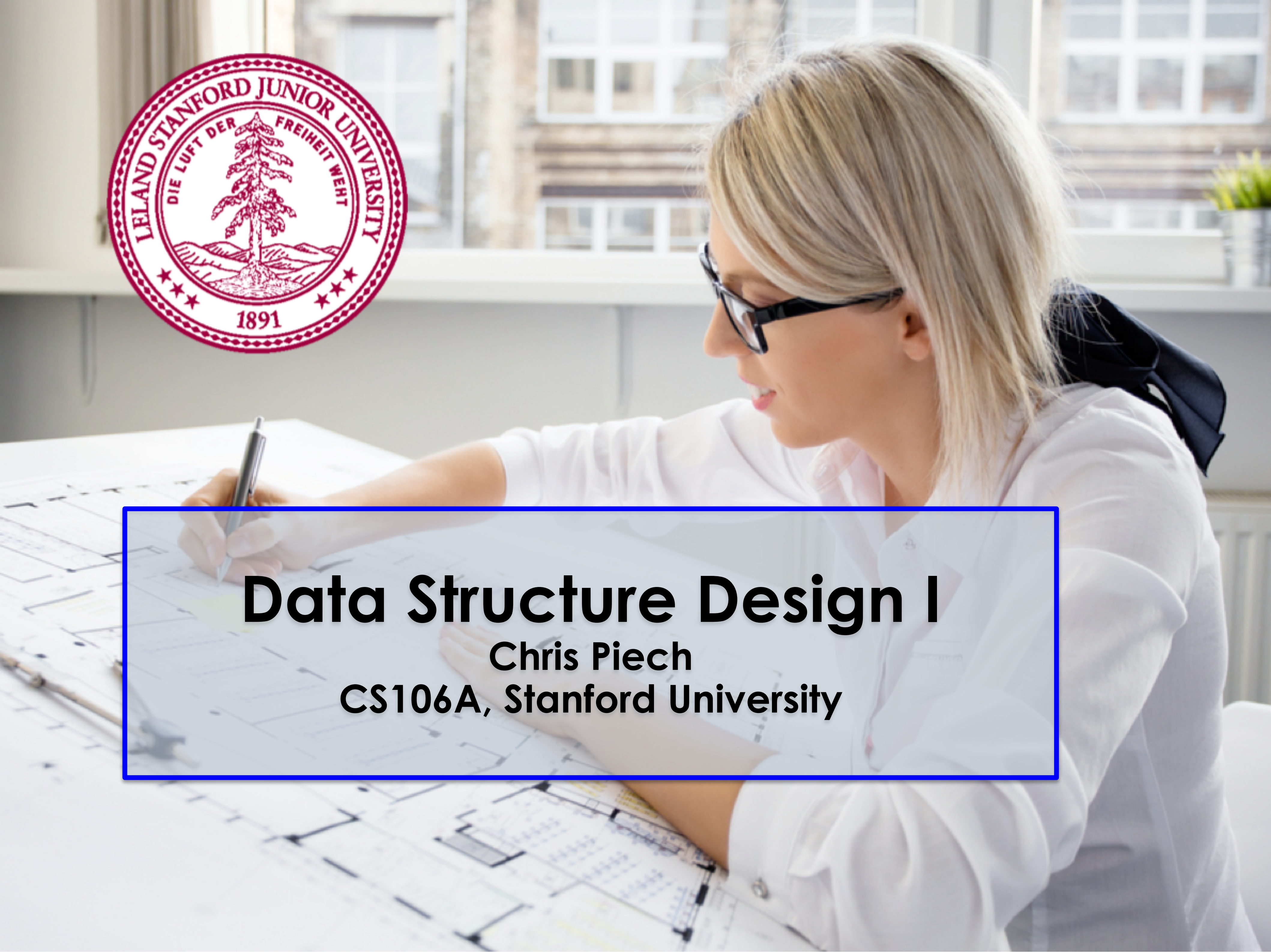
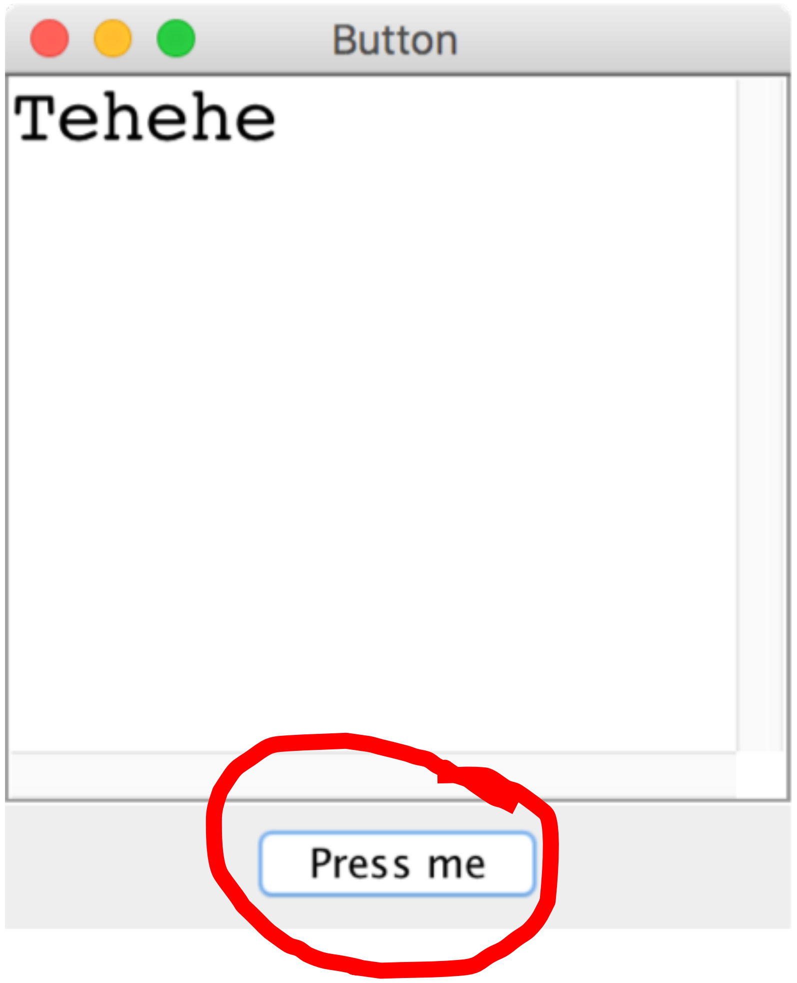


