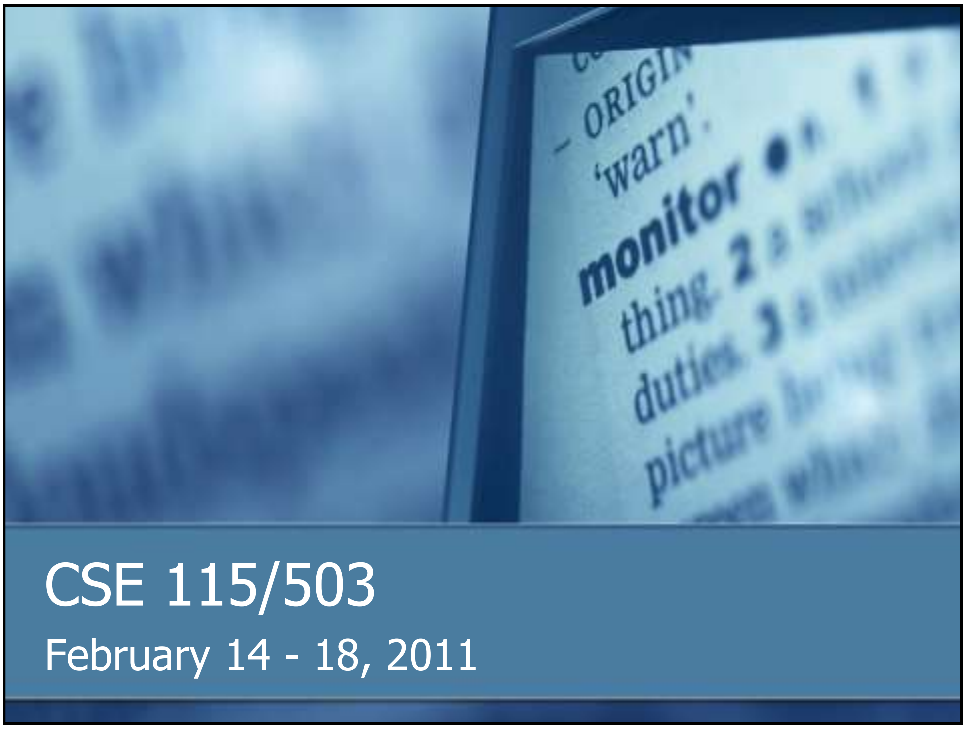
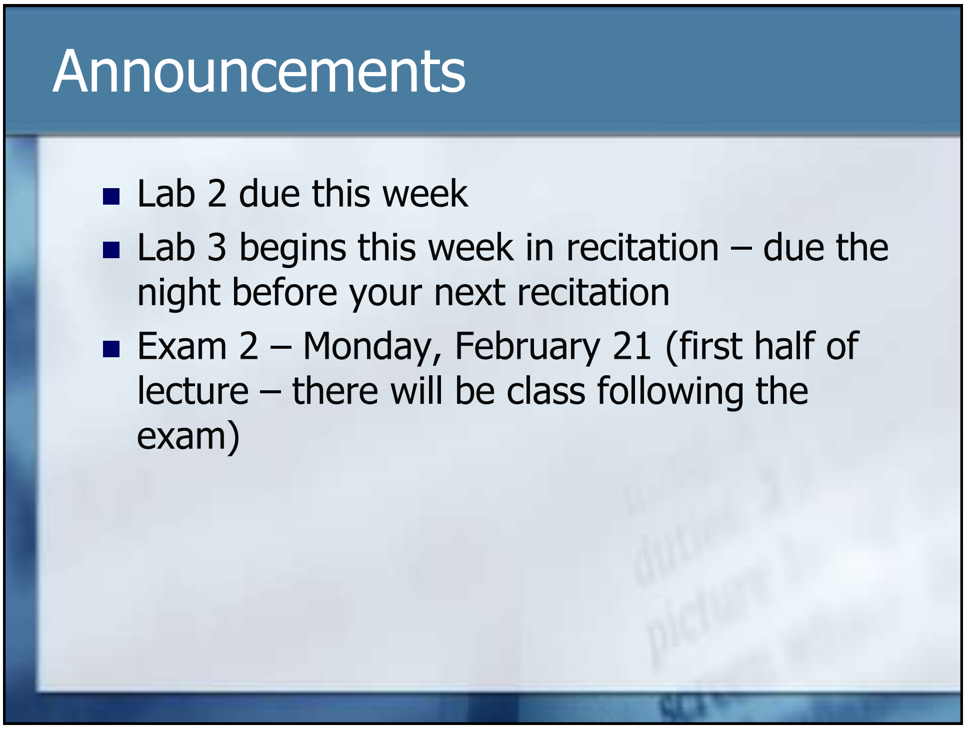


