

if (s == 0) {

s = -2;

System.out.print("GAS PUMP IS TURNED OFF ");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DRIVER FOR GAS PUMP CLASS**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void main(String[] args)

{

try

{

GasPump gas=new GasPump();

BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

System.out.println("Software testing and Analsysis(CS 589)-Project");

System.out.println("----------------------------------------------");

System.out.println(" Prof: Dr. Bogdan Korel");

System.out.println(" ");

System.out.println("Test Driver for Gas Pump Implementation...");

System.out.println("Enter your choice");

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

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

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

System.out.println("4.Reject");

System.out.println("5.Approved");

System.out.println("6.PayCash");

System.out.println("7.Cancel");

System.out.println("8.Regular");

System.out.println("9.Super");

System.out.println("10.StartPump");

System.out.println("11.PumpGallon");

System.out.println("12.StopPump");

System.out.println("13.NoReceipt");

System.out.println("14.Receipt");

System.out.println("15.TurnOff");

int a,b,c,output;

while(true)

{

System.out.println("Enter your Choice:");

String str = br.readLine();

int ch = Integer.parseInt(str);

switch(ch)

{

case 1:

System.out.println("Enter the price for Regular fuel:");

a = Integer.parseInt(br.readLine());

System.out.println("Enter the price for Super fuel:");

b = Integer.parseInt(br.readLine());

output=gas.Activate(a,b);

System.out.println("Result="+output);

break;

case 2:

output=gas.Start();

System.out.println("Result="+output);

break;

case 3:

output=gas.PayCredit();

System.out.println("Result="+output);

break;

case 4:

output=gas.Reject();

System.out.println("Result="+output);

break;

case 5:

output=gas.Approved();

System.out.println("Result="+output);

break;

case 6:

System.out.println("Please enter the cash amount");

c = Integer.parseInt(br.readLine());

output=gas.PayCash(c);

System.out.println("Result="+output);

break;

case 7:

output=gas.Cancel();

System.out.println("Result="+output);

break;

case 8:

output=gas.Regular();

System.out.println("Result="+output);

break;

case 9:

output=gas.Super();

System.out.println("Result="+output);

break;

case 10:

output=gas.StartPump();

System.out.println("Result="+output);

break;

case 11:

output=gas.PumpGallon();

System.out.println("Result="+output);

break;

case 12:

output=gas.StopPump();

System.out.println("Result="+output);

break;

case 13:

output=gas.NoReceipt();

System.out.println("Result="+output);

break;

case 14:

output=gas.Receipt();

System.out.println("Result="+output);

break;

case 15:

output=gas.TurnOff();

System.out.println("Result="+output);

break;

}

}

}

catch(NumberFormatException e)

{

System.out.println(

"**Exit successfully from the program.**");

}

catch(IOException e)

{

}

}

}

# 8. Detailed Instruction for executing the code in CD

**Please follow the below steps for executing the code.**

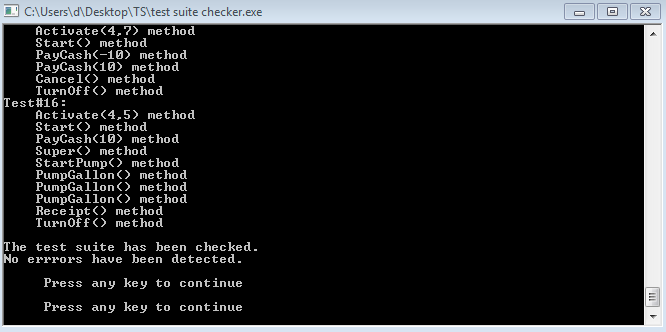
**NOTE: Source code used is JAVA, so JDK must be installed on the system to run this “.exe”.**

