

## Categories of Variables

## Categories of Variables

- Categories of variables
  - Instance fields (`balance` in `BankAccount`)
  - Local variables (`newBalance` in `deposit` method)
  - Parameter variables (`amount` in `deposit` method)
- An instance field belongs to an object
- The fields stay alive until no method uses the object any longer

## Categories of Variables

- In Java, the *garbage collector* periodically reclaims objects when they are no longer used
- Local and parameter variables belong to a method
- Instance fields are initialized to a default value, but you must initialize local variables

## Lifetime of Variables

```
harrysChecking.deposit(500);  
double newBalance = balance + amount;  
balance = newBalance;
```

## Lifetime of Variables

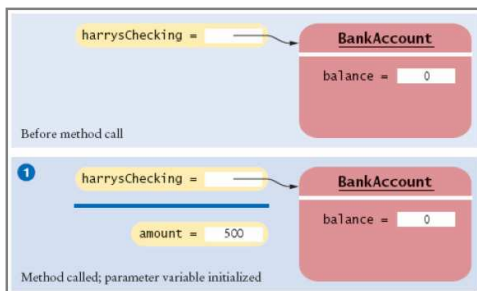


Figure 7:  
Lifetime of Variables

## Lifetime of Variables

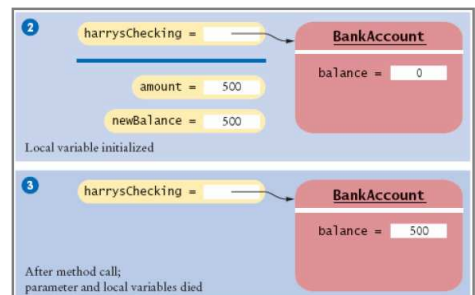


Figure 7:  
Lifetime of Variables

## Implicit and Explicit Method Parameters

- The implicit parameter of a method is the object on which the method is invoked
- The `this` reference denotes the implicit parameter

## Implicit and Explicit Method Parameters

- Use of an instance field name in a method denotes the instance field of the implicit parameter

```
public void withdraw(double amount)
{
    double newBalance = balance - amount;
    balance = newBalance;
}
```

## Implicit and Explicit Method Parameters

- `balance` is the balance of the object to the left of the dot:

```
momsSavings.withdraw(500)
```

means

```
momsSavings.balance = momsSavings.balance - amount;
```

## Implicit Parameters and `this`

- Every method has one implicit parameter
- The implicit parameter is always called `this`
- Exception: Static methods do not have an implicit parameter (more on Chapter 9)

```
double newBalance = balance + amount;
// actually means
double newBalance = this.balance + amount;
```

## Implicit Parameters and `this`

- When you refer to an instance field in a method, the compiler automatically applies it to the `this` parameter

```
momsSavings.deposit(500);
```

## Implicit Parameters and `this`

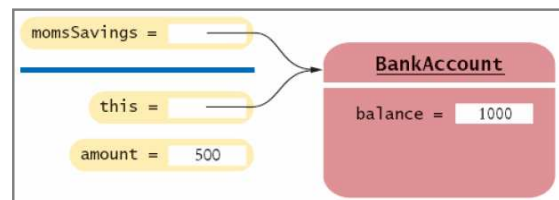


Figure 8:  
The Implicit Parameter of a Method Call

## Questions

- How many implicit and explicit parameters does the `withdraw` method of the `BankAccount` class have, and what are their names and types?
  - One implicit parameter, called `this`, of type `BankAccount`, and one explicit parameter, called `amount`, of type `double`.

## Questions

- In the `deposit` method, what is the meaning of `this.amount`? Or, if the expression has no meaning, why not?
  - It is not a legal expression. `this` is of type `BankAccount` and the `BankAccount` class has no field named `amount`.

## Questions

- How many implicit and explicit parameters does the `main` method of the `BankAccountTester` class have, and what are they called?
  - No implicit parameter—the method is static—and one explicit parameter, called `args`.