////////////////////////////

///// Base Tests /////

////////////////////////////

Testing Base Default ctor

**// 1.First line is printing the cout**

Vehicle #0: Default-ctor

**// 2. Constructor class created and calls the default constructor, and assigning it to vin 0, and increments s\_idgen by one.**

Testing Base insertion operator

**//3. Printing cout**

Vehicle #0 @ 1.4013e-45 0 0

**// 4. Insertion operator takes class object v1 and prints out the default constructor, which was pointed to v1**

Base idgen: 1

**//5. Prints out idgen**

Testing Base Parametrized ctor

**//6. Printing cout**

Vehicle #99Parametized-ctor

**//7. Calling Parametized constructor, and creating a class object v99 and assigns the parametized values to that class object**

Vehicle #99 @ 39.54 119.82 4500

**// 8. Prints out v99**

Base idgen: 1

**// 9. Prints out idgen, doesn’t increment because of ternary operator(*a <= s\_idgen ? s\_idgen++ : a*) vin is less than 99, so it does not increment.**

Testing Base Copy ctor

**//10. Printing cout**

Vehicle #1 :Copy-ctor

**//11. Calling Copy constructor, and creating a class object for it(v99\_cpy) and assigning the values from the parametized (v99) to the copy object (v99\_cpy) and increments idgen.**

Vehicle #1 @ 39.54 119.82 4500

**//12. Prints out copy constructor(v99\_cpy)**

Base idgen: 2

**//13. Prints out idgen**

Testing Base Assignment operator

**//14. Printing cout**

Vehicle #1: Assignment

**//15. Calling assignment operator, which sets the default object class(v1) equal to the copy constructor object (v99\_cpy)**

Vehicle #0 @ 39.54 119.82 4500

**//16. Prints out default constructor object(v1)**

Base idgen: 2

**//17. Prints out idgen**

Testing Base Move Function

**//18. Printing cout**

Vehicle #0 CANNOT MOVE - I DON'T KNOW HOW

**//19. Calling the move function, which does nothing except print out this message**

////////////////////////////

///// Derived Tests /////

////////////////////////////

Testing Derived Default ctor

**//20. Printing cout**

Vehicle #2: Default-ctor

Car #2: Default-ctor

**//21. Creating a Car default constructor object and calling the default constructor (c1) and assigning vin to the current idgen and incrementing idgen.**

Testing Derived insertion operator

**//22. Printing cout**

Car #2 Plates , Throttle: 0 @ -3.4171e-19 4.57664e-41 -2.87308e-19

**//23. Insertion operator takes car class and assigns the values of the default constructor to the car class object c1.**

Derived idgen: 3

**//24. Printing idgen**

Testing Derived Parametrized ctor

**//25. Printing cout**

Vehicle #999Parametized-ctor

Car #999: Parametized-Ctor

**//26. Created a class object (c999) and assigned the values of the parametized constructor of car class to the class object c99.**

Car #999 Plates Gandalf , Throttle: -1599548864 @ 39.54 119.82 4500

**//27. Prints the newly parametized constructor**

Derived idgen: 3

**//28. Prints idgen**

Testing Derived Copy ctor

**//29. Prints cout**

Vehicle #3: Default-ctor

Car # 3: Copy-ctor

**//30. Calls the copy constructor, which copies the car class object(c99) values into the new constructor object (c99\_cpy) and increments idgen**

Car #3 Plates Gandalf , Throttle: 0 @ 39.54 119.82 4500

**//31. Prints the copy constructor**

Derived idgen: 4

**//32. Prints idgen**

Testing Derived Assignment operator

**//33. Prints cout**

Car #3 Assignment

Car #2 Plates Gandalf , Throttle: 0 @ 39.54 119.82 4500

**//34. Calls the assignment operator, passing through the default car object class(c1) and setting it equal to the copy constructor object created(c99\_cpy)**

Derived idgen: 4

**//35. Prints idgen**

Testing Derived Move Function

Car #2DRIVE to destination [37.77 122.42 52] , with throttle @ 75

**//36. Calls the move function, which passes through the new LLA and assigns it to m\_lla and, once the function is called, it passes the throttle value of 75 to the drive function.**

////////////////////////////

///// Tests Done /////

////////////////////////////

Car #3 : Dtor

Vehicle #3: Dtor

Car #999 : Dtor

Vehicle #999: Dtor

Car #2 : Dtor

Vehicle #2: Dtor

Vehicle #1: Dtor

Vehicle #99: Dtor

Vehicle #0: Dtor

**//37. After the main function is completed, all vehicle objects create are destroyed**

A big problem I had while coding this project was my compiler was only allowed to compile with the usage of stdc++11 flag on my compile and my make file, although it worked perfectly fine on NoMachine. I was unable to figure out how to update my vin number on my assignment operator, as each of the numbers should have matched within the constructor call, but on my assignment operator function call, they ended up being different numbers. I was unable solve this problem with my code, but otherwise I believe everything is function correctly. It was extremely difficult to know whether or not everything printed correctly, if we had a sample output, it would’ve been easier to determine whether or not outputs were going correctly or how you wanted it. If I had more time, I would go more in depth with the constructors and try to fix the problem of the assignment operator not updating vin properly without using methods which would only work for this project and not be actually applicable. I had a method to make them the same, but decided not to use it because I felt it would be a cheap way to increment it. I would also looked more into the ternary operator created in order to try and solve the issue as to why my assignment operators are not initializing vin correctly.