1. **Code**

**Truck.cpp**

#include "Truck.h"

Truck::Truck()

{

dieselTypeStatus = false;

}

void Truck::setDieselTypeStatus(bool thisStatus)

{

dieselTypeStatus = thisStatus;

}

bool Truck::getDieselTypeStatus()

{

return dieselTypeStatus;

}

**Truck.h**

#pragma once

#ifndef TRUCK\_H

#define TRUCK\_H

class Truck

{

public:

Truck();

void setDieselTypeStatus(bool thisStatus);

bool getDieselTypeStatus();

private:

bool dieselTypeStatus;

};

#endif // !TRUCK\_H

**Stub\_main\_TruckTest.cpp**

#include "Truck.h"

int main()

{

Truck myTruck;

int tempAge;

int tempStatus;

float tempPrice;

do {

cout << "Type 1 if it IS a diesel-type and type 0 if it IS NOT a diesel-type: ";

cin >> tempStatus;

if (tempStatus < 0 || tempStatus > 1)

{

cout << endl << "Invalid input. Please try again with number 1 or number 0" << endl;

}

} while (tempStatus < 0 || tempStatus > 1);

myTruck.setDieselTypeStatus(tempStatus);

cout << endl << "Diesel Type Status stored" << endl;

if (myTruck.getDieselTypeStatus())

{

cout << "Your truck, Ford F-150, IS a diesel-type" << endl;

}

else {

cout << "Your truck, Ford F-150, IS NOT a diesel-type" << endl;

}

return 0;

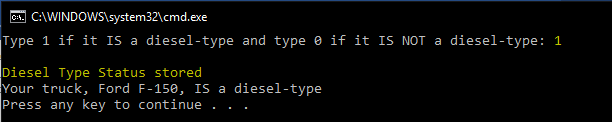
}

1. **Test Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Strategy | Test Number | Description | Input | Expected Output | Actual Output | Pass/Fail |
| Valid | 1 | User enters corresponding number for choosing either true or false for diesel type status | User enters “1” for true for diesel type status | “Diesel Type Status stored” | “Diesel Type Status stored” | Pass |
| Invalid | 1 | User enters number not corresponding to choosing either true or false for diesel type status | User enters “10” for true for diesel type status | “Invalid input. Please try again with number 1 or number 0” | “Invalid input. Please try again with number 1 or number 0” | Pass |

1. **Screenshots**

Valid Test Case 1:



Invalid Test Case 1:

