

# Objectives

- Course overview
- Picasso

Looks like we  
(almost) made it!

What have you learned this semester?

What are you taking with you?

**OH, THE PLACES YOU HAVE BEEN!**

# Review

- What have you learned this semester?
- Where will you go from here?
  - What do you think you're most likely to take with you?
  - What are you going to update on your resume?
  - What will your design philosophy/development process be?

# Review: What to Expect from this Class

- Programming intensive
  - Variety of assignments and projects
  - More freedom in design, \*ilities
    - Larger portion of your grade
    - Correctness is **NOT** enough
  - Building on large library of classes
  - Read others' code! Learn from the good and the bad
  - Building larger applications
- Compare/Contrast with Python
  - PL design; what's the best PL for your needs
- Learning on your own
  - Online resources

# Review: Learning Objectives

- Discuss software development and practices **knowledgably**, using appropriate **terminology**
- Design, implement, test, and document efficient applications of **increasing size** and **complexity** 
- Understand the designs and implementations of **others**
- Use a **version control system**
- Use many of the capabilities of the **Eclipse IDE**
- Test and debug large applications **systematically**, using standard tools
- Understand **design principles** such as DRY and shy
- Discuss the benefits and limitations of a **statically typed** language

# My Philosophy

- Balance imparting knowledge and creating learning experiences
- Goals
  - Help you recognize bad design, fixes for it
  - Learn to read others' code—not just mine
  - Transferrable skills
    - VCS, IDE use, abstraction, design
  - Best practices of Java
    - Small assignments on Java specifics
    - *Effective Java*

# WHERE WE ARE NOW

# New Functionality Request

- Scenario: The Picasso client has a new feature request!
- What functionality would be easy to add?
  - Why?
  - What design principles/patterns are being applied?
- What would be difficult to add?

# Picasso Best Practices

- You wrote (or should have written) JUnit test cases for tokenizing, parsing, and evaluating
- You changed your code
- Rerun your JUnit tests and make sure everything still works!

# Intermediate Implementation Deadline

- Demo the required content
  - Image-manipulation function
  - Assignment
  - Order of operations
  - Error handling
  - Reading expression from a file
- Show me Picasso evaluating your favorite expressions you've generated so far (saved in files!)
- Talk about next steps
- Your questions

# Hints

- Check out the FAQ
- Create unit tests, when possible/appropriate
  - Run using coverage to identify gaps
- Draw things (e.g., stacks, trees) out on paper
- Trace through the code

# Project Deadlines

- Friday: Intermediate Implementation Deadline, before class
  - Demo, in class, format similar to last time
- Exam week
  - Final implementation (team), Thursday at 11:59 p.m.
  - Analysis (individual), Friday at noon (end of exam period)

# Looking Ahead

- Course Evaluations

- