Name: proj6.cpp, Car.cpp, Vehicle.cpp
Author: Dylan Wong
Class: CS 202.1001
Purpose: Demonstrate an understanding of inheritance and polymorphism by creating classes to be utilized by the driver code provided.

Vehicle (base) Class:

- The rental car class has a default, parameterized, and copy constructor, the first takes no arguments, the second takes an LLA, the third takes a Vehicle object address.
- The setters and getters do as their name implies changing and retrieving the values of the private member variables.
- Overload and insertion operators were utilized to print out the object to the screen in a readable format and to change the LLA of one Vehicle object to that of another.
- Pure virtual “Move” method which serves as a template for derived classes.
- Private variables: m_lla.

Car (derived) Class:

- The rental car class has a default, parameterized, and copy constructor, the first takes no arguments, the second takes an LLA, the third takes a Car object address.
- The setters and getters do as there name implies changing and retrieving the values of the private member variables.
- Assignment operator was utilized to change the LLA and throttle of one Car object to that of another.
- “Drive” method sets m_throttle to a given value.
- “Move” method changes a Car object's LLA property and calls Drive() at a speed of 75.
- Virtual private “serialize” method, overrides the vehicle serialize method to print the car object to the screen.
- Private variables: m_throttle.

The main function was provided and creates Car objects, Vehicle objects, and tests other class methods enumerated above.

Ways the program could be improved:

- I could have let the user have control over the throttle value when Drive()-ing the Car/Vehicle.

- The program does generate errors when compiling but that is partially because the linter I use is cranked up to 11.