ITEC1620
Object-Based Programming

Lecture 15

Object-Based Programming

Object-Based Programming

- Object-Based Programming is structured programming using objects
- Objects are user-defined datatypes
  - Groups of related data
- Objects are created with constructors
- Objects are accessed via methods
- All object information is in the API

Groups of Related Data

- BankAccount
  - String name
  - String accountNumber
  - float balance
  - float interestRate
  - int accountType

Actions with Data

- Open a BankAccount
  - new BankAccount(String accountHolder)
- Banking activities
  - getBalance()
  - withdraw()
  - deposit()
    - ➢Note: no setBalance()

Class Data vs. Instance Data

- Is data related to a single instance or the entire class?
- interestRate
  - Does each account have an interest rate, or do all accounts have the same interest rate?
- balance
  - Does each account have its own balance?

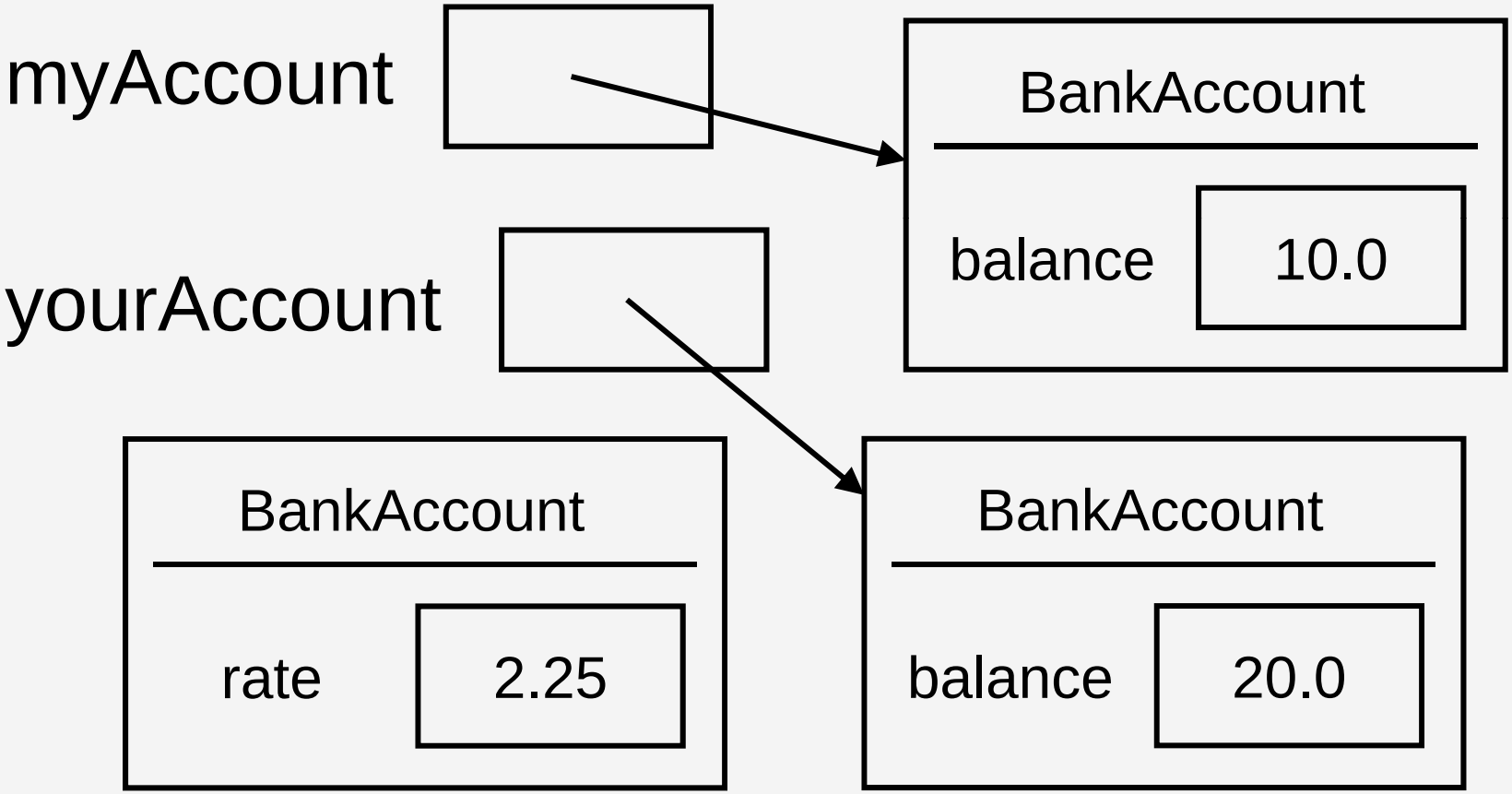
Class Methods vs. Instance Methods

- Methods that interact with instance data must be instance methods
  - getBalance(), withdraw(), deposit(), etc
- Methods that do not interact with instance data can be class methods
  - setInterestRate()
- What about calculateInterest()?

