Lecture 9

We write another class definition, this one is for simple bank accounts. We see constructors, instance methods and instance variables again.

We introduce the object reference `this`.

The object reference `this`

- Sometimes you want to refer to the current object, i.e.
  - the object that is being constructed *or*
  - the object that has received a message and is executing a method.
- Refer to it using `this`
  - `this` is a local variable
  - it is automatically declared in every constructor and instance method
  - it is automatically initialised to refer to the current object
- (There is another separate use of `this`. Don't confuse them.)

What do we want the `BankAccount` class to offer?

Some client code

```
// Ann arrives at the UCC branch and opens
// a new account, with 25 pounds in it.

BankAccount annsAccount =
   new BankAccount("UCC", 2500);

// Ben arrives at the UCC branch and opens
// an account, with 34.56 in it.

BankAccount bensAccount =
   new BankAccount("UCC", 3456);

// Ann returns and deposits 26.43.
annsAccount.credit(2643);

// Ben returns; he withdraws 20 pounds...

bensAccount.debit(2000);
```



```
// ...and he transfers 20 pounds to Ann...

bensAccount.transfer(2000, annsAccount);

// ...and he changes the branch where his
// account is held.

bensAccount.setBranch("Patrick Street");

// And now Ann's back: she withdraws the
// whole amount in her account, how ever
// much that is.

annsAccount.debit(
   annsAccount.getCurrentBalance());
```


The `BankAccount` interface

Constructor

```
public BankAccount(String theBranch,
   int anOpeningBalance)
```

Getters

```
public int getCurrentBalance()
public String getBranch();
```

Setters

```
public void credit(int anAmount)
public void debit(int anAmount)
public void transfer(int anAmount,
   BankAccount theDestinationAccount)
public void setBranch(String branch)
```

(A useful class might offer many more.)

The `BankAccount` class definition

```
/**
 * A class that represents bank accounts.
 * @author Derek Bridge 666 d.bridge@cs.ucc.
 */
public class BankAccount
{ /* ====================================
         CONSTRUCTORS
     ====================================
  */
   /**
    * Allocates a new bank account object
    *  with the given opening balance.
    * @param theBranch the branch where this
    * account is held.
    * @param anOpeningBalance the opening
    * balance.
    */
   public BankAccount(String theBranch,
      int anOpeningBalance)
   {
   }
```

```java
/* ====================================
      PUBLIC INTERFACE
   ====================================
*/
/* --Getters------------------------ */
   /**
    * Returns the current balance.
    * @return the current balance.
    */
   public int getCurrentBalance()
   {
   }

   /**
    * Returns the branch where the
    * account is held.
    * @return the branch where the
    * account is held.
    */
   public String getBranch()
   {
   }
```



```java
/* --Setters------------------------ */

   /**
    * Credit the account with the specified
    * amount.
    *
    * @param anAmount the amount by which
    * to credit this account.
    */
   public void credit(int anAmount)
   {
   }

   /**
    * Debit the account by the specified
    * amount.
    *
    * @param anAmount the amount by which
    * to debit this account.
    */
   public void debit(int anAmount)
   {
   }
```



```java
   /**
    * Transfer the specified amount from
    * this account to the specified account.
    *
    * @param anAmount the amount to transfer
    * @param theDestinationAccount the
    * account to be credited.
    */
   public void transfer(int anAmount,
      BankAccount theDestinationAccount)
   {
   }

   /**
    * Sets the branch where this account is
    * held to the specified branch.
    */
   public void setBranch(String branch)
   {
   }
```



```java
/* ====================================
      INSTANCE VARIABLES
   ====================================
*/

}
```