**Step1:** Open the folder “**KUMAR\_SINGH\_DHIRENDRA\_PROJECT”**

**Step2:** Double click on “**test suite checker.exe**” to check if there is any error in ‘**TS.txt’** file

**a:** Press any key to execute.

**b:** After pressing any key to execute the checker , the code will show no errors.



**Step3:** Double click on “**GasPump.exe**”

**Step4:** Select your choice by entering the number.

1.Activate

2.Start

3.PayCredit

4.Reject

5.Approved

6.PayCash

7.Cancel

8.Regular

9.Super

10.StartPump

11.PumpGallon

12.StopPump

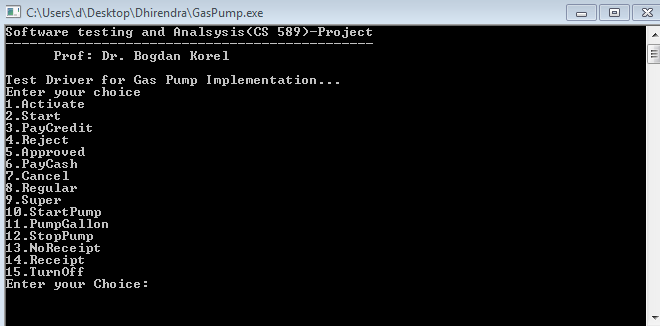
13.NoReceipt

14.Receipt

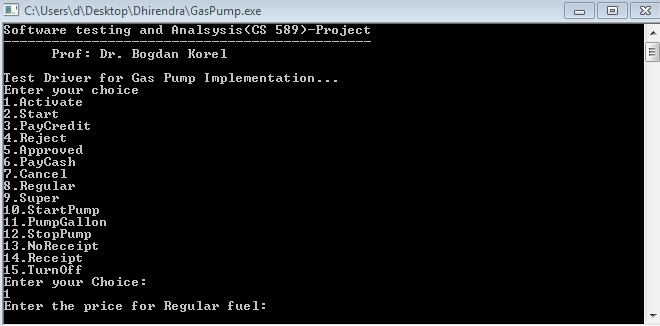
15.TurnOff

Let us take the example of executing the first test case(Test#2). Below are the screenshots taken after each step.

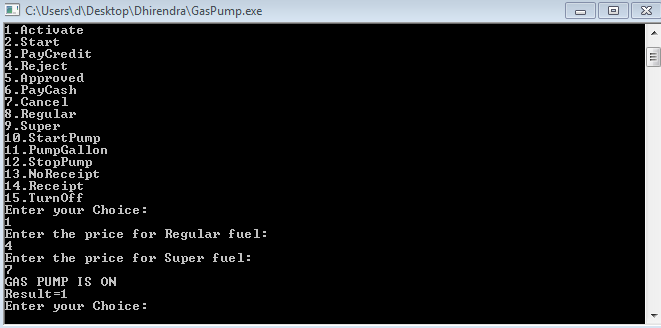
**Test#2**: Activate(4 ,7) ,Start() ,PayCredit() ,Reject() ,Start() ,PayCash(10) ,Cancel() ,Start() ,PayCash(10), Regular() ,StartPump() ,StopPump() ,Receipt() ,TurnOff()



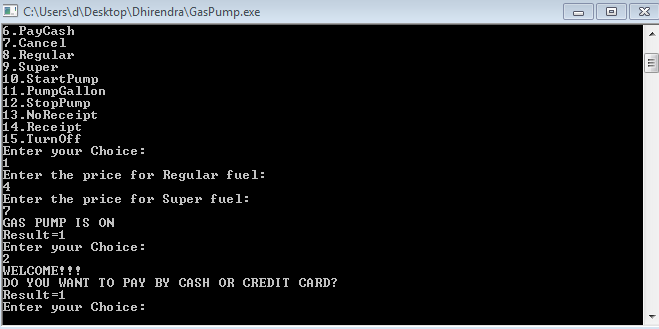
**Step 4a:** Enter your choice(i.e. Activate)



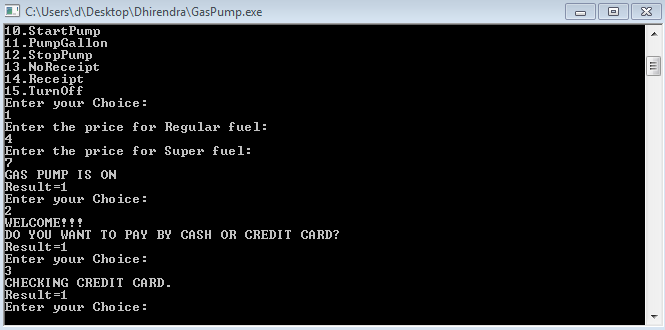
**Step4b:** Now, enter the price for “Regular” and “Super” fuel.