Class Data and Methods

- Class data and methods are identified as “static”
  - static data is NOT constant
  - final data is constant
- static fields and methods can be accessed by the class name or by an instance variable/identifier
  - Use class name for clarity

Example


myAccount
yourAccount
BankAccount
balance
10.0
BankAccount
rate
2.25
BankAccount
balance
20.0

Example II

BankAccount.setRate(2.50);

- Sets interest rate for myAccount and yourAccount

myAccount.setRate(2.75);

- Sets interest rate for myAccount and yourAccount
- One interest rate for all accounts

Example III

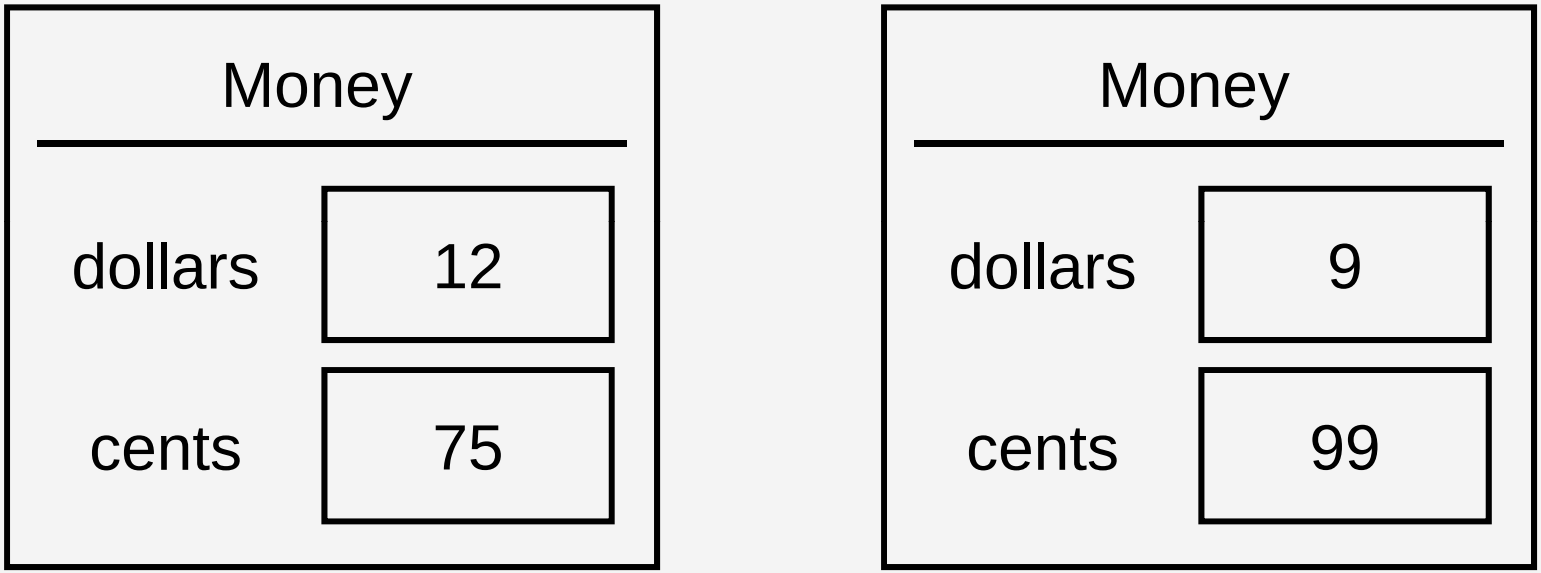
- yourAccount.setBalance()
  - You wish!
  - API controls how interaction occurs
- myAccount.calculateInterest()
  - Non-static methods have access to both static and non-static fields and methods
- BankAccount.calculateInterest()
  - Using what account data?

Questions?

Understanding Objects

- Is balance a float?
  - No
- Money is an integer number of dollars and an integer number of cents (from 0 to 99)
  - int dollars and int cents
  - Group together into a Money object

Understanding Objects II



- Which one is larger?

