Making a bankroll

Let's write a bean style class called **`Bankroll`** that represents currency held by the player of a game.

The bean will have a single private instance variable called **`money`** of type **`double`**.

The bean will have a default constructor that initializes the double to zero.

The bean will have a constructor that accepts an argument of type **`double`** and assigns it to the **`money`** variable.

The bean will have a getter and setter for the instance variable **`money`**.

Question: Should there be a setter? Why or why not?

The bean will have a method called **`add`** that accepts an argument of type **`double`** and adds it to the value in **`money`**.

The bean will have a method called **`subtract`** that accepts an argument of type **`double`** and subtracts it from the **`money`**.

Finally, the bean will have a method called **`getFormattedMoney()`** that has no parameters and returns the value of money as a formatted **`String`** in the Currency format.

Next, let's write a test class called **`MoneyTester`**.

This class will have two methods. The first is the standard **`main`** method and it will be written as:

```
public static void main(String[] args) {
    MoneyTester mt = new MoneyTester();
    mt.perform();
    System.exit(0);
}
```

The second method is **`perfom()`** and it should create two **`Bankroll`** objects, one of which is initialized to 100.00. Verify with **`System.out.println`** statements that each object holds the value expected of it by using the **`getFormattedMoney`** method. Add and subtract 10 different values of your choice to each object and display the value after each operation.

Run the program and check that the results displayed are what you expected.