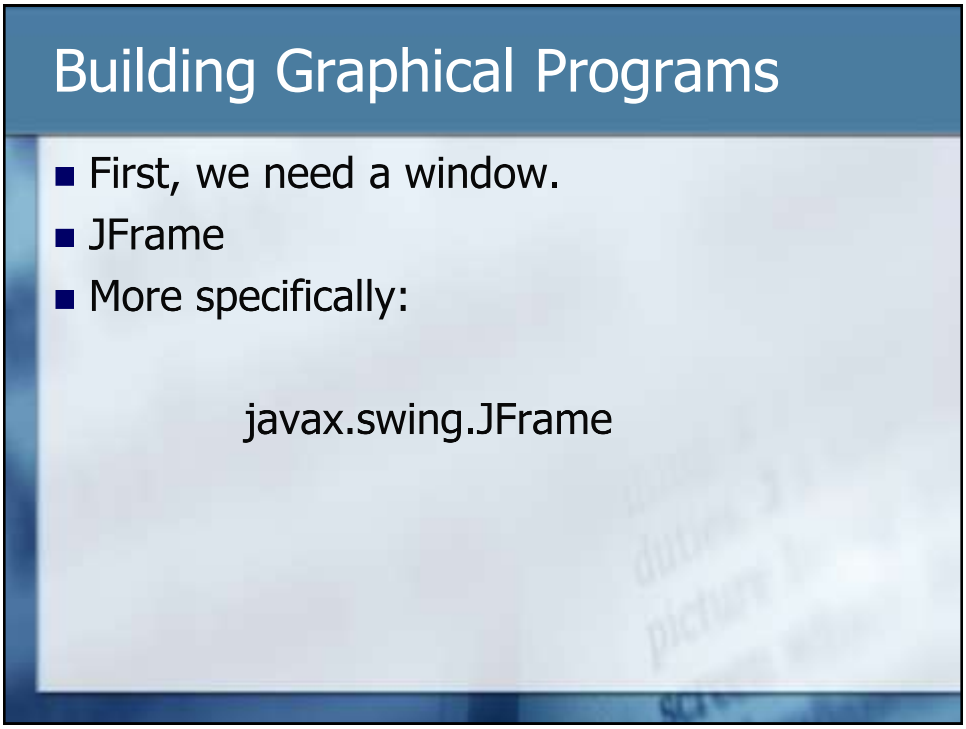
CSE 115/503

February 14 - 18, 2011



Announcements

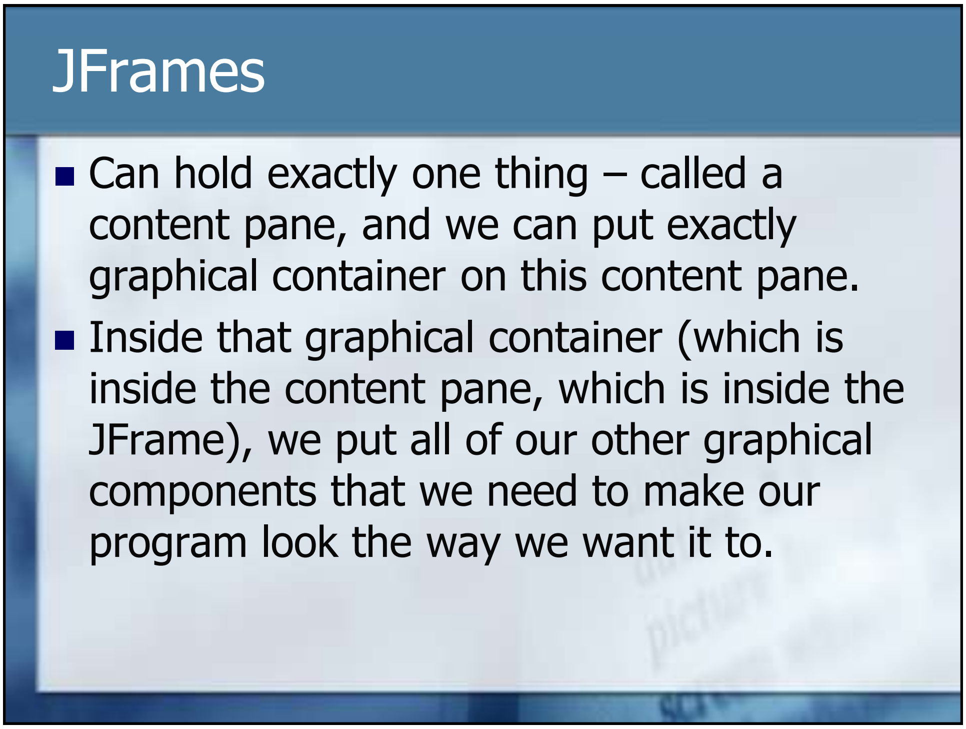
- Lab 2 due this week
- Lab 3 begins this week in recitation – due the night before your next recitation
- Exam 2 – Monday, February 21 (first half of lecture – there will be class following the exam)



