Assignment 2

For this assignment you need to write a Java program that defines an abstract class named `Bank` to represent common functionalities of a bank. Below are the tasks broken down:

1. **Define the Class and Properties:**
    - Create a public abstract class named `Bank`.
    - The class should have three private properties:
        - `accNumber`: a `String` representing the account number.
        - `accName`: a `String` representing the account holder's name.
        - `balance`: a `double` representing the account balance.
2. **Create a Constructor:**
    - Define a constructor for the `Bank` class that takes three parameters: `String accNumber`, `String accName`, and `double balance`.
    - Initialize the class properties (`accNumber`, `accName`, `balance`) using the constructor parameters.
3. **Define Getter Methods:**
    - Implement three public getter methods:
        - `getAccNumber()`: returns the `accNumber`.
        - `getAccName()`: returns the `accName`.
        - `getBalance()`: returns the `balance`.
4. **Define a Setter Method:**
    - Implement a public method `setBalance(double amount)` to update the `balance` property.
5. **Create Abstract Methods:**
    - Define four abstract methods that must be implemented by any subclass of `Bank`:
        - `checkBalance()`: returns a `double`.
        - `deposit(double amount)`: takes a `double` parameter.

- `withdrawal(double amount)`: returns a `boolean` and takes a `double` parameter.
- `transfer(Bank receiver, double amount)`: returns a `boolean` and takes a `Bank` and `double` as parameters.

6. **Create a class call `Saving`:**
    - The `Saving` class will extend the `Bank` and implements all the `abstract` methods defined on the superclass `Bank`.
    - Implement `checkBalance`, `deposit`, and `withdrawal` methods for your `Saving` class.
7. Create a `Tester` class to test your application.
    - Your `Tester` class should create and instance of `Saving` class passing all arguments.
    - Use the `Saving` to perform various banking operation like; `checkBalance`, `deposit`, and `withdrawal`

Happy coding!