

# STATIC VARIABLES & STATIC METHODS

---

CS302 – Introduction to Programming  
University of Wisconsin – Madison  
Lecture 22

By Matthew Bernstein – [matthewb@cs.wisc.edu](mailto:matthewb@cs.wisc.edu)

# Static Variables

- Sometimes, a value properly belongs to a class and not any object of the class.
- A variable that has the same value across all objects of a class is called a **static variable** and is denoted with the “static” reserved word
- Example:

```
private static int accountNumber;
```

# Uses of Static Variables

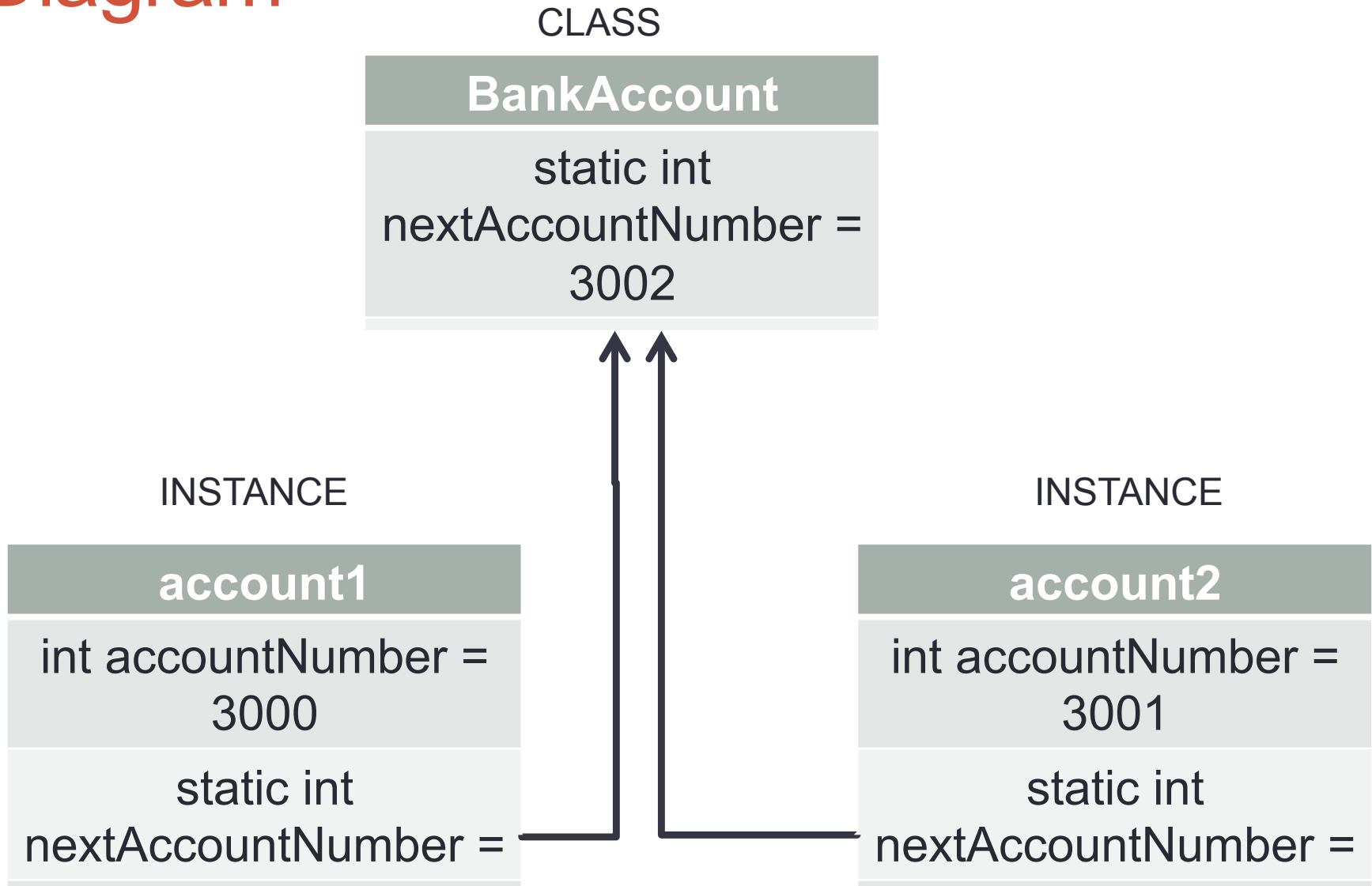
- For example, let's say we want to assign bank account numbers sequentially
- How do we coordinate how different objects of the class `BankAccount` initialize their account number?
- We use a static variable that belongs to the class itself and not to each instantiation

# Example of Static Variables

```
class BankAccount
{
    // Static variable
    private static int nextAccountNumber;

    // Instance variable
    private int accountNumber
}
```

# Diagram



# Constant Static Variables

- If you have constant variables that belong to a certain class, then you should declare it as a final static variable

## Example: Math.PI

- The `Math.PI` references a constant, static variable called `PI` that belongs to the `Math` class
- This design makes sense because the number `PI` never changes.
- Furthermore, it is natural for this variable to belong to the `Math` class because it is commonly used with mathematical applications

# Constant Static Variables

- It is okay to make constant, static variables public
- Note how we do NOT do the following to retrieve the number Pi:

`Math.getPi()`

(this method doesn't exist)



# Static Methods

- A method that belongs to the class itself and is not called from an instance is a **static method**
- We call a static method from the class and NOT from an instance of the class:
- Example:

`Math.pow(2,3);`

- We call `pow()` from the `Math` class itself and not from any instantiation of the `Math` class
- Why?
- Conceptually, `pow()` is an operation that belongs in the category of Mathematical operations. Furthermore, it acts on two values and not on an object

# Static Methods

- Declaring a static method:

```
public static void calculateMean(int[] array)
{
    ...
}
```

- Static Methods **MUST NOT** make any reference to an instance variable
- This makes sense, because instance variables belong to instantiations of a class. The class itself does not own any instance variables.
- You will get a compile-time error if you use an instance variable in a static method

# More on Static Methods

- You **CANNOT** call a non-static method from a static method. However you **CAN** call a static method from a either another static method or a non-static method.
- The **main** method is always static. If you want to call a method in the same class from your main method, the called method must be also be static