

bankaccounttester

## Class BankAccount

- java.lang.Object
- 
- bankaccounttester.BankAccount

- ---

```
public class BankAccount
extends java.lang.Object
```

Describes a bank account whose balance can be changed by deposits and withdrawals.

Since:

```
99-99-9999
Course: csc 1350 Section ¾
Lab # 8
Fall 2019
```

Author:

```
YOUR NAME
```

- 

### • *Field Summary*

#### Fields

Modifier and Type	Field and Description
private double	<b>balance</b> the account balance

### • *Constructor Summary*

#### Constructors

##### Constructor and Description

**BankAccount**()  
Constructs a bank account with a zero balance.

**BankAccount**(**BankAccount** other)  
Constructs a bank account equivalent to the specified bank account

**BankAccount**(double initialBalance)  
Constructs a bank account with a given balance

- **Method Summary**

**All Methods**Instance MethodsConcrete Methods

Modifier and Type	Method and Description
Void	<b>deposit</b> (double amount) Deposits money into the account.
double	<b>getBalance</b> () Gets the account balance.
java.lang.String	<b>toString</b> () Returns a string representation of this bank account in the format "BankAccount[balance = ...]", where ...
void	<b>withdraw</b> (double amount) Withdraws money from the account.

- **Field Detail**

- **balance**

private double balance

the account balance

- **Constructor Detail**

- **BankAccount**

public BankAccount()

Constructs a bank account with a zero balance.

- **BankAccount**

public BankAccount(BankAccount other)

Constructs a bank account equivalent to the specified bank account

**Parameters:**

other - a bank account

- **BankAccount**

- ```
public BankAccount(double initialBalance)
    throws java.lang.IllegalArgumentException
```

Constructs a bank account with a given balance

**Parameters:**

`initialBalance` - the initial balance

**Throws:**

`java.lang.IllegalArgumentException` - when the initial balance is negative

- ***Method Detail***

- **deposit**

- ```
public void deposit(double amount)
    throws java.lang.IllegalArgumentException
```

Deposits money into the account.

**Parameters:**

`amount` - the amount of money to withdraw

**Throws:**

`java.lang.IllegalArgumentException` - when `amount < 0`

- **withdraw**

- ```
public void withdraw(double amount)
    throws java.lang.IllegalArgumentException
```

Withdraws money from the account.

**Parameters:**

`amount` - the amount of money to deposit

**Throws:**

`java.lang.IllegalArgumentException` - when `amount > balance`

- **getBalance**

```
public double getBalance()
```

Gets the account balance.

**Returns:**

the account balance

- **toString**

```
public java.lang.String toString()
```

Returns a string representation of this bank account in the format "BankAccount[balance = ...]", where ... denotes the balance.

**Overrides:**

toString in class java.lang.Object

**Returns:**

a string representation of this bank account in the format "BankAccount[balance = ...]", where ... denotes the balance.