Model 1 Static Variables

Consider the definition for a bank account:

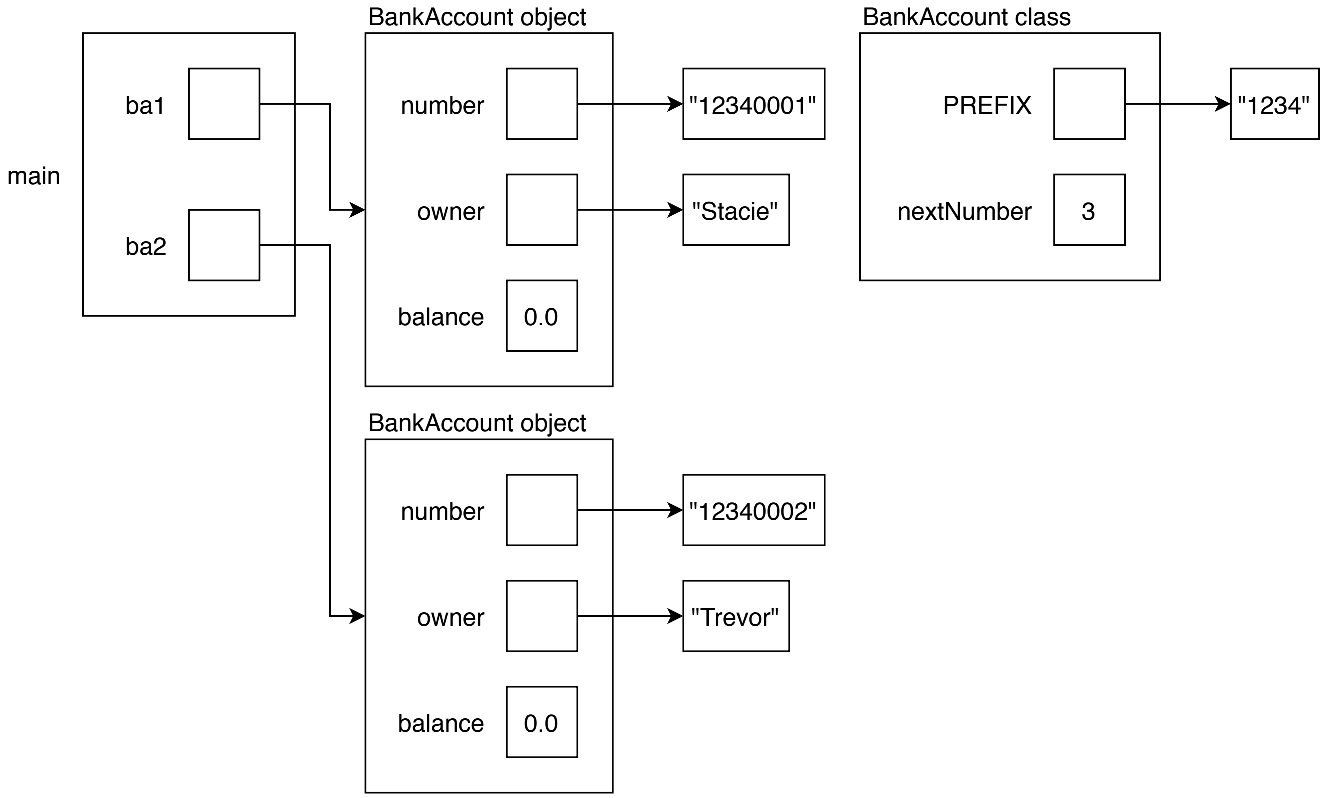
```
public class BankAccount {
    private static final String PREFIX = "1234";
    private static int nextNumber = 1;

    private String number;
    private String owner;
    private double balance;

    public BankAccount(String owner) {
        this.number = PREFIX + String.format("%04d", nextNumber);
        this.owner = owner;
        nextNumber++;
    }
}
```

Here is a memory diagram of two `BankAccount` objects:

```
public static void main(String[] args) {
    BankAccount ba1 = new BankAccount("Stacie");
    BankAccount ba2 = new BankAccount("Trevor");
}
```



Questions (15 min)

Start time:

1. Based on the source code and memory diagram:

a) How many `BankAccount` variables were declared? 2

b) How many `BankAccount` objects were created? 2

2. How many instances of each variable are in memory?

a) `PREFIX` 1

b) `nextNumber` 1

c) `number` 2

d) `owner` 2

e) `balance` 2

3. What is the difference between `static` and non-`static` variables of a class? Explain your answer in terms of the diagram.

Static variables are shared by all instances of the class. They are shown in the box labeled "BankAccount class". Non-static variables are specific to each object. They are shown in boxes labeled "BankAccount object".

4. Why are all the strings shown in separate boxes as opposed to being written inside of the variable boxes?

Strings are objects, so their variables contain references. (There is not enough memory inside the variable to store the entire string.)

5. How would you modify the memory diagram if the following line were added at the end of the `main` method?

```
BankAccount ba3 = ba2;
```

Add a new box for `ba3` and draw an arrow to the object for `ba2`. Notice that assigning reference types does not create new objects!

6. (Optional) Paste the contents of *BankAccount.java* into Java Visualizer. What differences do you notice between the diagram in Java Visualizer and those in Model 1?

Answers might include:

- The static fields are drawn above the top frame.
- The strings are drawn inside the objects (for convenience).