This document is part of the RMIT Online course, Programming 1.

Task 6.2.3 Practice Exercises

Implementing subclass methods and method overriding

1. In the program below a programmer derived a subclass to model a fixed rate savings account. Since the constructor for the subclass takes an additional argument, the (fixed) interest- rate, they added an instance variable intRate.

   As they were unable to access the superclass private instance variable directly, an additional instance variable for balance was added in the subclass.

   The driver program they wrote to test the class added interest for 3 months before displaying the final balance. The output showed that the interest has not been added. Identify the problem and correct it.

```
public class SubTest {
	public static void main(String args[]) {
	FixedRateSAccount account = new FixedRateSAccount("f1234", "Joe", 10000.0, 1.0);
	account.addInterest(); // interest for Jan
	account.addInterest(); // interest for Feb
	account.addInterest(); // interest for Mar
	System.out.println(account); }
}

class FixedRateSAccount extends Account {
	private double balance;
	private double intRate;
	public FixedRateSAccount(String accountID, String name, double balance, double
	intRate) {
		super(accountID, name, balance);
		this.intRate = intRate; }
		public void addInterest() {
		balance += balance * intRate / 100.0;
	}
}
```

```
class Account {
	private String accID;
	private String name;
	private double balance;

	public Account(String accountID, String name, double balance) {
		accID = accountID;
		this.name = name;
		this.balance = balance;
	}

	public void deposit(double amount) {
		balance = balance + amount;
	}
	public boolean withdraw(double amount) {
		if (balance >= amount) {
			balance = balance - amount;
			return true;
		}
		else
			return false;
	}
	public double getBalance() {
		return balance;
	}
	public String toString() {
		return String.format("ID = %s, name = %s, Bal = %s", accID, name,
					balance);
	}
}
```

2. Continue working on the ChequeAccount subclass mentioned in the Task 6.1 Practice Exercises to incorporate the following program logic for depositing and withdrawing funds.

a) Define a new method getAvailableFunds() which returns the maximum amount the account holder can withdraw (including the overdraft facility) based on the following formula:

`Total funds available = current balance + (overdraft limit - overdraft amount)`

Question: how do we access the current balance from the ChequeAccount subclass?

b) The withdraw() method should be overridden incorporating the new overdraft facility into the withdrawal process as follows:

- if the withdrawal amount is greater than the total funds available
  - Reject transaction
- else
  - if the current balance is 0
  - Add withdrawal amount to amount overdrawn
- else
  - subtract current balance from the withdrawal amount
  - withdraw the entire current balance
  - (Hint: super.withdraw() and the balance accessor might be useful here)
  - add the remaining withdrawal amount to the amount overdrawn
  - Increment the transaction count by 1

c) The deposit method should be overridden to incorporate the new overdraft facility into the deposit process as follows:

- if overdraft amount is 0
    - Add deposit amount to current balance
- else if the deposit amount is less than or equal to the amount overdrawn
    - subtract deposit amount from amount overdrawn
- else
    - subtract amount overdrawn from deposit amount
    - add remaining deposit amount to current balance