Data Structure Design I

Chris Piech
CS106A, Stanford University

Interactors

Button



```
Tehehe
```

JButton

JButton

```java
JButton button = new JButton("Press me");
```

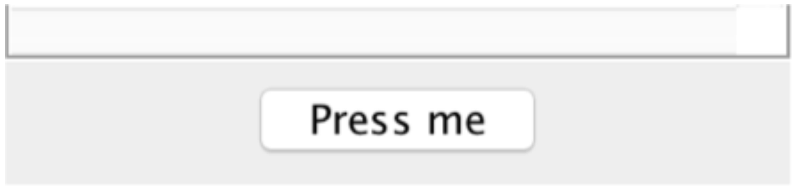
JButton

Button Text

```
JButton button = new JButton("Press me");
```

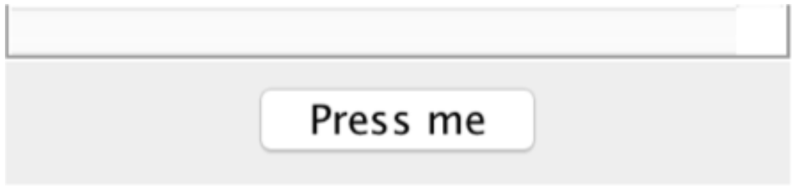
JButton

```
JButton button = new JButton("Press me");
add(button, SOUTH);
```



JButton

```
JButton button = new JButton("Press me");
add(button, SOUTH);
addActionListeners();
```



JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals(“Press me”)) {
      println(“Tehehe”);
   }
}
```

JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals(“Press me”)) {
      println(“Tehehe”);
   }
}
```

JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals(“Press me”)) {
      println(“Tehehe”);
   }
}
```

JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals("Press me")) {
      println("Tehehe");
   }
}
```

End review

Some *large* programs are in Java

Apache OpenOffice™

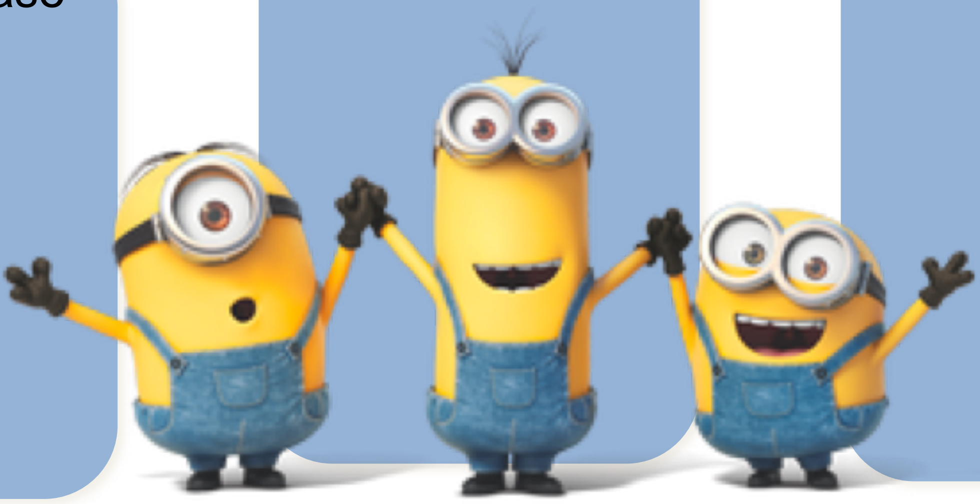
How?

