

## Implementing Constructors and Methods

## Implementing Constructors

- Constructors contain instructions to initialize the instance fields of an object

```
public BankAccount()  
{  
    balance = 0;  
}  
public BankAccount(double initialBalance)  
{  
    balance = initialBalance;  
}
```

## Constructor Call Example

```
BankAccount harrysChecking = new BankAccount(1000);
```

- - Create a new object of type `BankAccount`
  - Call the second constructor (since a construction parameter is supplied)
  - Set the parameter variable `initialBalance` to 1000
  - Set the `balance` instance field of the newly created object to `initialBalance`
  - Return an object reference, that is, the memory location of the object, as the value of the `new` expression
  - Store that object reference in the `harrysChecking` variable

## Implementing Methods

- Some methods do not return a value

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

- Some methods return an output value

```
public double getBalance()  
{  
    return balance;  
}
```

## Method Call Example

```
harrysChecking.deposit(500);
```

- - Set the parameter variable `amount` to 500
  - Fetch the `balance` field of the object whose location is stored in `harrysChecking`
  - Add the value of `amount` to `balance` and store the result in the variable `newBalance`
  - Store the value of `newBalance` in the `balance` instance field, overwriting the old value

## Syntax 3.5: The `return` Statement

```
return expression;  
or  
return;
```

**Example:**  
`return balance;`

**Purpose:**  
To specify the value that a method returns, and exit the method immediately. The return value becomes the value of the method call expression.