Building Graphical Programs

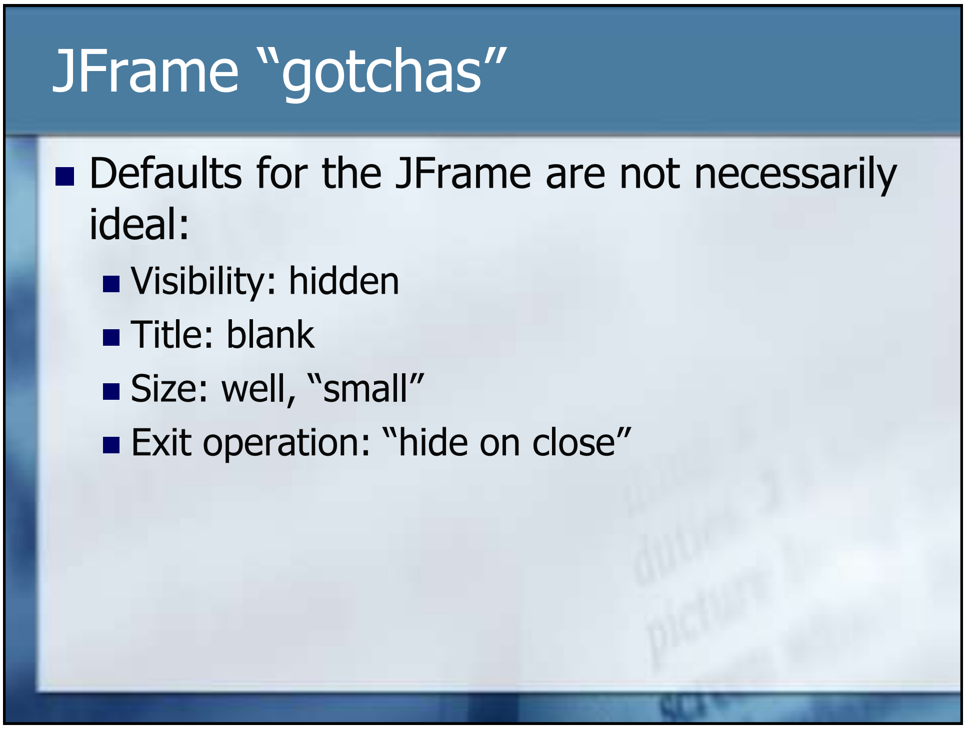
- First, we need a window.
- JFrame
- More specifically:

javax.swing.JFrame



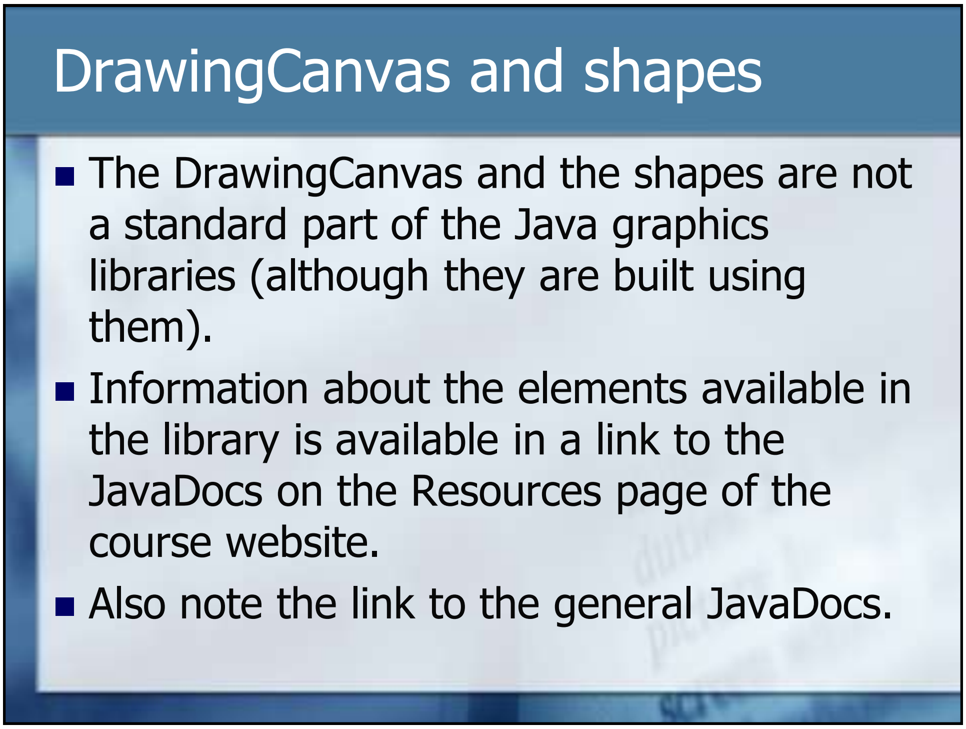
JFrames

- Can hold exactly one thing – called a content pane, and we can put exactly graphical container on this content pane.
- Inside that graphical container (which is inside the content pane, which is inside the JFrame), we put all of our other graphical components that we need to make our program look the way we want it to.



