Lecture 3 – Java Classes

08-671
Java Programming for App Developers

September 8, 2015

Jeffrey L. Eppinger & Terry Lee

Our Teaching Assistants

- ✓ Anny Ni
- ✓ Yifu Wang
- ✓ Hao Fu
- ✓ Tianyue Xiao
- ➢ Hao Tang
- ➢ Xinyue Wu
- • Renzo Bautisa
- • Akhil Prakash

See Blackboard/Piazza for office hours & additional contact info

08-671 Lecture Topics

(subject to change – but only a little bit)

#1 Intro

#2 Primitive Types

#3 Java Classes

#4 Reference Types

#5 Loops & Arrays

#6 Lists & Sorting

#7 Maps

#8 File & Network I/O

#9 Swing Interfaces

#10 Swing Actions

#11 Threads

#12 Exceptions

#13 Functional Programming

#14 In-class Written Exam

* Final Exam – this will be a 3-hour programming problem

Outline

⇨ Administrative Issues

Questions

Our First Class Discussion

Sample Final Exam Question

Java Security

- Keep Java up to date
- Especially important in your browser
  - Java sandbox keeps web pages from doing bad things
    - But can be security flaws
  - Turn-off Java plug-in in browser or make it ask
- To check your browser plug-ins, go to:

  https://www.cmu.edu/iso/patch-check
- Suggestion:
  - Use different browsers for different purposes
  - Make browser ask you before running all plug-ins
    - Enable on a per-site basis

Copyright Violations

- You can take copies of course materials for personal use
  - This applies especially to the slides
  - This applies to the course examples if they are so marked
  - This should also apply to the videos
    - But we don't make it easy for you to copy them
    - We also tell you not to copy the videos

Video Recordings

- We’re making videos of the class
- I’ve been notified that there are rules and restrictions
- I’ve been advised to warn you about the videos

Video Warnings

- Videos of lectures are prepared by CMU using Panopto software. These videos are for use by students enrolled in the course and are not to be copied or made available to others.
- As the class is over-subscribed, I will give access, upon request, to CMU students not enrolled in the class, but you are not to copy the videos or make them available to others.
- If you ask questions in class they will possibly be captured on video. If you do not feel comfortable having your questions recorded, please talk to the instructor about other arrangements.
- No student may record or tape any classroom activity without the express written consent of the instructor

Getting Help

- You can ask questions after class
  - Terry & I stay after class to answer questions
  - We can talk with you in the seating outside the auditorium
- You can go to recitation
- You can go to TA office hours

So That’s Help on Six Days/Week

- Sunday: Renzo, Akhil, Tianyue Office Hours
- Monday: Anny & Yifu Office Hours
- Tuesday: Lecture & Xinyue Office Hours
- Wednesday: Hao Tang Office Hours
- Thursday: Lecture & Hao Fu Office hours
- Friday: Recitation

Note: Homework is expected to be due on Mondays.

Outline

✓ Administrative Issues

⇨ Questions

Our First Class Discussion

Sample Final Exam Question

Remember HelloWorld.java?

- Do I need to indent it this way?

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

White Space Makes it Pretty

- Spaces separating words are significant
- Other white space just makes the program readable by us
  - White space characters: “space” and “tab” and “new line”
  - The compiler ignores this extra white space
- So this is also HelloWorld:

```
public class HelloWorld{public static void main
(String[]args){System.out.println("Hello, World!"
);}}
```

```
public
class
HelloWorld
{
public
static
void
main
(
String
[
]
args
)
{
System.out.println
(
"Hello, World!"
)
;
}
}
```

This is also HelloWorld

Why the semicolons?

- Java needs the semicolons to separate statements
- Curly braces are used to group statements

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
    System.out.println("Good Bye, Cruel World!");
}
```

Outline

✓ Administrative Issues

✓ Questions

⇨ Our First Class Discussion

Sample Final Exam Question

So What's a Class, Anyway?

- A class is a template definition
- It defines the data contained in objects that are instances (members) of a class
- It defines the methods (operations) that can be performed on those instances
- HelloWorld is a degenerate example of all of the above

HelloWorld.java (from Lecture 1)

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Example with Variables

```
public class AddressBookEntry {
    String firstName;
    String lastName;
    String telephoneNumber;
    String eMailAddress;

    public static void main(String[] args) {
        AddressBookEntry jeff = new AddressBookEntry();
        jeff.firstName = "Jeffrey";
        jeff.lastName  = "Eppinger";
        jeff.telephoneNumber = "412-268-7688";
        jeff.eMailAddress = "eppinger@cmu.edu";
    }
}
```

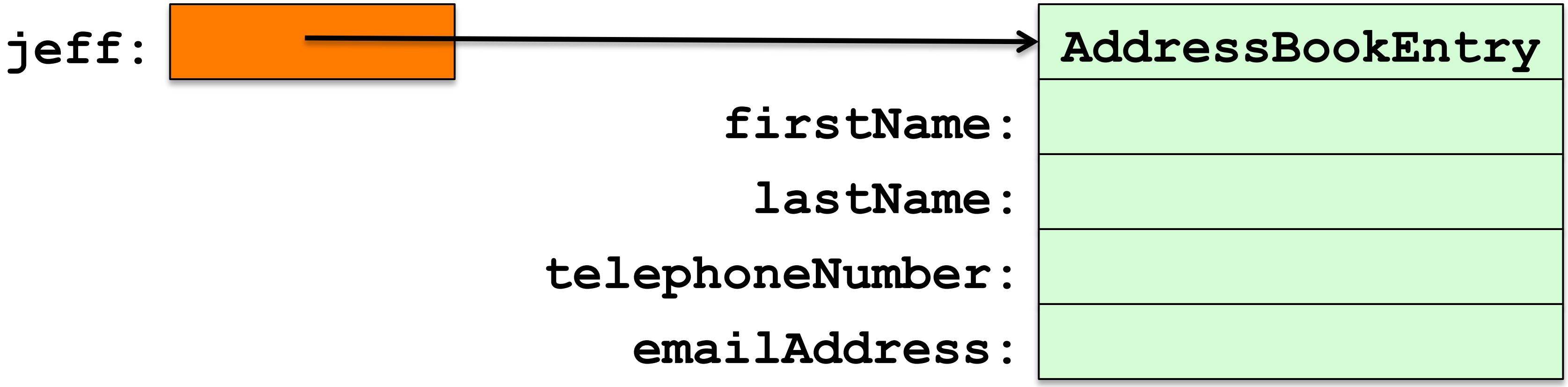
Instance Variables

Local Variable

Instances of Objects

- The "new" operator creates instances of an object

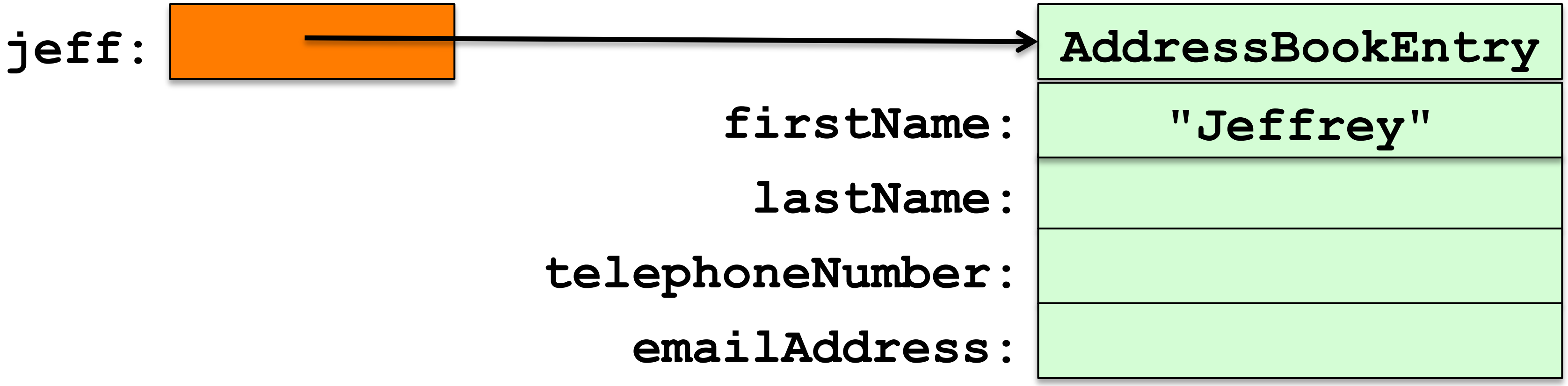
```
AddressBookEntry jeff = new AddressBookEntry();
```



Field access

- Instance variables are “fields” in an object
  - There are other types of fields which we will discuss later

```
jeff.firstName = "Jeffrey”;
```



jeff:

AddressBookEntry

firstName: "Jeffrey"

lastName:

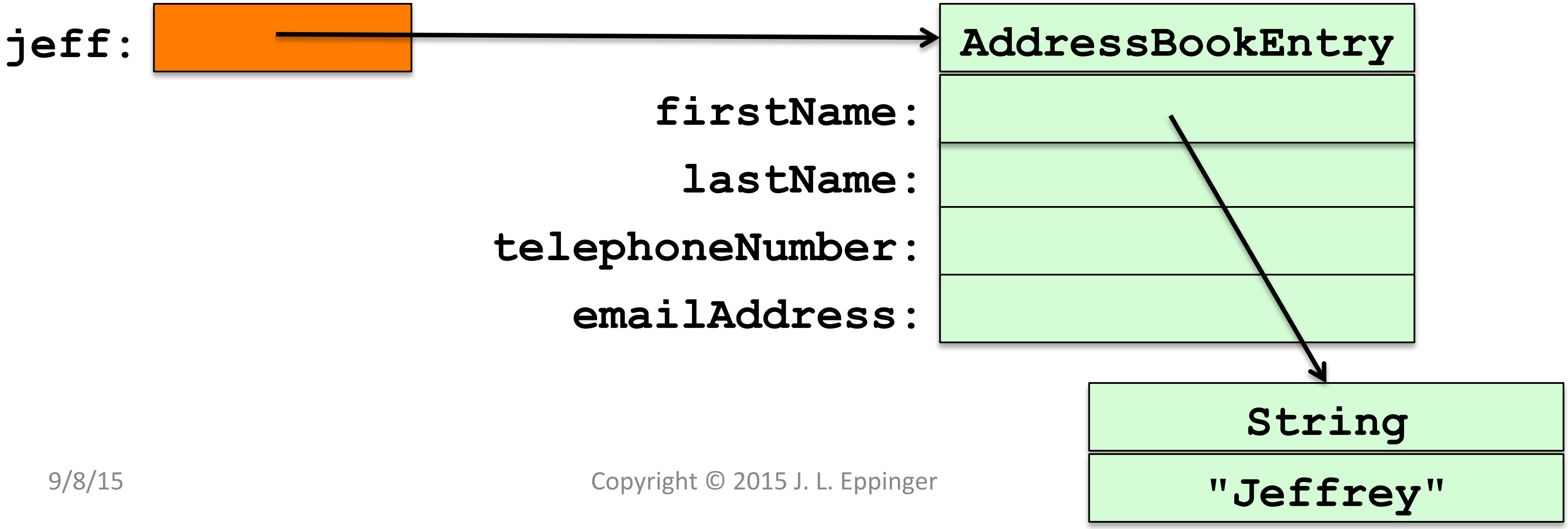
telephoneNumber:

emailAddress:

Strings are Objects, too

- Later we will discuss that Strings are objects

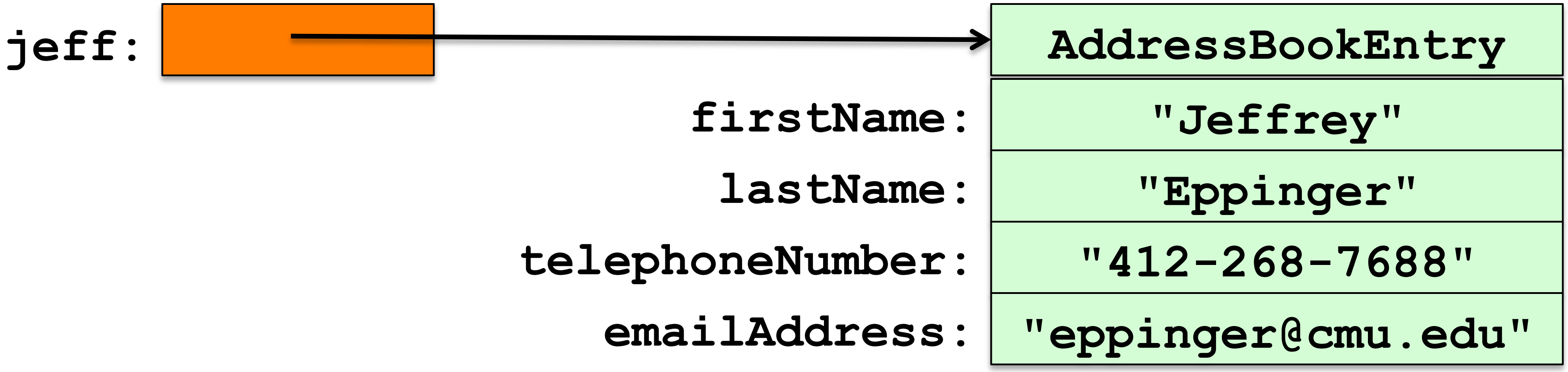
```
jeff.firstName = "Jeffrey";
```



So for now…

- Just consider the Strings as values

```
jeff.firstName = "Jeffrey";
jeff.lastName  = "Eppinger";
jeff.telephoneNumber = "412-268-7688";
jeff.eMailAddress = "eppinger@cmu.edu";
```



jeff:

AddressBookEntry

firstName: "Jeffrey"

lastName: "Eppinger"

telephoneNumber: "412-268-7688"

emailAddress: "eppinger@cmu.edu"

We Need Some Output

```
public class AddressBookEntry {

    // Instance variable declarations removed for space

    public static void main(String[] args) {
        AddressBookEntry jeff = new AddressBookEntry();
        jeff.firstName = "Jeffrey";
        jeff.lastName  = "Eppinger";
        jeff.telephoneNumber = "412-268-7688";
        jeff.eMailAddress = "eppinger@cmu.edu";

        System.out.println("Name:   " + jeff.firstName +
                       " " + jeff.lastName);
        System.out.println("Tel:    " + jeff.telephon…);
        System.out.println("E-Mail: " + jeff.eMailAdd…);
    }
}
```

We Need An Output Method

```
public class AddressBookEntry {


    // Instance variable declaratio       emov

    void print() {
        System.out.println("Name:    " + firstName +
                           " " + lastName);
        System.out.println("Tel:     " + telephoneNumber);
        System.out.println("E-Mail: " + eMailAddress);
    }

    public static void main(String[] args) {
        AddressBookEntry jeff = new AddressBookEntry();
        // Setting of instance variables removed for space
        jeff.print();
    }
}
```

Instance Method

Lots of Entries

```
public class AddressBookEntry {
    // Instance variables removed for space
    // Instance methods (print) removed for space

    public static void main(String[] args) {
        AddressBookEntry jeff = new AddressBookEntry();
        // Setting of fields removed for space
        jeff.print();

        AddressBookEntry barry = new AddressBookEntry();
        // Setting of fields removed for space
        barry.print();
    }
}
```

The Whole Example

- Look in `AddressBookEntry.java`
- How would you run this program?

Some Numbers

```
public class AddressBookEntry2 {
    String firstName;
    String lastName;
    String telephoneNumber;
    String eMailAddress;
    int    age;

    void birthday() {
        age = age + 1;
    }

    // fixed up print and main removed for space
}
```

Inheritance

- Bother you that I had to copy all of **`AddressBookEntry`** into **`AddressBookEntry2`**?
- A subclass can extend and modify a class

Inheritance Example

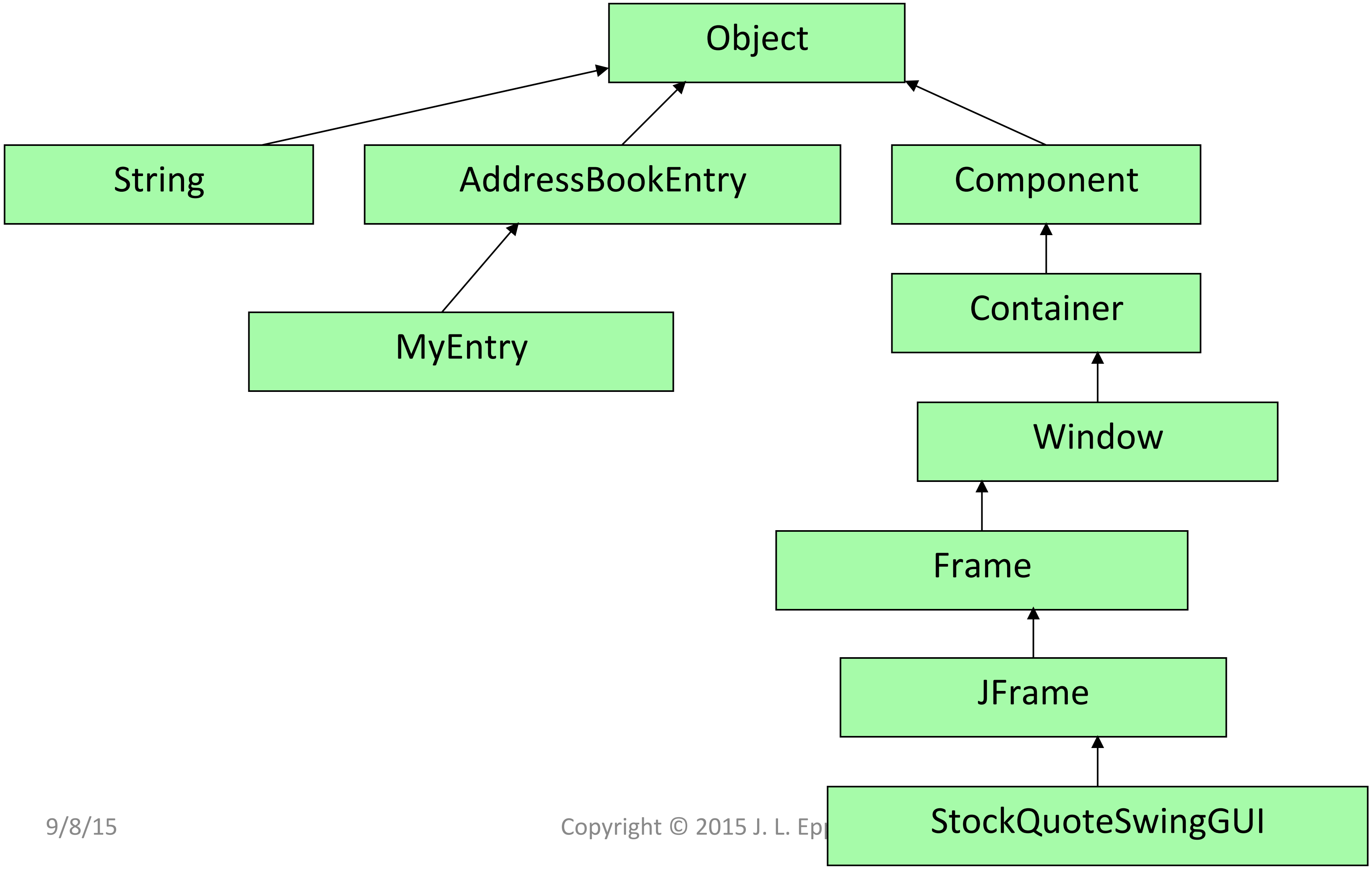
```
public class MyEntry extends AddressBookEntry {
    int    age;

    void birthday() {
        age = age + 1;
    }

    void print() {
        super.print();
        System.out.println("Age:    " + age);
        System.out.println();
    }

    // fixed up main removed for space
}
```

StockQuoteSwingGUI (from Lecture 1)

Code excerpt from **StockQuoteSwingGUI.java:**

```
public class StockQuoteSwingGUI extends JFrame … {
    private JButton      button;
    private JTextField   textField;
    private JTextArea    textArea;

    private java.util.ArrayList quoteHistory;

    public StockQuoteSwingGUI() {
        …
    }

    …

    public static void main(String[] args) {
        new StockQuoteSwingGUI();
    }
}
```

The Class Hierarchy



Object

String

AddressBookEntry

Component

MyEntry

Container

Window

Frame

JFrame

StockQuoteSwingGUI

Java Class Library

- Many Classes are included in Java
- See the Java Docs
- Examples:
  - java.lang.String
  - java.lang.StringBuilder

How would you implement StringBuilder?

```
public class StringBuilder {
    private String buffer;

    public void append(String str) {
        buffer = buffer + str;
    }

    public void reverse() {
        …
    }

    public String toString() {
        return buffer;
    }
}
```

Not Really

Outline

✓ Administrative Issues

✓ Our First Class Discussion

⇨ Sample Final Exam Question

Sample Final Exam Question

- What is object-oriented programming?

Object-Oriented Programming

- Writing programs using objects
  - Objects that specify their (internal) data
  - Objects that specify the methods by which you can manipulate their data

Outline

✓Administrative Issues

✓Our First Class Discussion

✓Sample Final Exam Question

08-671 Lecture Topics

(subject to change – but only a little bit)

#1 Intro

#2 Primitive Types

#3 Java Classes

#4 Reference Types

#5 Loops & Arrays

#6 Lists & Sorting

#7 Maps

#8 File & Network I/O

#9 Swing Interfaces

#10 Swing Actions

#11 Threads

#12 Exceptions

#13 Functional Programming

#14 In-class Written Exam

* Final Exam – this will be a 3-hour programming problem

Homework #3

- Will be released on Thursday