

Michael Romero

CS 1122

Week 2

Exercises

R8.4 The public interface of a class is the features of a class that are accessible to all other assessors. This would include methods and variables not set to private. The implementation of the class is just creating a new object, well the interface is more of the components within an object.

R8.8 A. A mutator method is a method that changes the state of an object. This would include a setter method.

B. An accessor method is a method that accesses an object but never changes the state of an object. This would include a getter method.

R8.11 A constructor is a base method for a class mostly for initializing variables in a class. This method has no return type and shares its name with the class.

R8.14

Step 1

Deposit, withdraw, overdraft penalty, add interest (can change)

Step 2

Public void deposit (double amount)

Public void withdraw (double amount)

Public void addInterest (double rate)

Public double getBalance()

Step 3

Add comments for methods

//A bank account that can deposit or withdraw and take interest into account.

Public class BankAccount{

//A constructor to make a bank account that starts at zero

Public BankAccount()

//deposits money into the account

@param the amount to be deposited

Public void deposit (double amount)

//Makes a withdraw to the account, also adds a overdraw penalty is insufficient funds

@param the amount to be withdrawn

Public void withdraw (double amount)

// Adds interest to the account

@param rate of the interest in percentage.

Public void addInterest (double rate)

//Gets the balance of the account

@return the current balance

Public double getBalance()

}

Step 4

We need private double balance

Step 5

Public double getBalance(){

return balance;

}

Public void deposit (double amount){

balance = balance + amount;

}

Public void withdraw (double amount){

final double Penalty = 10;

if(amount > balance){

balance = balance – Penalty;

}else{

balance = balance - amount;

}

}

Public void addInterest (double rate){

Double amount = balance * rate / 100;

```
balance = balance + amount;
```

```
}
```

```
public BanckAccount(){
```

```
balance = 0;
```

```
}
```

```
public BankAcount(double initialBalance){
```

```
balance = initialBalance;
```

```
}
```

Step 6 test.