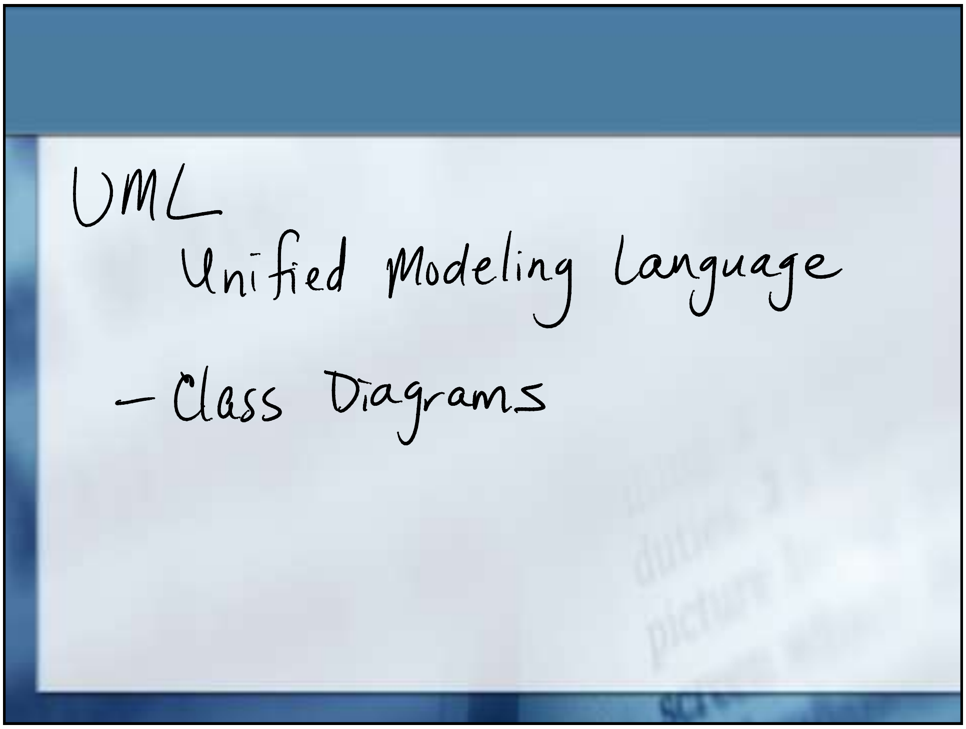
JFrame "gotchas"

- Defaults for the JFrame are not necessarily ideal:
  - Visibility: hidden
  - Title: blank
  - Size: well, "small"
  - Exit operation: "hide on close"



DrawingCanvas and shapes

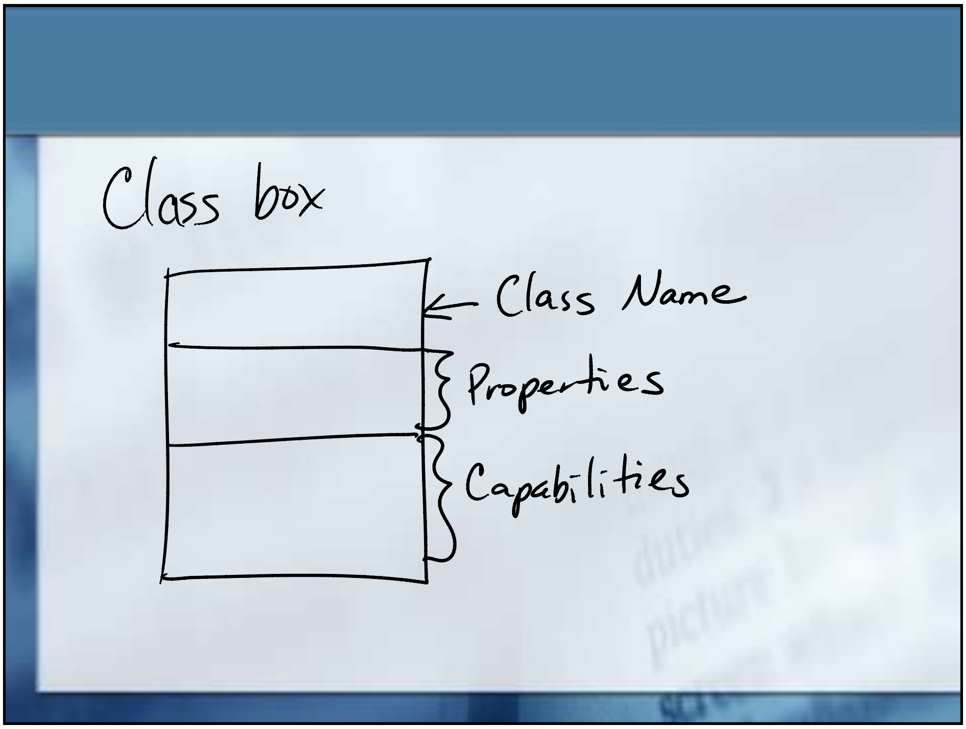
- The DrawingCanvas and the shapes are not a standard part of the Java graphics libraries (although they are built using them).
- Information about the elements available in the library is available in a link to the JavaDocs on the Resources page of the course website.
- Also note the link to the general JavaDocs.



