Name [ ] Date [ ]

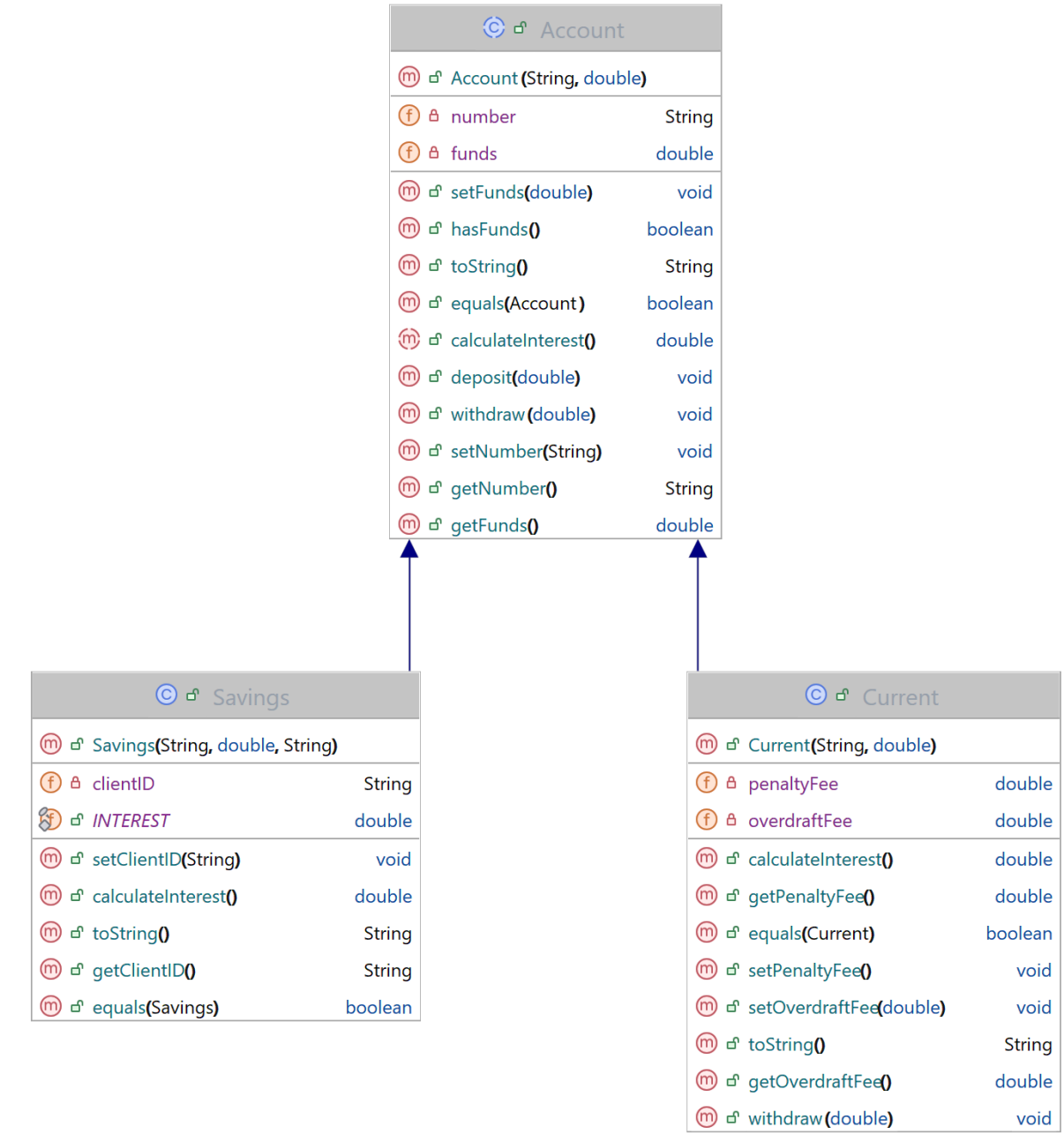
Instructions

The **`Account`** and **`TestAll`** classes are given to you. Write the ***`Current`*** and ***`Savings`*** classes according to the class diagram below:

Work through the test from the beginning. Your classes should build and grow –do not start a new program for each point.

During this test, you may use any resources that you have created, but you may **not** use Internet.

Refer to the UML class diagram on this page when writing your code.



Marks Distribution

1. Successfully implement ***inheritance*** in all your classes as shown in the class diagram.
2. Successfully implement ***all fields/variables*** in all your classes
3. Successfully use ***data hiding*** as required by the UML class diagram given.
4. Successfully implement ***constructors*** in all your classes as per the class diagram.
5. Successfully implement ***accessors*** as shown in the class diagram.
6. Successfully implement ***mutators*** as required by the class diagram.
7. Successfully implement ***toString*** methods in all your classes. They should return Strings as similar as possible to the display output on next page/expected output in the program.
8. Successfully implement ***equals*** methods in the relevant classes as shown in the class diagram.

   In the *Current* class, both *account numbers* must be the same...

   In the *Savings* class, the *account number* and *client IDs* must be the same…

   ...for the objects to be considered equal
9. Implement a ***calculateInterest*** method in the *Savings* class to calculate the interest. Use the formula `funds * INTEREST` (the `INTEREST` is fixed to 5%).
10. In the *Current* class, ***calculateInterest*** always returns zero and the ***overdraftFee*** is $300 by default. If a *withdrawal* goes into overdraft, the ***penaltyFee*** must be *set* to the amount of the overdraft plus the ***overdraftFee***.

<<< Please Turn Over >>>

Display Output

```
-----------------------------------------------------------------------------------------------
Expected output: ClientID: pepeperez | Acct: Savings | Number: SA-1111 | Funds: $5,000.00
>>> Your output: ClientID: pepeperez | Acct: Savings | Number: SA-1111 | Funds: $5,000.00

Expected output: Withdrawing $5k -> Funds: false -> Deposit $2k -> Funds? true
>>> Your output: Withdrawing $5k -> Funds: false -> Deposit $2k -> Funds? true

Expected output: Savings acct. interest rate(%) = 0.05 -> Interest for SA-1111 = $100.0
>>> Your output: Savings acct. interest rate(%) = 0.05 -> Interest for SA-1111 = $100.0

Testing Savings account class equals method
Expected output: mySavingsAcc.equals(dodgy) = true; mySavingsAcc.equals(yourSavings) = false
>>> Your output: mySavingsAcc.equals(dodgy) = true; mySavingsAcc.equals(yourSavings) = false
-----------------------------------------------------------------------------------------------
Expected output: Acct: Current | Number: CA-1234 | Funds: $3,500.00 | Overdraft penalty fee:
$0.00
>>> Your output: Acct: Current | Number: CA-1234 | Funds: $3,500.00 | Overdraft penalty fee:
$0.00

Overdraft fee: $300.0  Expected output: $300.0
Overdraft fee: $875.0  Expected output: $875.0

Expected output: Withdrawing $5k -> Funds: false -> Overdraft penalty fee: $2,375.00
>>> Your output: Withdrawing $5k -> Funds: false -> Overdraft penalty fee: $2,375.00

Testing Current account class equals method
Expected output: dos.equals(tres) = true; myCurrentAcc.equals(dos) = false
>>> Your output: dos.equals(tres) = true; myCurrentAcc.equals(dos) = false
-----------------------------------------------------------------------------------------------
```