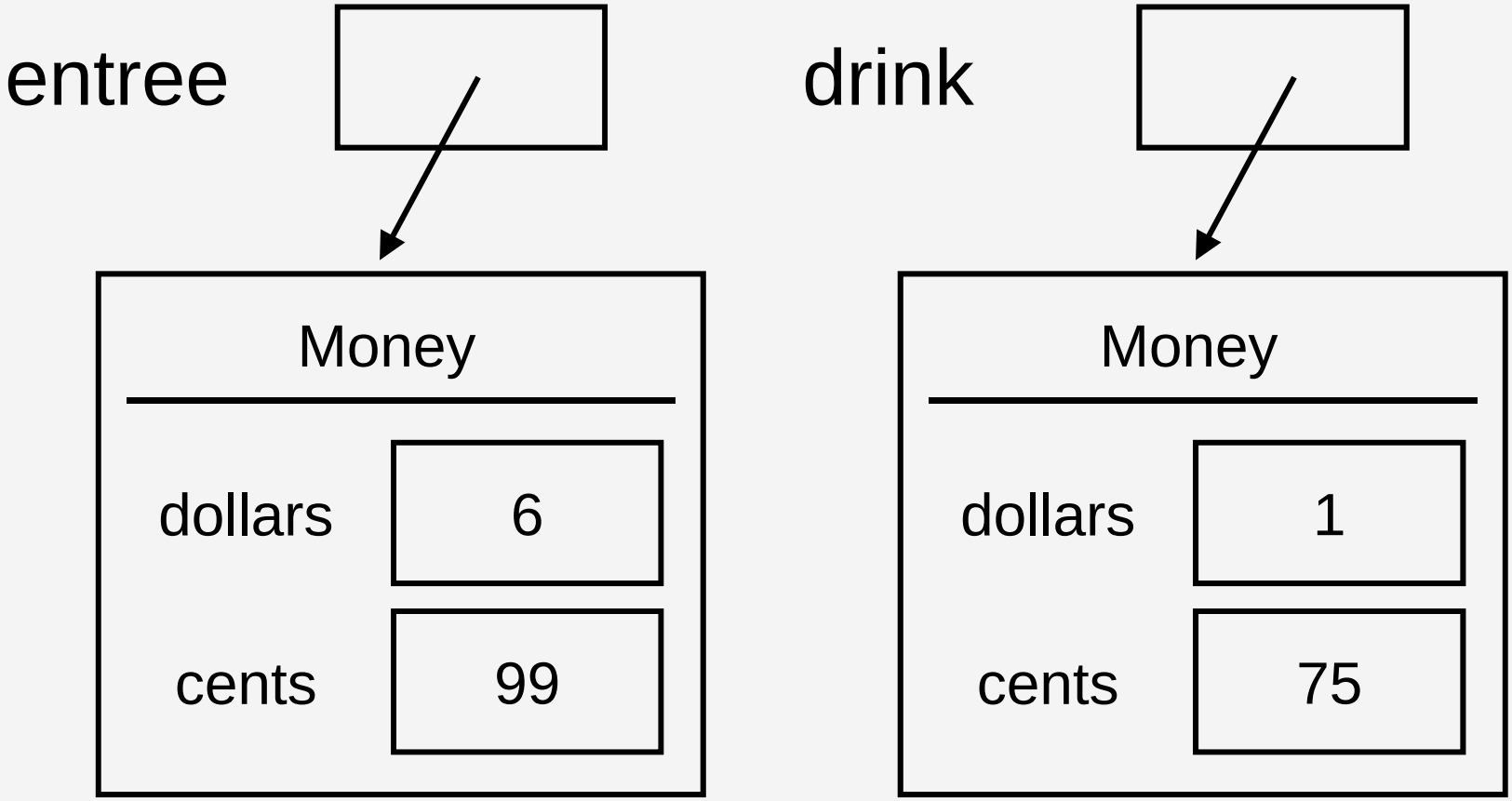
Understanding Objects III

- User-defined data requires user-defined operations
  - add, subtract, multiply, less than, equals
- Java supplies these for java's (primitive) datatypes
- You have to provide them for your datatypes

Understanding Objects IV

static Money	add(Money first, Money second) Returns a new Money object that is sum of first and second
void	addTo(Money amount) Adds the given Money amount to the current money object
boolean	isGreaterThan(Money amount) Returns true if the current amount is greater than the given Money amount

Understanding Objects V



Understanding Objects VI

cost = entree + drink;

- Syntax error
  - Cannot use primitive addition on user-defined datatypes

Money cost = Money.add(entree, drink);

Money total = new Money(0, 0);

total.addTo(entree);

total.addTo(drink);

Understanding Objects VII

if(cash > cost)

- Syntax error
  - Cannot use boolean operators on user-defined datatypes

if(cash.isGreaterThan(cost))

Questions?

Example

- Three classes
  - Money
  - BankAccount
  - CashCard
- Two code fragments (part of Bank)

```
// transfer
boolean successful;
CashCard from;
CashCard to;
Money amount;
```

```
// buyCard
CashCard newCard;
Money amount;
BankAccount account;
```

Readings and Assignments

- Tutorial – Object-Based Programming
- Lab Assignment 4