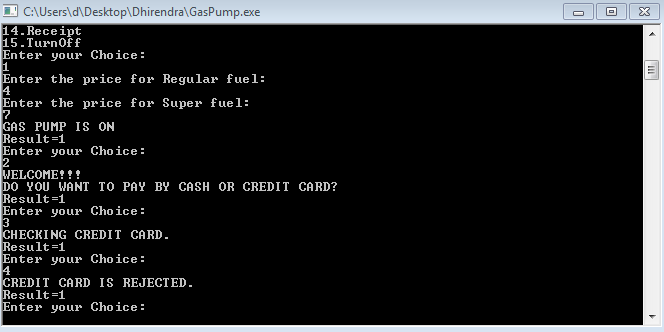
**Step4c:** Now, again Enter your choice(in this Test case “Start”)



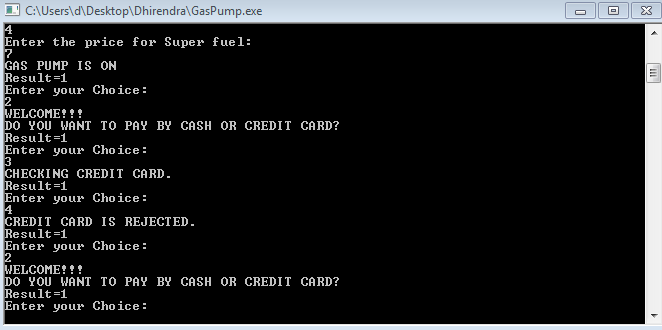
Step4d: Now, again Enter your choice(in this Test case “PayCredit”)



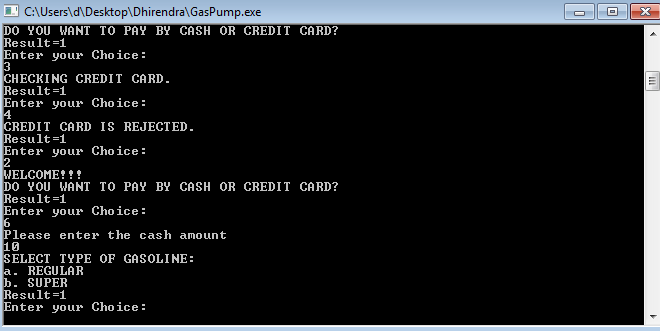
**Step4e:** Now, again Enter your choice(in this Test case “Reject”)



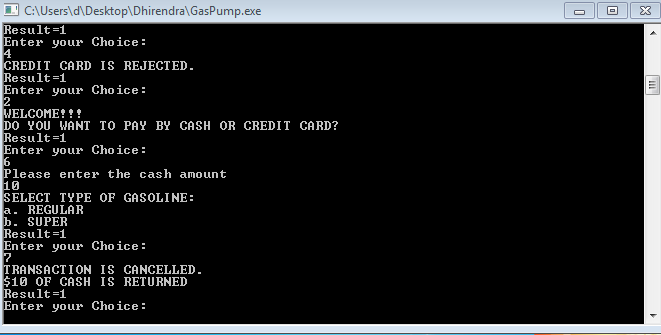
**Step4f:** Now, again Enter your choice(in this Test case “Start”)



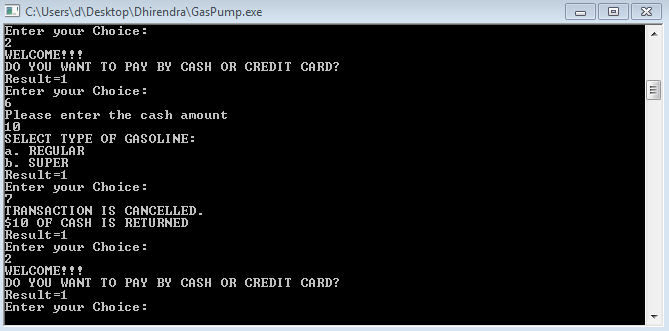
**Step4g:** Now, again Enter your choice(in this Test case “PayCash(10)”). Then Enter the cash amount.