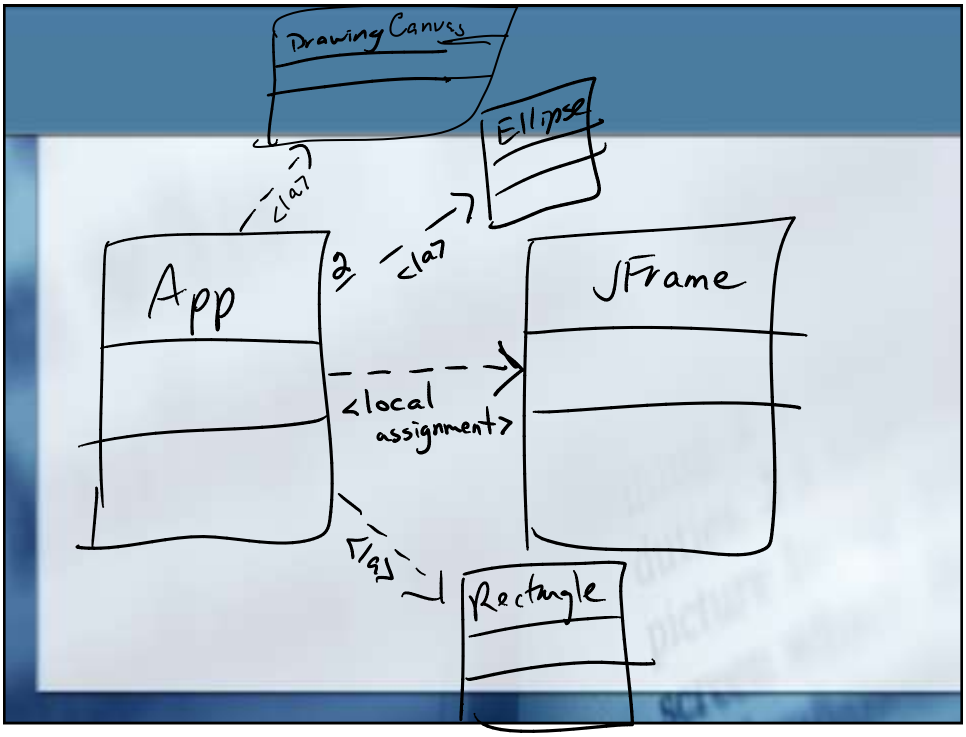
UML

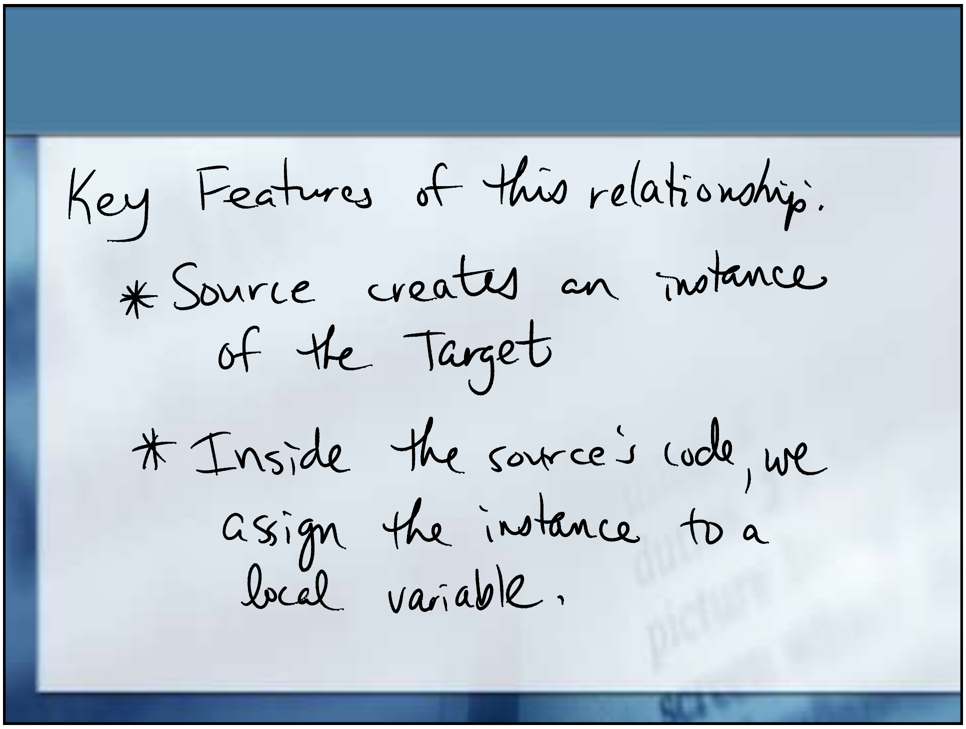
Unified Modeling Language

- Class Diagrams



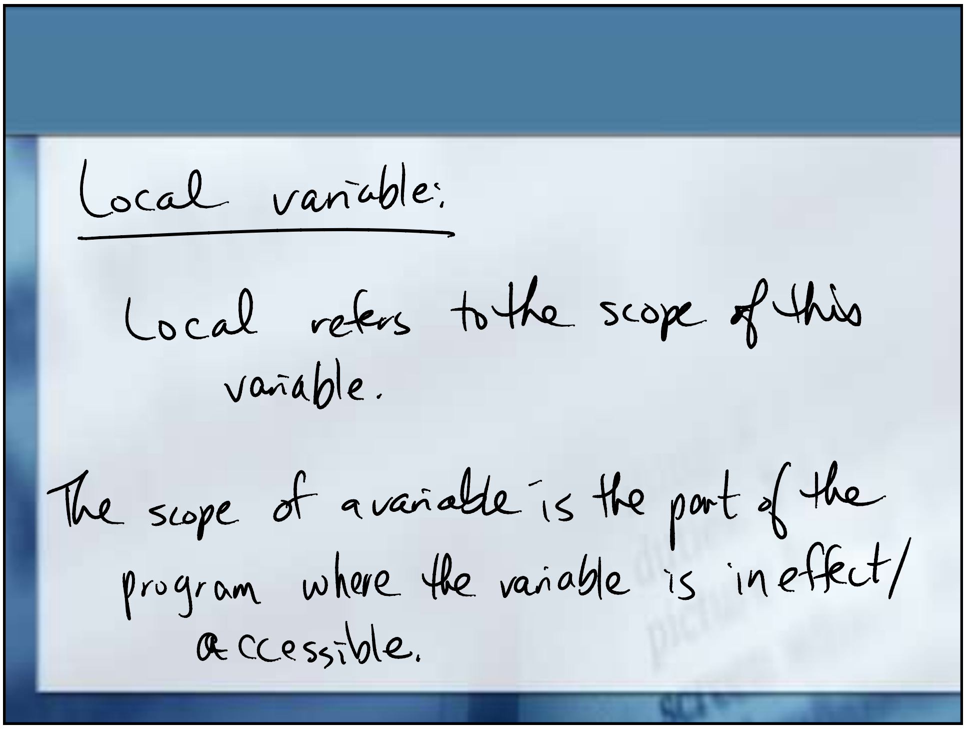
Class box





Key Features of this relationship:

