ArrayList

The `for each` loop

Introduction to OO Programming

The `for each` loop

- The basic `for` loop was extended in Java 5 to make iteration over arrays and other collections more convenient.
- This newer `for` loop is called the ***enhanced for*** or ***for-each***
- The loop is used to access each successive element in a collection
- It is commonly used to traverse or iterate over an array or a `Collections` class (eg, `ArrayList`).
- It can also be used with Arrays

The `for each` loop

```
ArrayList<BankAccount> accounts = new ArrayList<BankAccount>();
double sum = 0;

//for each BankAccount b in accounts
for (BankAccount b : accounts)
{
   sum = sum + b.getBalance();
}

//for loop to do exactly the same thing
for (int i = 0; i < accounts.size(); i++)
{
   BankAccount b = accounts.get(i);
   sum = sum + b.getBalance();
}
```

Rewrite using enhanced `for`

```
for(int i = 0; i < accounts.size(); i++)
{
   //get object at element i and assign to b
   BankAccount b= accounts.get(i);

   //view the details of acc
   System.out.print("Acc no. is " + b.getAccNumber());
   System.out.println(" Balance is " + b.getBalance());
}
```

Solution

```
for(BankAccount b : accounts)
{
   //view the details of b
   System.out.print("Account no. is " +b.getAccNumber());
   System.out.println(" Balance is "  + b.getBalance());
}
```

Solution

BankAccount object reference b

ArrayList object reference

```
for(BankAccount b : accounts)
{
   //view the details of b
   System.out.print("Account no. is " +b.getAccNumber());

   System.out.println(" Balance is "  + b.getBalance());
}
```

Methods of the BankAccount class

Rewrite using enhanced `for`

```
for(int i = 0; i<myList.size(); i++)
{
   String s = myList.get(i);
   System.out.println(s);
}
```

Solution

```
for(String s : myList)
{
   System.out.println(s);
}
```

Using with arrays

- Traverses all elements of a collection:

```
double[] data = new double[10];
double sum = 0;
for (double d : data) // You should read this loop as
{                     //"for each element d in data"
   sum = sum + d;
}
```

Note: not particularly helpful when working with arrays of primitives!

The traditional `for` Loop

- Traditional alternative:

```
double[] data = new double[10];
double sum = 0;
for (int i = 0; i < data.length; i++)
{
   sum = sum + data[i];
}
```