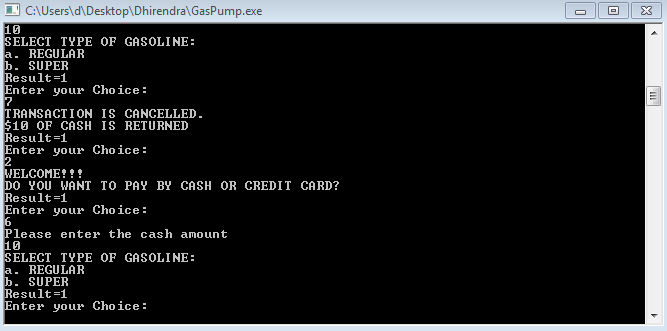
**Step4h:** Now, again Enter your choice(in this Test case “Cancel”).



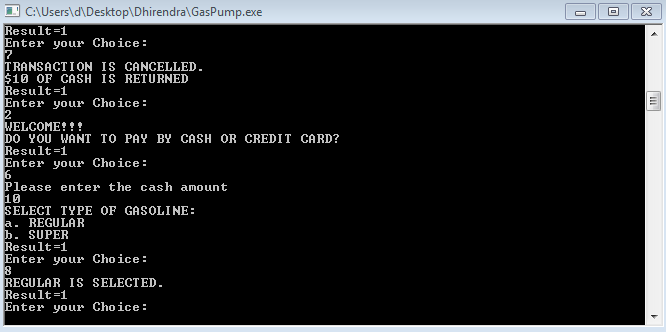
**Step4i:** Now, again Enter your choice(in this Test case “Start”).



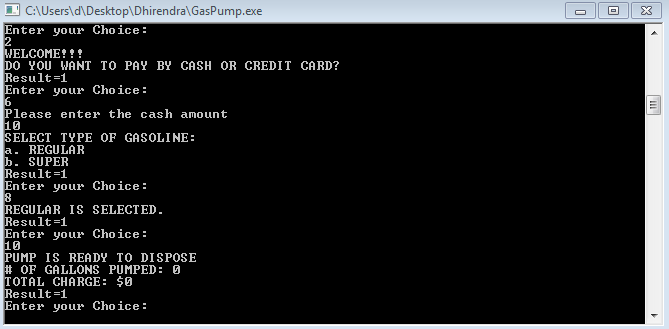
**Step4j:** Now, again Enter your choice(in this Test case “PayCash(10)”). Then Enter the cash amount.



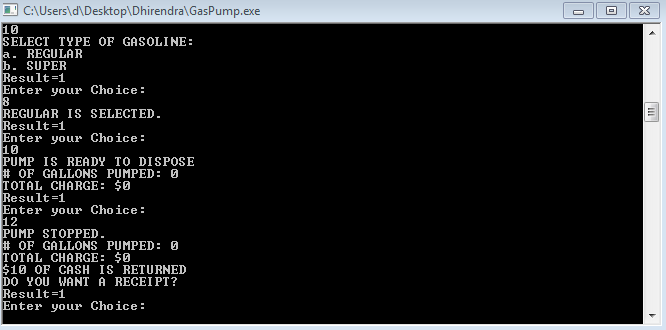
**Step4k:** Now, again Enter your choice(in this Test case “Regular”).



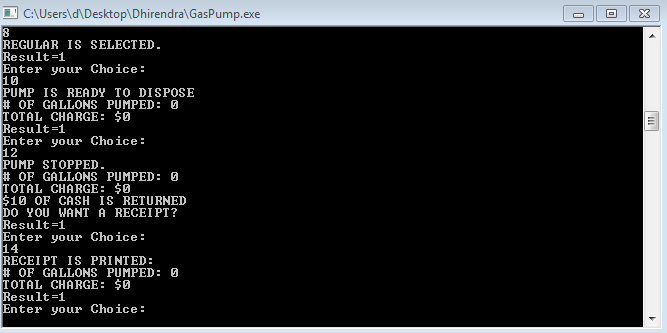
**Step4l:** Now, again Enter your choice(in this Test case “StartPump”).



**Step4m:** Now, again Enter your choice(in this Test case “StopPump”).



**Step4n:** Now, again Enter your choice(in this Test case “Receipt”).



**Step4o:** Now, again Enter your choice(in this Test case “Turnoff”).