Define New Variable Types

Inbox Database

Email Sender

Login Manager



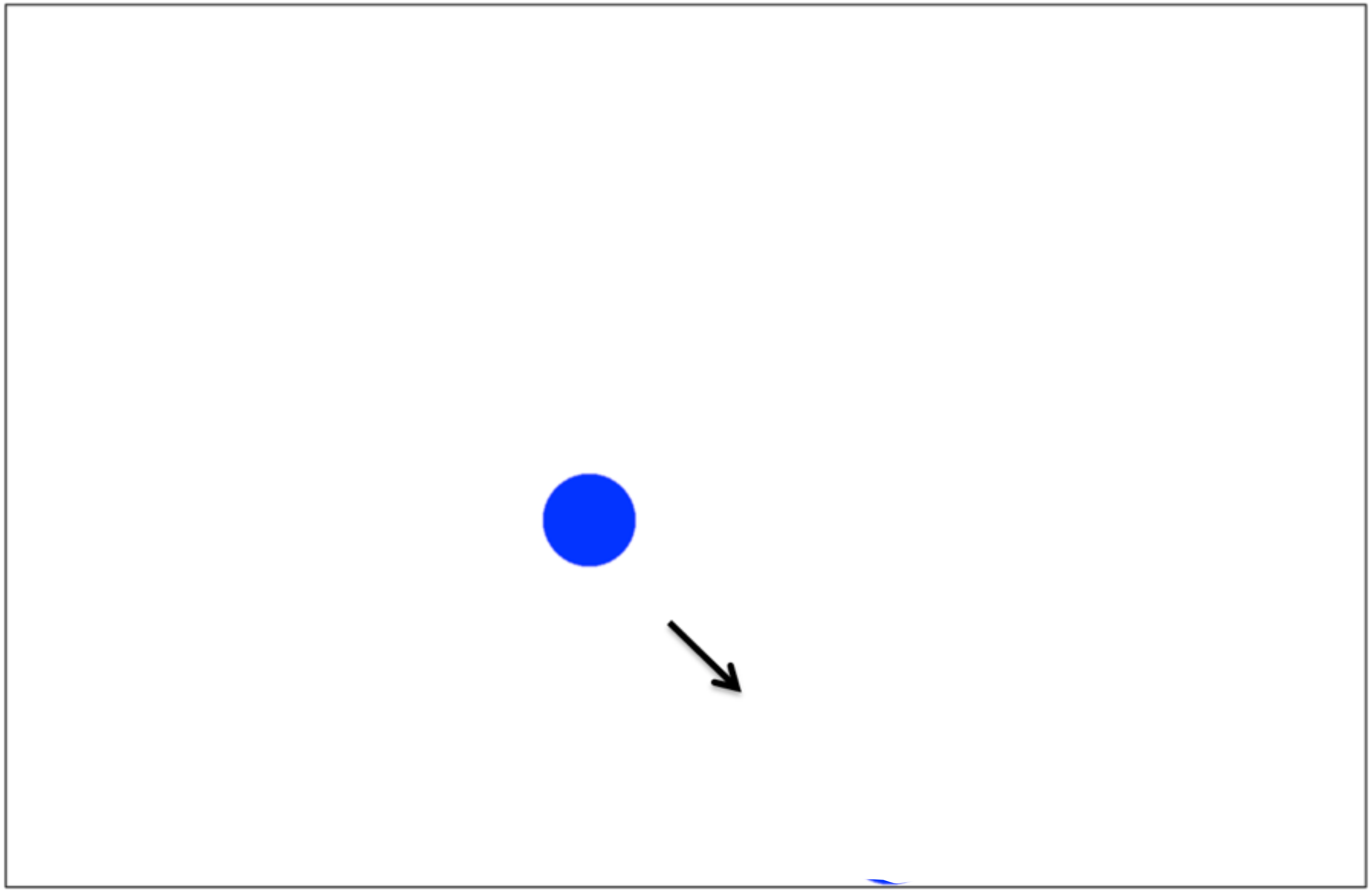
Email

User

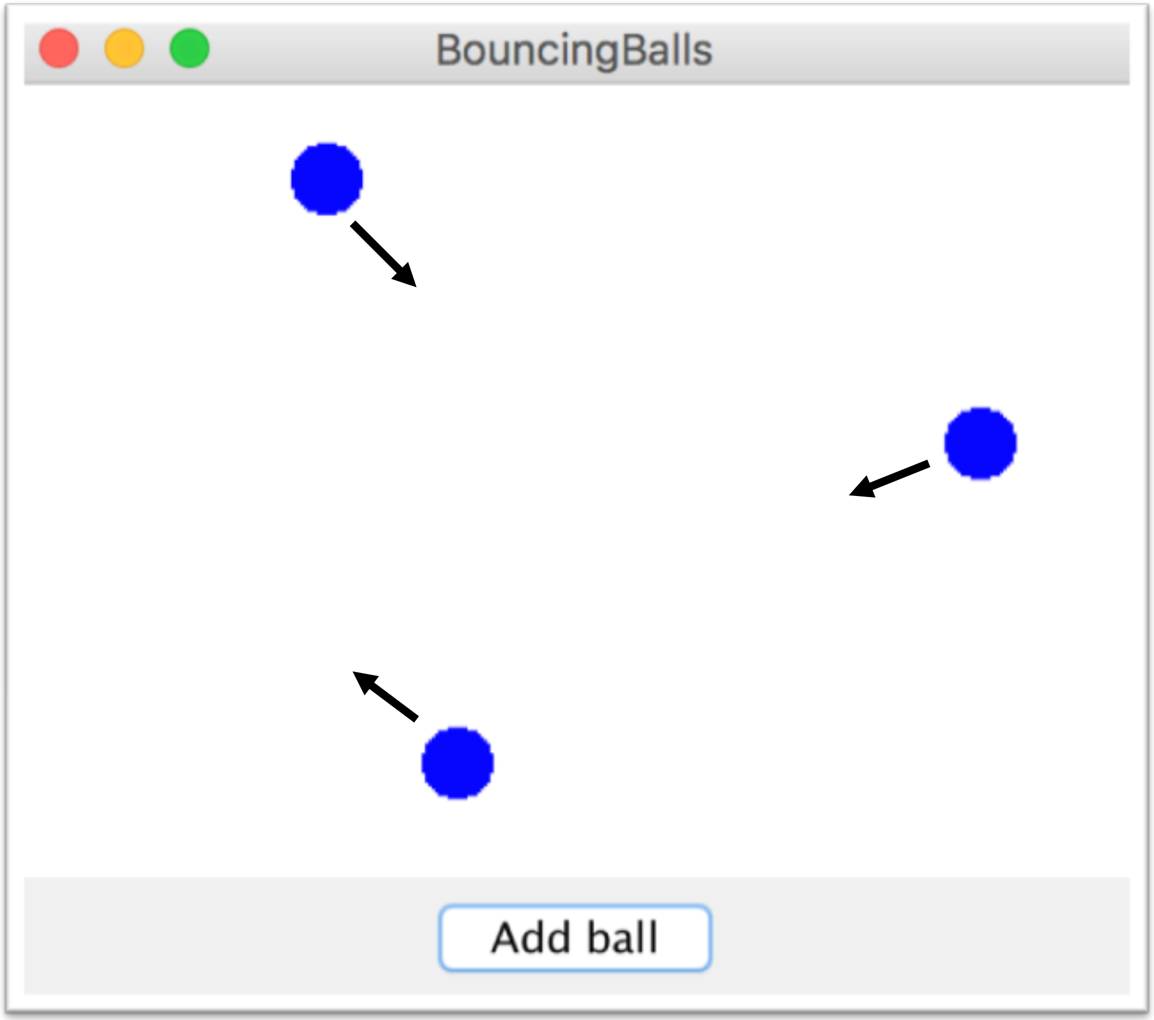
Inbox

Even small programs
define new variable types

Can you do this?



Bouncing Balls



Classes define new Variable Types

- A student registration system needs to store info about students, but Java has no **`Student`** type.

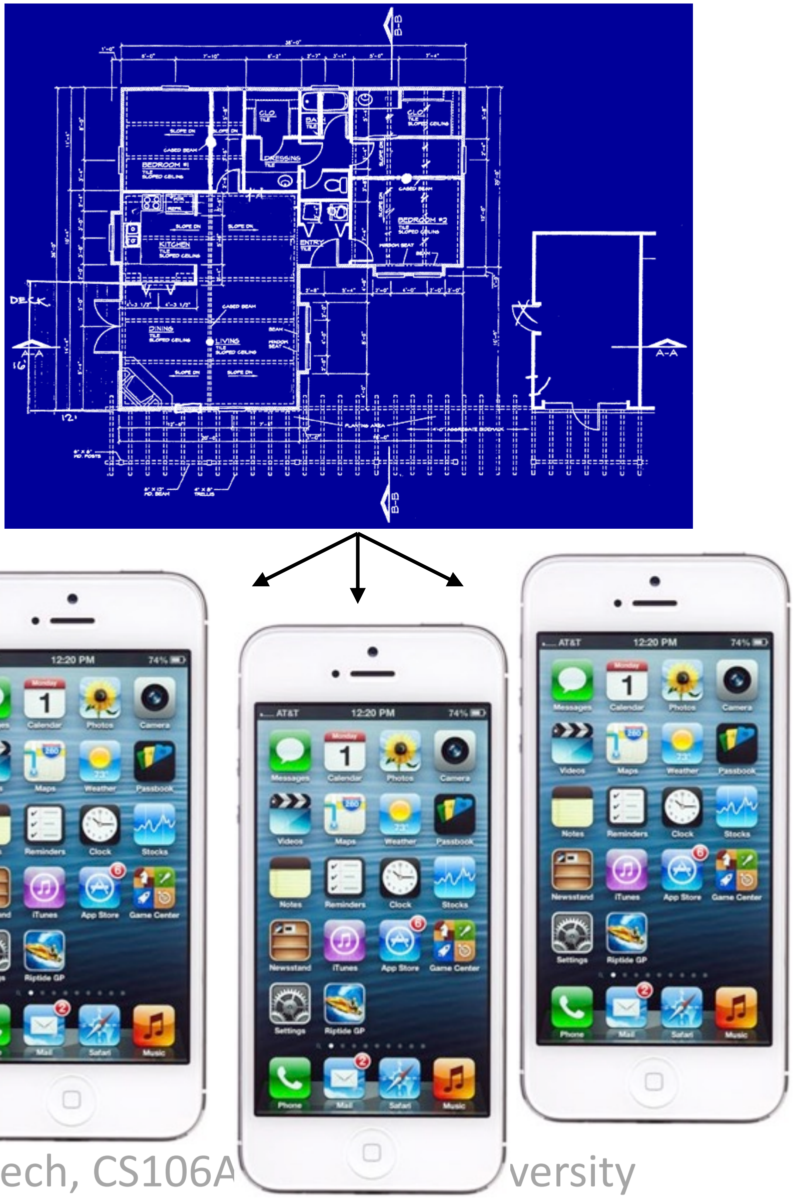
- A music synthesizer app might want to store information about users' accounts, but Java has no **`Instrument`** type.

- However, Java does provide a feature for us to add new data types to the language: **classes**.
  - Writing a class defines a new data type.

Classes are like blueprints

class: A template for a new type of variable.

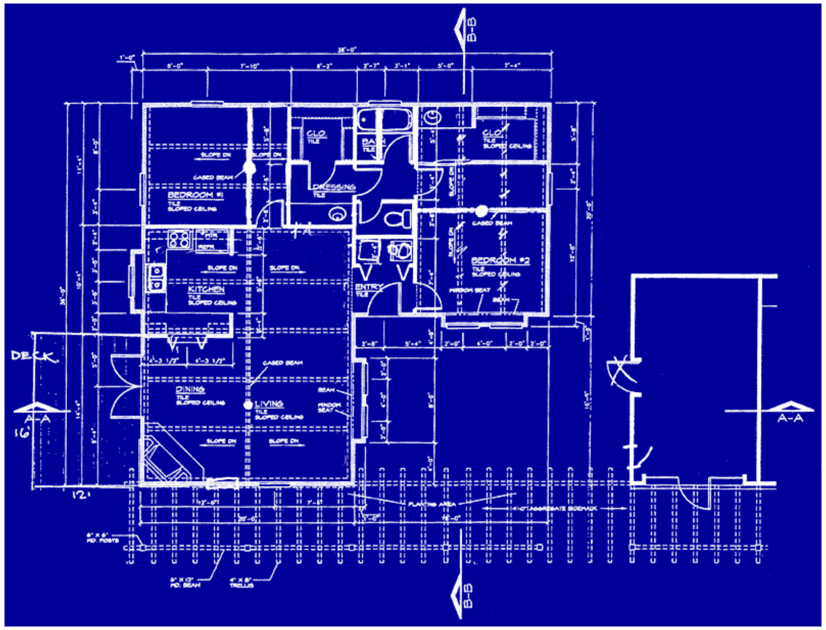
A blueprint is a helpful analogy



#key: Classes define new variable *types*

#key: Classes decompose your program across files

Classes are like blueprints



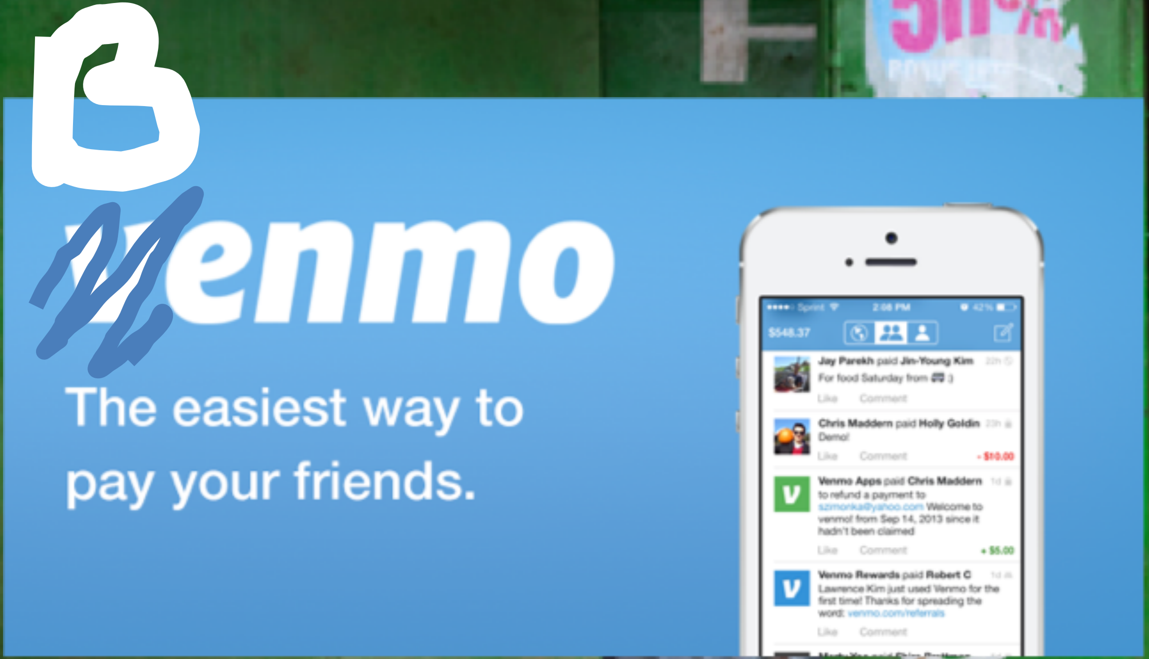
To design a new variable type you must specify three things:

1. What subvariables make up this new variable type?
2. What methods can you call on a variable of this type?
3. What happens when you make a new instance of this type?

What is a class?

A class defines a new variable type

Kenya has used mobile banking for 10 years



Jay Parekh paid Jin-Young Kim 22h
For food Saturday from 🚌 ;)

Chris Maddern paid Holly Goldin 23h
Demo!
- $10.00

Venmo Apps paid Chris Maddern 1d
to refund a payment to szimonka@yahoo.com Welcome to venmo! from Sep 14, 2013 since it hadn't been claimed
+ $5.00

Venmo Rewards paid Robert C 1d
Lawrence Kim just used Venmo for the first time! Thanks for spreading the word: venmo.com/referrals

Classes: Take 1

This goes in BankAccount.java!

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

Instance variables have a special meaning

Classes: Take 1

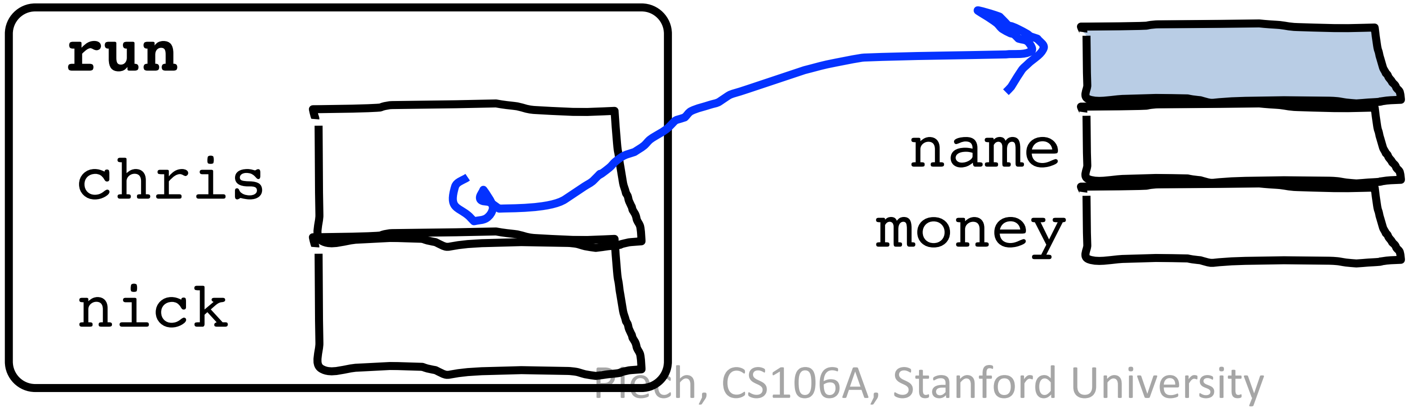
```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```

Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

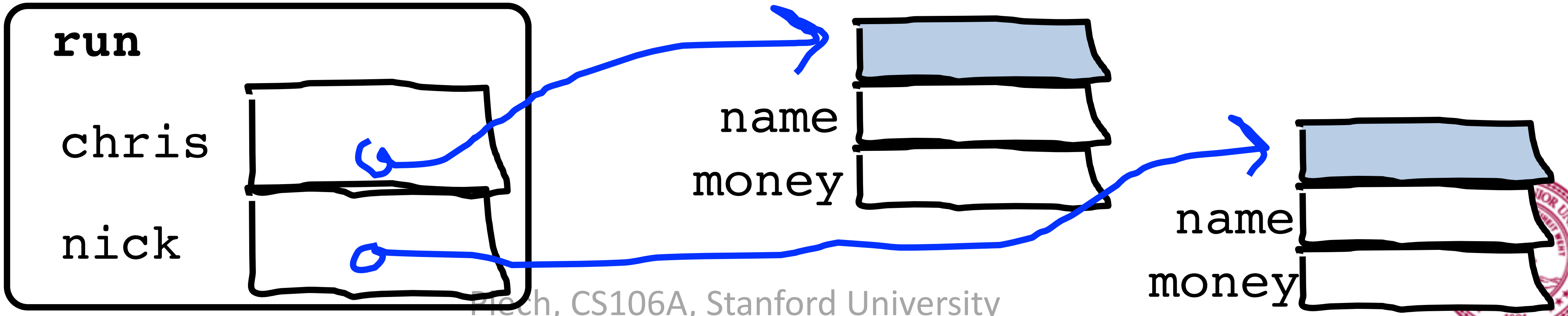
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

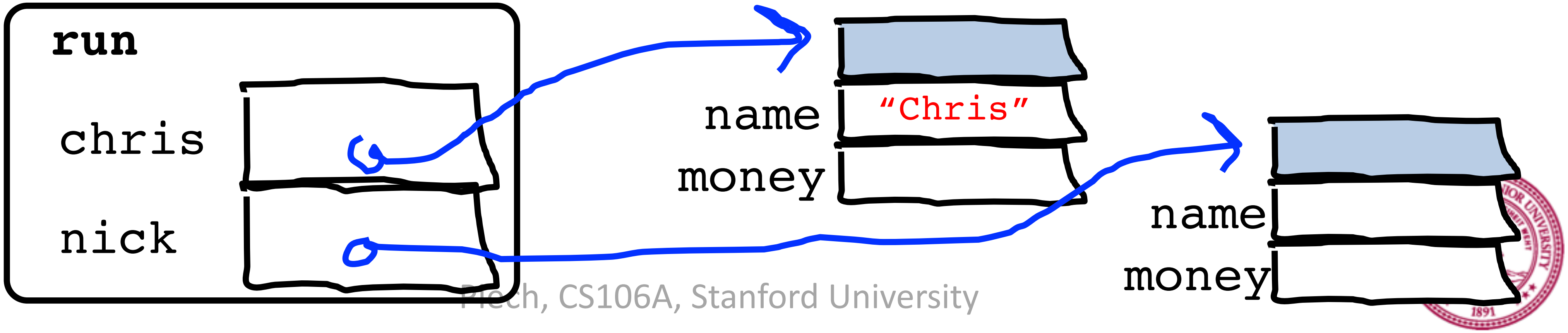
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

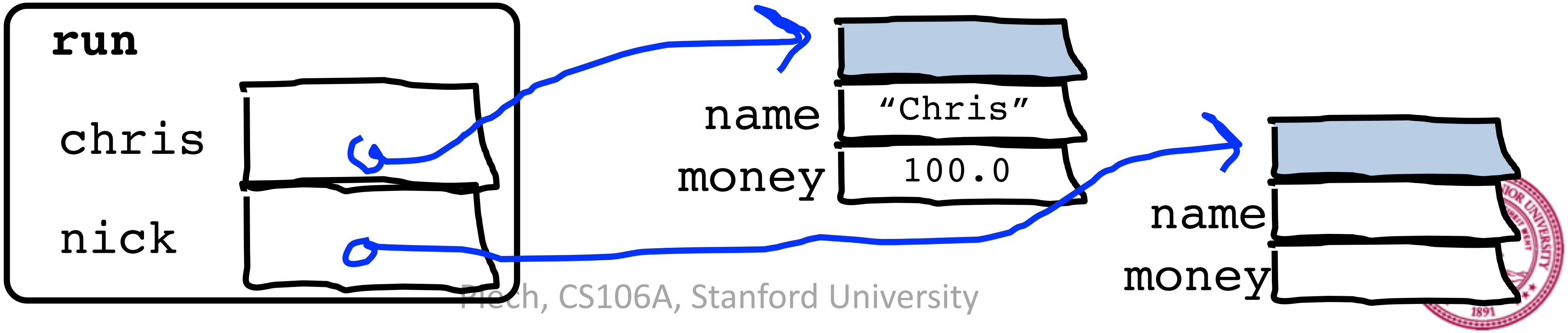
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

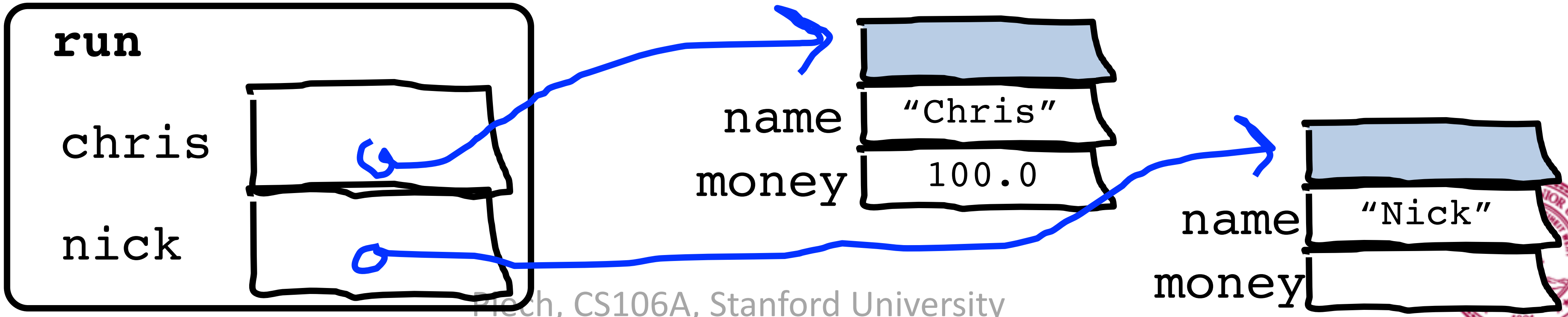
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

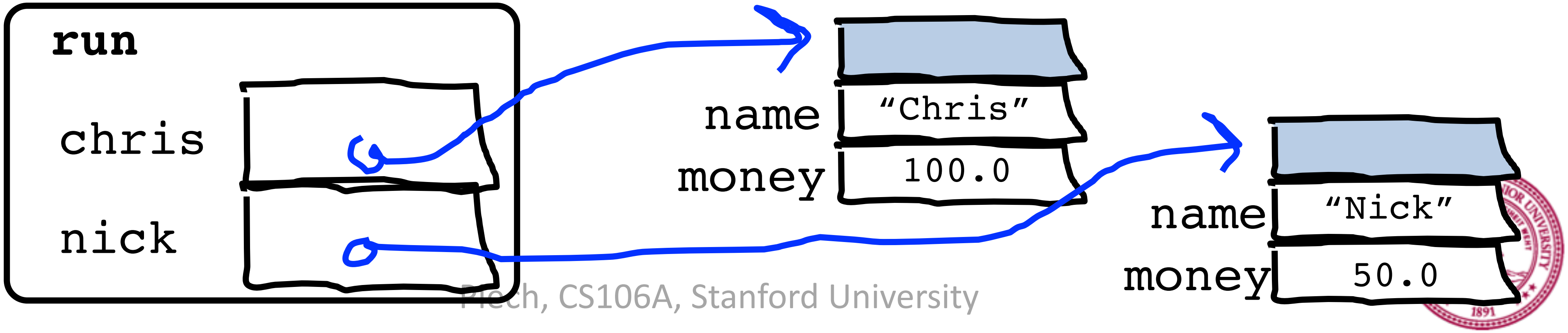
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

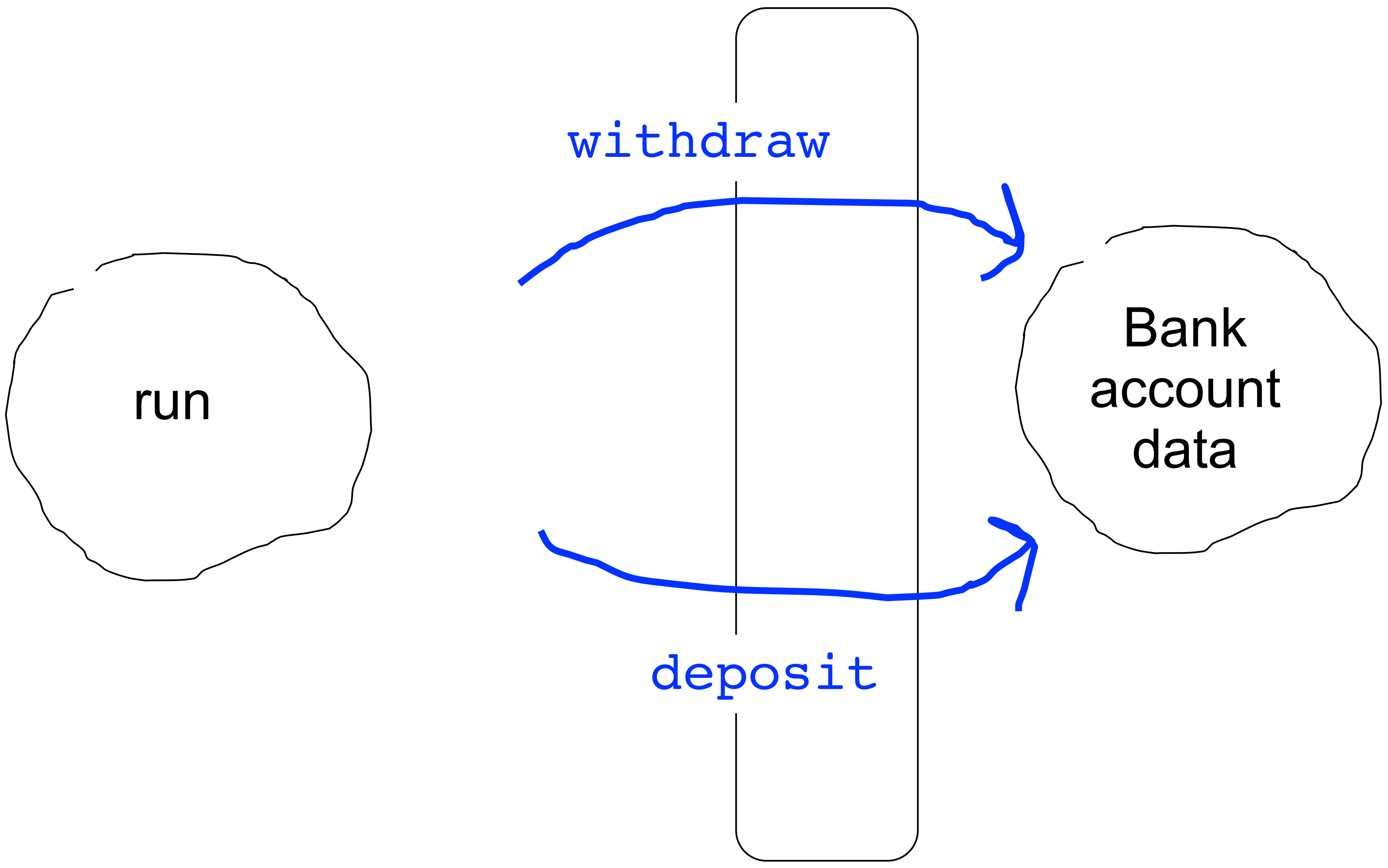
```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



What is a class?

A class defines a new variable type



Wall of abstraction

Adding Privacy

```
private double money;
```

- **encapsulation**: Hiding implementation details of an object from its clients.
  - Encapsulation provides *abstraction*.
    - separates external view (behavior) from internal view (state)
  - Encapsulation protects the integrity of an object's data.

- A class's instance variables should be declared *private*.
  - No code outside the class can access or change it.

Classes: Take 2

This goes in its own file!

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

Instance variables have a special meaning

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   public String name;
   public double money;
}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;
}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      ...
   }

   public boolean withdraw(double amount) {
      ...
   }

}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      money += amount;
   }

   public boolean withdraw(double amount) {
      ...
   }

}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      ...
   }

}
```

`this`

Classes: Take 2

```java
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      ...
   }

}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      if(amount <= this.money) {
         this.money -= amount;
         return true;
      }
      return false;
   }

}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      if(amount <= this.money) {
         this.money -= amount;
         return true;
      }
      return false;
   }

    // 3. How do you make a new one?
   public BankAccount(String name, double amount) {
      this.money = amount;
      this.money = name;
   }
}
```

$548.37

Jay Parekh paid Jin-Young Kim 22h
For food Saturday from 🚌 :)
Like Comment

Chris Maddern paid Holly Goldin 23h
Demo!
Like Comment - $10.00

Venmo Apps paid Chris Maddern 1d
to refund a payment to szimonka@yahoo.com Welcome to venmo! from Sep 14, 2013 since it hadn't been claimed
Like Comment + $5.00

Venmo Rewards paid Robert C 1d
Lawrence Kim just used Venmo for the first time! Thanks for spreading the word: venmo.com/referrals
Like Comment

You must define three things

1. What **variables** does each instance store?

2. What **methods** can you call on an instance?

3. What happens when you make a `new` one?

Classes on one slide

1. What variables make up this new super variable type?

   Instance variables

2. What methods can you call on a variable of this type?

   It's public methods

3. What happens when the user makes a **`new`** instance?

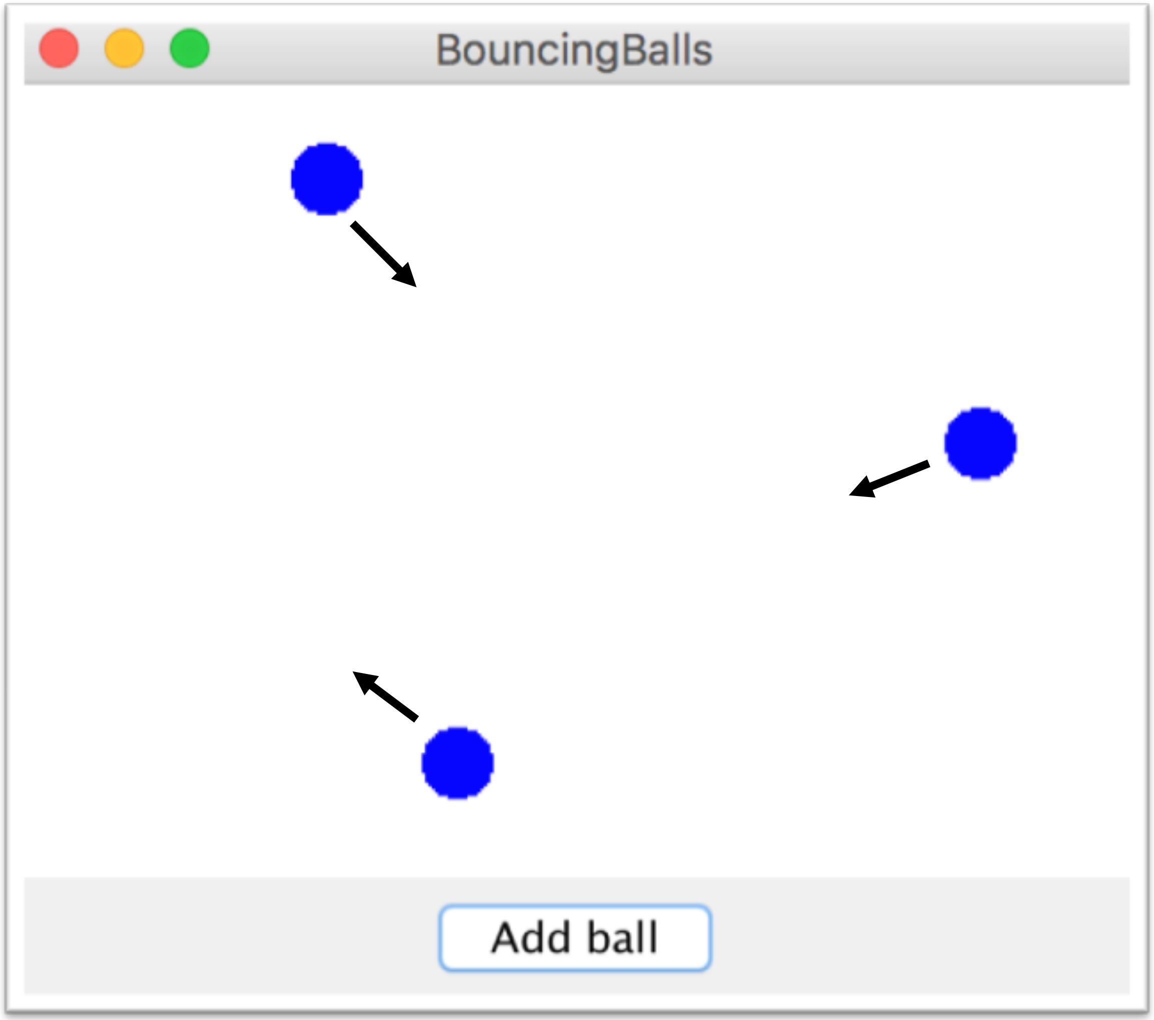
   The "constructor"

* Don't forget that all methods and constructors have access to a **`this`** reference

What is a class?

A class defines a new variable type

Bouncing Balls



A Ball Variable Type

The Ball class

1. What **variables** does each instance store?

   - Each ball has its own Goval (lets call it shape)
   - Each ball has its own dx
   - Each ball has its own dy

2. What **methods** can you call on an instance?

   - `heartbeat();`
   - `getShape();`

3. What happens when you make a **`new`** one?

   - Sets initial values for all the ”instance” vars

*details on how to define these three things coming soon

What classes?

What classes?