Lab 08

Lab 8 continues exploring inheritance, multiple inheritance, including grandchildren, and polymorphism. Especially, polymorphism is demonstrated with the function of the same name, `display`, in two different classes, as well as demonstrating inheritance where a child class does not have a `display` function.

IMPORTANT

Include the lab number and student name in comments at the top of your cpp file.
Also include your output in comments at the bottom of your cpp cile.

INSTRUCTIONS

Define a class named Vehicle with:

- a private integer to store miles per gallon
- a getter and a setter function for the private variable
- a constructor with 1 parameter to populate the value of the private variable
    - constructor also displays that it is running
- the Vehicle class has no other functions or variables

Define a Horn class with:

- a private string to store what sound the horn makes
- a getter and a setter function for the private variable
- the Horn class has **no** constructor, and no other functions or variables

Define a Car class that is a child of both the Vehicle and Horn classes with:

- a protected string to store the model name of the car
- a public void function named `display` with no parameters to display the model, mpg, and horn sound, and assert that the car is not a truck – see sample output below
- a constructor with 3 parameters to populate the model, mpg and horn sound
    - constructor also displays that it is running
- the Car class has no other functions or variables

Define a SUV class that is a child of the Car class with:

- a public void function named `display` with no parameters to display the model, mpg, and horn sound, and assert that it is an SUV – see sample output below
- a constructor to populate the model name, mpg, and horn sound
    - constructor also displays that it is running
- the SUV class has no other functions, and no variables

Define a Compact class that is a child of the Car class with:

- a constructor to populate the model name, mpg, and horn sound
    - constructor also displays that it is running
- the Compact class has no other functions, and no variables

In the main function:

- Instantiate a SUV with the values: Jeep, 28, and Honk honk.
- Instantiate another SUV with the values: Humvee, 14, and Ahooga!
- Instantiate a Compact with the values: Civic, 36, and Beep beep.
- Instantiate another Compact with the values: Road Runner, 34, and Meep meep!
- Call the display function for each of your objects.

Note: Do not add variables or functions not listed in the above instructions.

See sample output below.

Save your cpp file as lab08.txt and submit lab08.txt on Canvas.

```
Constructor: Vehicle
Constructor: Car
Constructor: SUV

Constructor: Vehicle
Constructor: Car
Constructor: SUV

Constructor: Vehicle
Constructor: Car
Constructor: Compact

Constructor: Vehicle
Constructor: Car
Constructor: Compact

Jeep is an SUV and gets 28 mpg, and says Honk honk.
Humvee is an SUV and gets 14 mpg, and says Ahooga!
Civic is not a truck!  It gets 36 mpg, and says Beep beep.
Road Runner is not a truck!  It gets 34 mpg, and says Meep meep!
```