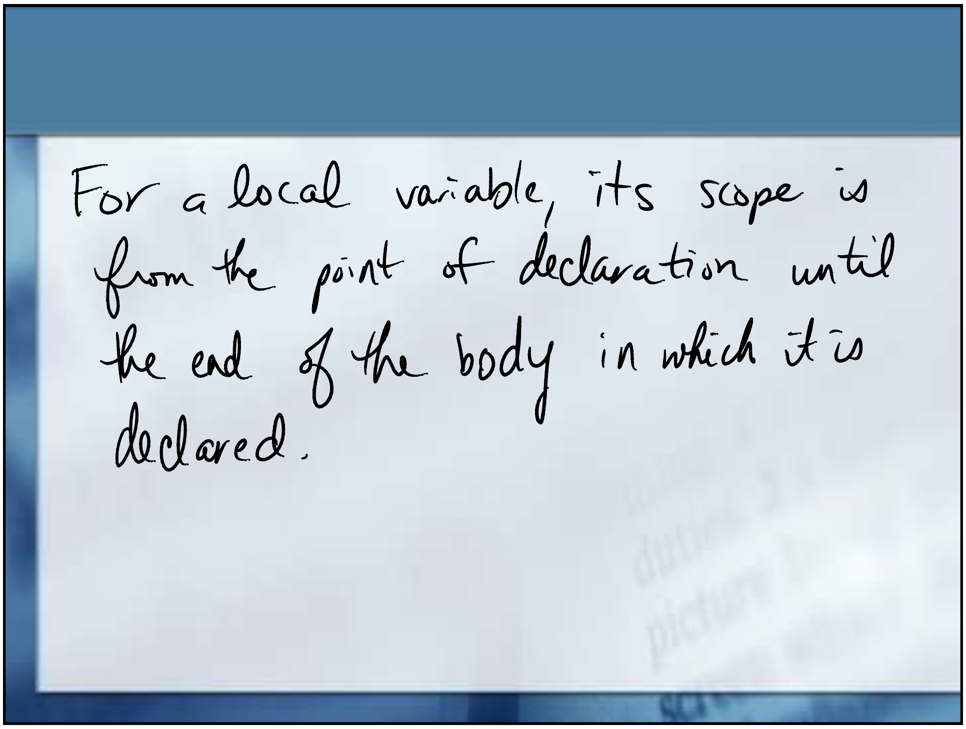
- Source creates an instance of the Target
- Inside the source's code, we assign the instance to a local variable.



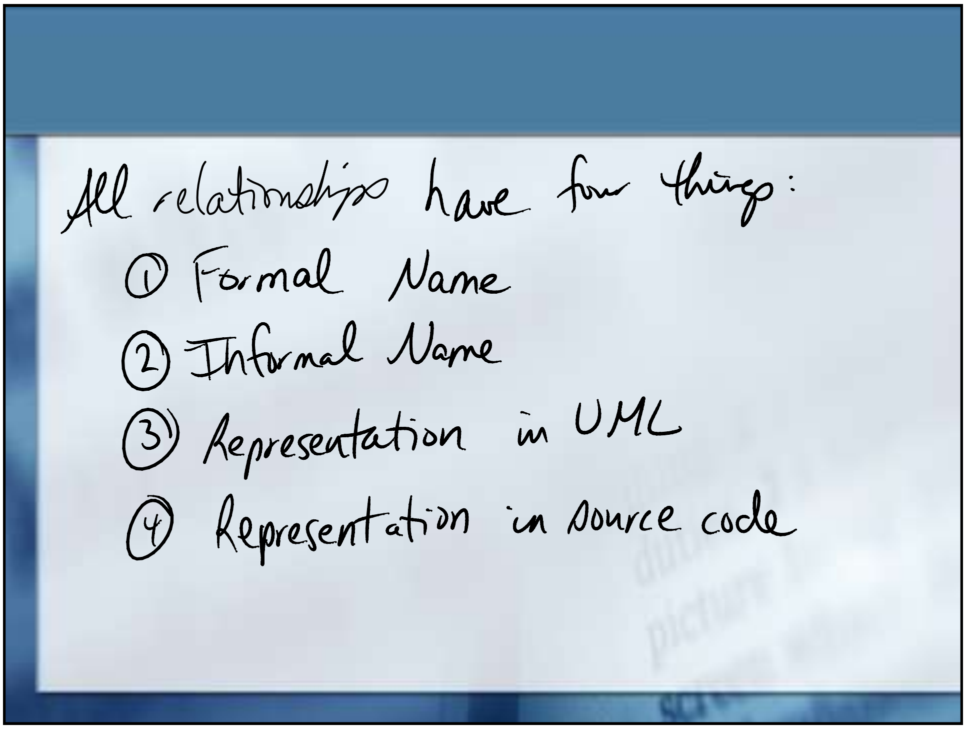
Local variable:

Local refers to the scope of this variable.

