```
from abc import ABC, abstractmethod

# Pizza Interface (Abstract Base Class)
# This is the target interface that defines the methods we expect in the adapted classes.
# In this case, the interface contains abstract methods 'toppings' and 'bun'.
class Pizza(ABC):
    @abstractmethod
    def toppings(self):
        pass

    @abstractmethod
    def bun(self):
        pass

# DhakaStylePizza class implementing Pizza. This is a concrete class that implements the Pizza interface.
# It provides specific implementations for the 'toppings' and 'bun' methods.
class DhakaStylePizza(Pizza):
    def toppings(self):
        print("Dhaka cheese toppings")

    def bun(self):
        print("Dhaka bread bun")

# ChittagongPizza class (Adaptee)
# This is the class that we want to adapt. It does not implement the Pizza interface.
# It has methods 'sausage' and 'bread'.
class ChittagongPizza:
    def sausage(self):
        print("Ctg pizza")

    def bread(self):
        print("Ctg bread")

# Class Adapter Method, ChittagongClassAdapter (Adapter)

# This is the Adapter class that adapts the ChittagongPizza class to the Pizza interface.
# It uses inheritance from both ChittagongPizza (the Adaptee) and Pizza (the Interface).
class ChittagongClassAdapter(ChittagongPizza, Pizza):
    def toppings(self):
        # The Adapter delegates the 'toppings' method to the 'sausage' method of ChittagongPizza (Adaptee).
        self.sausage()

    def bun(self):
        # The Adapter delegates the 'bun' method to the 'bread' method of ChittagongPizza (Adaptee).
        self.bread()

# The Adapter allows us to use an object of ChittagongClassAdapter as if it were a Pizza,
# even though ChittagongPizza (the Adaptee) doesn't implement the Pizza interface.
adaptedPizza = ChittagongClassAdapter()
adaptedPizza.toppings()  # Output: Ctg pizza
adaptedPizza.bun()       # Output: Ctg bread
```

```
# Object Adapter Method, ChittagongObjectAdapter (Object Adapter)

# This is the Adapter class that adapts the ChittagongPizza class to the Pizza interface.
# It holds a reference to an instance of ChittagongPizza (the Adaptee) and delegates calls to it.
class ChittagongObjectAdapter(Pizza):
    def __init__(self, chittagong_pizza: ChittagongPizza):
        # The adapter holds a reference to the ChittagongPizza (Adaptee).
        self._chittagong_pizza = chittagong_pizza

    def toppings(self):
        # The Adapter delegates the 'toppings' method to the 'sausage' method of ChittagongPizza (Adaptee).
        self._chittagong_pizza.sausage()

    def bun(self):
        # The Adapter delegates the 'bun' method to the 'bread' method of ChittagongPizza (Adaptee).
        self._chittagong_pizza.bread()

chittagong_pizza = ChittagongPizza()  # Create an instance of ChittagongPizza
adaptedPizza = ChittagongObjectAdapter(chittagong_pizza)  # Pass it to the Object Adapter

adaptedPizza.toppings()  # Output: Ctg pizza
adaptedPizza.bun()       # Output: Ctg bread
```