Program Title: Carbon Footprint System.

Objective:

Design and implement a Java program that demonstrates the principles of Abstraction, Polymorphism, Inheritance, and Encapsulation (APIE) by creating a CarbonFootprint System. The system will calculate and display the carbon footprint of various entities (such as vehicles, buildings, and bicycles), based on the fuel they consume or the energy they use.

In this program, you will use liters per kilometer (L/km) for fuel consumption, as is standard in Canada. The carbon footprint of vehicles will be calculated based on the liters of fuel consumed per kilometer and the distance traveled, while bicycles and buildings will also contribute to the overall carbon footprint.

You will define an interface, an abstract class, and concrete classes that represent different entities, and use polymorphism to calculate and display the carbon footprints.

Requirements:

1. Interface - CarbonFootprint:

- Define an interface CarbonFootprint with a method getCarbonFootprint() that will be implemented by each class. This method will be responsible for calculating and displaying the carbon footprint of an entity.

2. Abstract Class - Vehicle:

- Create an abstract class Vehicle that implements the CarbonFootprint interface.
- This class should include common attributes such as model (a string to represent the vehicle's name or model), and any other relevant properties shared by different types of vehicles.
- The Vehicle class must implement the getCarbonFootprint() method, but it can provide a default implementation or leave it to be overridden by subclasses.

3. Concrete Classes (Car, Truck, Bicycle):

- Car: Represent a car that consumes fuel. Include attributes such as fuelConsumption (in liters per kilometer), and distanceTraveled (in kilometers). Implement the getCarbonFootprint() method to calculate and display the carbon footprint based on fuel consumption and distance traveled.
- Truck: Represent a truck (similar to Car, but with different fuel consumption and characteristics). Include relevant attributes and calculate the carbon footprint using a similar approach as the Car, but tailored for the truck's characteristics (e.g., larger fuel consumption).

- Bicycle: Represent a bicycle, which does not consume fuel but can still have a carbon footprint associated with its production and maintenance. For simplicity, assume bicycles have zero carbon emissions during operation, but you may include carbon footprint calculations related to production or materials used.

4. Concrete Class - Building:
- Create a Building class that represents a building with energy consumption (e.g., heating, air conditioning, or lighting). This class should also implement the CarbonFootprint interface and calculate the carbon footprint based on the building's energy use.
- Include attributes such as square footage, fuel type (e.g., electricity, natural gas), and energy consumption.

5. Main Tester Class:
- In your Main class, create a list (ArrayList<CarbonFootprint>) that can store objects of different types, including Car, Truck, Bicycle, and Building.
- Iterate over the list and call the getCarbonFootprint() method on each object. For each object, print both the class name and the calculated carbon footprint.

6. Output:
- For each object in the ArrayList, print out the following:
- The name of the class (e.g., "Car", "Truck", "Bicycle", or "Building")
- The calculated carbon footprint of that object in an appropriate unit (e.g., pounds of CO2)

Guidelines for Implementation:

- Encapsulation: Ensure that all fields in your classes (such as model, fuelConsumption, etc.) are private, and provide appropriate getters and setters to access them.

- Polymorphism: Demonstrate polymorphism in your Main class by storing objects of different types (Car, Truck, Bicycle, Building) in the same ArrayList<CarbonFootprint> and calling the getCarbonFootprint() method on each.

- Inheritance: The Car, Truck, and Bicycle classes should inherit from the Vehicle abstract class, and they should implement the getCarbonFootprint() method according to their specific characteristics.

- Abstraction: Use the CarbonFootprint interface to abstract the common method getCarbonFootprint(). The interface defines the method signature, and each implementing class provides its own implementation.

- Class Design: Make sure to design the classes in such a way that they adhere to object-oriented principles and are easy to extend in the future.

Bonus (Optional):

- Additional Entity: You can extend the system by adding more types of entities that contribute to carbon footprints, such as Plane, Bus, or Factory. Implement the CarbonFootprint interface and provide an appropriate implementation for the getCarbonFootprint() method.

- Refinement: Refine the carbon footprint calculation formulas based on more detailed data for each entity type (e.g., specific types of fuel, energy sources, or operational factors).

Submission:

- Submit the full code for all the classes (CarbonFootprint, Vehicle, Car, Truck, Bicycle, Building, and Main).

- Ensure your code is well-commented, especially where calculations are made for the carbon footprints.

Assumption:

- Bicycle: Bicycles themselves have no emissions, assume zero carbon footprint.

- Car: Carbon footprint = fuel consumption (L/km) * distance traveled (km) * CO2 per liter. Assuming 2.3 kg of CO2 per liter of gasoline.
-
- Truck: Carbon footprint = fuel consumption (L/km) * distance traveled (km) * CO2 per liter. Assuming trucks use more fuel, so we calculate similarly.

- Building: Assuming 0.5 kg CO2 per square foot for electricity usage. Assuming 0.3 kg CO2 per square foot for natural gas.

How the Calculation Works:

1. For the **Car**:

   - Fuel consumption is **0.05 L/km** (liters per kilometer), and the distance traveled is **100 km**.

   - Carbon footprint = 0.05 L/km * 100 km * 2.3 kg CO2/L = **23.00 kg CO2**.

2. For the **Truck**:

   - Fuel consumption is **0.15 L/km**, and the distance traveled is **200 km**.

    - Carbon footprint = 0.15 L/km * 200 km * 2.3 kg CO2/L = **69.00 kg CO2**.

3. For the **Bicycle**:

    - Since bicycles have zero carbon emissions during operation, the footprint is simply **0 kg CO2**.

4. For the **Building**:

    - The building has **2000 square feet** and uses **electricity** (which has an assumed footprint of 0.5 kg CO2 per square foot).

    - Carbon footprint = 2000 sq. ft * 0.5 kg CO2/sq. ft = **1000.00 kg CO2**.

Expected Output Example:

```
Toyota Prius (Car): 9.80 kg CO2

Ford F-150 (Truck): 19.60 kg CO2

Mountain Bike (Bicycle): 0 kg CO2

Building with 2000 square feet using Electricity: 100.00 kg CO2
```

No worries if the output is different.