The scope of a variable is the part of the program where the variable is in effect/ accessible.

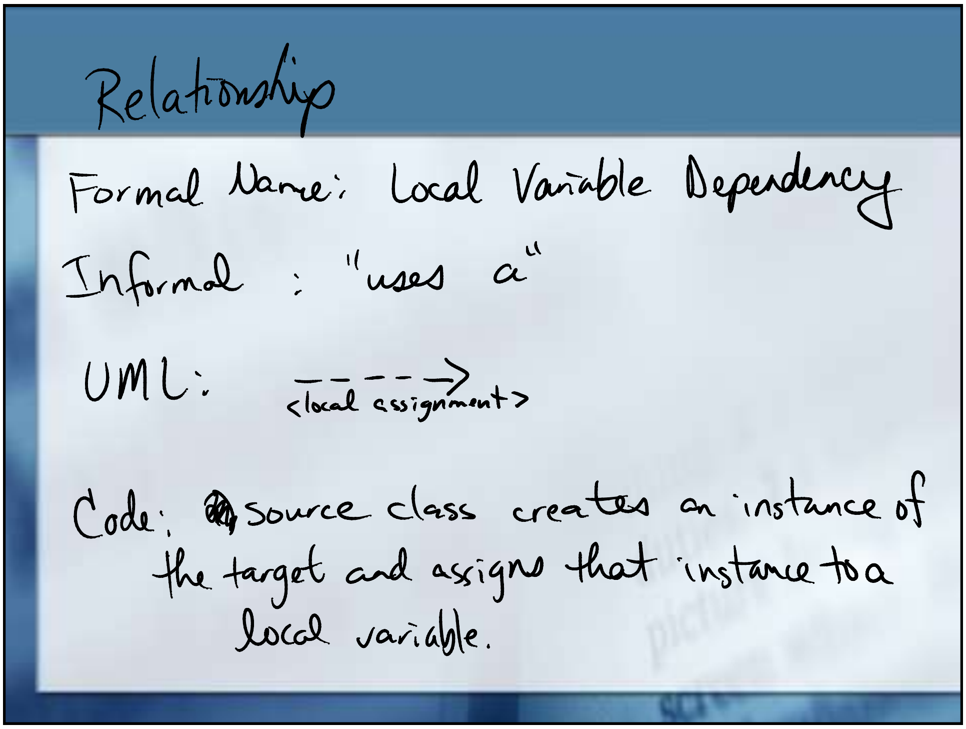


For a local variable, its scope is from the point of declaration until the end of the body in which it is declared.



All relationships have four things:

1. Formal Name
2. Informal Name
3. Representation in UML
4. Representation in source code



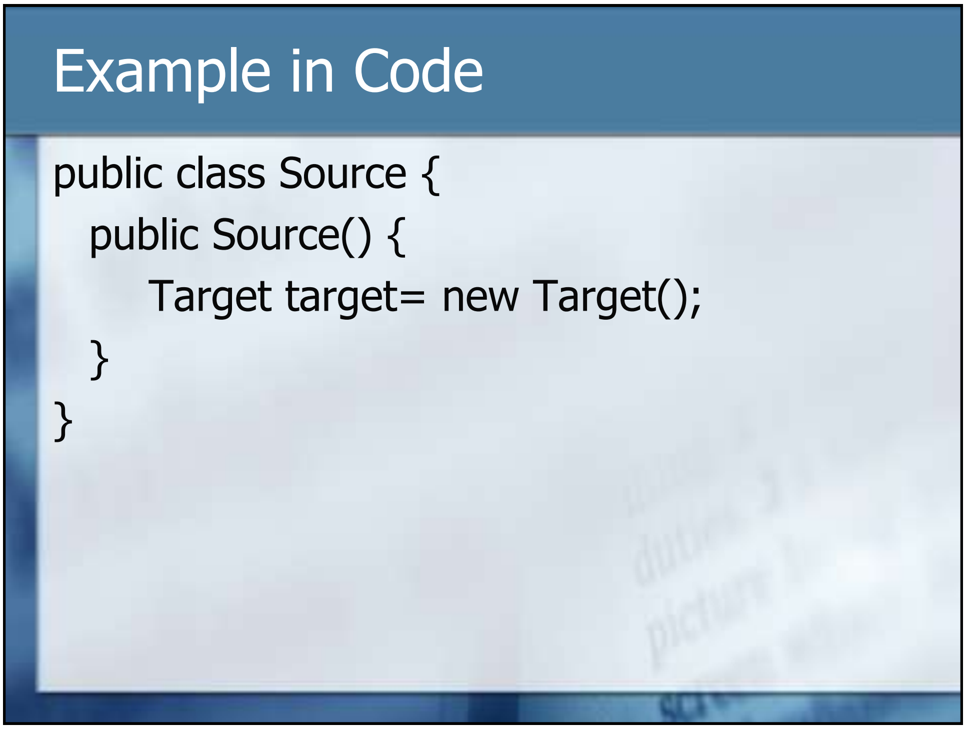
Relationship

Formal Name: Local Variable Dependency

Informal : "uses a"

