Interface implementation task

You are asked to **implement a new currency class** that will extend the abstract class Currency. The new class should be named after a currency of your choice (e.g., Euro, Yen, Pound, etc.). The implementation should include the following:

I. A constructor that takes two arguments representing this currency abbreviation and amount of money it stores.

II. An implementation of the abstract methods in the Currency class:

   A. Abbreviation: returns the currency abbreviation (e.g., "EUR" for Euro)

   B. ConvertedToDollars: takes one argument representing Cantor object and returns the current amount of your currency in USD based on the exchange rate

   C. Amount: returns about of money in original currency.

III. Additional methods or fields as needed.

You should also **create a main function** that creates an instance of the new currency class, sets the amount to a value of your choice, and prints out the abbreviation and amount in both the original currency and USD.
You can use FakeUsdCantor as source of exchange rate information.

Submission:
Submit a C++ file with the implementation of the new currency class and the main function. Make sure the code is properly formatted and documented.
Also, provide a brief description of the implemented class and any design decisions made.