Project 6 Documentation

Name: Cicelia Siu

Date: March 10, 2020

Code Zip File : PA6\_Siu\_Cicelia

Purpose of the Code:

The Purpose of this code was to be able to learn virtual functions and pure virtual functions. This code allowed to review class manipulation with const members/functions, operator overloading/overriding, and constructors. Vehicles are the base class whereas the derived is Cars.

Design:

This code uses the constructors to create Vehicles. Each vehicle has a longitude, latitude, and altitude that explains where the car is, which can be changed for cars, but not the Vehicles. Each car has a throttle that starts when the function drive or move is called. The function serialize basically sends the information into the output stream and the

Output:

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

///// Constructor Tests /////

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

Testing Derived Default ctor

Vehicle: Default-ctor

Car: Default-ctor

**//creates a Vehicle: Car called c1 with a throttle set to 0 and lla to [0,0,0]**

Testing Derived Parametrized ctor

Vehicle: Parameterized-ctor

Car: Parameterized-ctor

**//creates a Vehicle: Car called c\_rno with a throttle at 0 and lla to lla\_reno**

Testing Derived Copy ctor

Vehicle: Copy-ctor

Car: Copy-ctor

**//creates a Vehicle: Car called c\_cpy and copies the throttle and lla of c\_rno**

Testing Derived Assignment operator

Car: Assignment

**//assigns c1 with the information of c\_cpy meaning c1 should be the same as c\_rno**

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

///// Polymorphism Tests /////

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

Testing VIRTUAL Move Function for DERIVED Class Objects

Car: DRIVE to destination, with a throttle of 75

**//car c1 is set to a throttle of 75 and is moved to an lla of lla\_new**

Testing Insertion operator<< Overload for BASE Class Objects

Car: Throttle: 0 @ [39.54, 119.82, 4500]

**//car c\_rno that hasn't been moved (still has a lla of lla\_rno) is printed out**

**//sets c1’s lla to lla\_ny and c\_cpy’s lla to lla\_la**

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

///// Polymorphic Base Class Pointer Tests /////

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

Testing VIRTUAL Move Function on Base Class Pointers

Car: DRIVE to destination, with a throttle of 75

Car: DRIVE to destination, with a throttle of 75

Car: DRIVE to destination, with a throttle of 75

**// all three cars move to an lla of lla\_new and their throttle set to 75**

Testing Insertion operator<< Overload for Base Class Pointers

Car: Throttle: 75 @ [37.77, 122.42, 52]

Car: Throttle: 75 @ [37.77, 122.42, 52]

Car: Throttle: 75 @ [37.77, 122.42, 52]

**// all three cars are printed out with a throttle of 75 and their lla of lla\_new**

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

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

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

Car: Dtor

Vehicle: Dtor

Car: Dtor

Vehicle: Dtor

Car: Dtor

Vehicle: Dtor

**//all three cars are destroyed**

Problems:

* Understanding what the point of virtual functions are and how to implement it.
* Understanding what a pure virtual is compared to a regular virtual. Basically it is where Vehicle is an abstract class which doesn’t move, compared to a concrete class (Car) which moves and needs implementation for it.
* Const-correctness was the hardest part for me. I never understand where to put the const and where to not use a const.

If I were to do this again:

* I would start earlier and thoroughly understand const correctness.