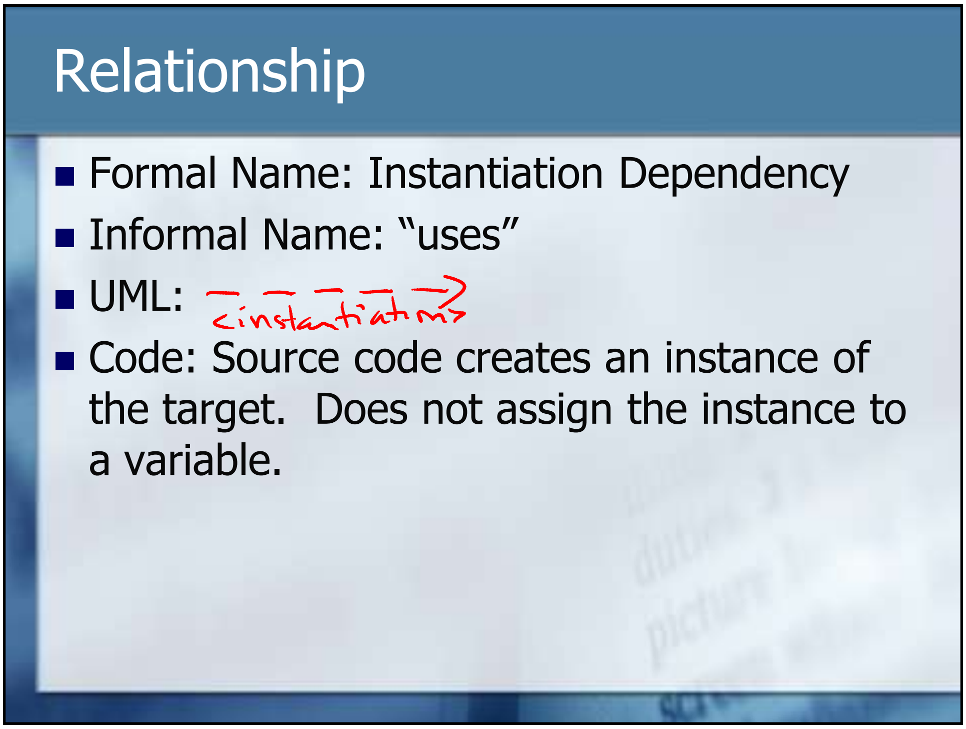
UML: $\underset{\text{<local assignment>}}{\dashrightarrow}$

Code: source class creates an instance of the target and assigns that instance to a local variable.



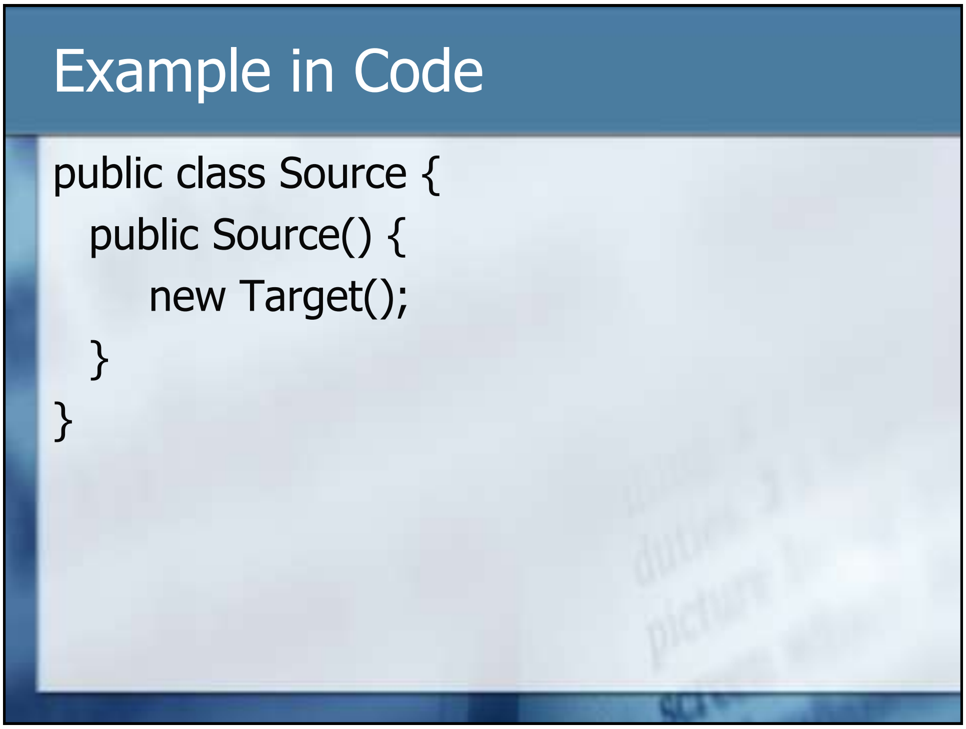
Example in Code

```
public class Source {
  public Source() {
      Target target= new Target();
  }
}
```



Relationship

- Formal Name: Instantiation Dependency
- Informal Name: “uses”
- UML: - - - - - -> <instantiation>
- Code: Source code creates an instance of the target. Does not assign the instance to a variable.



Example in Code

```
public class Source {
   public Source() {
        new Target();
   }
}
```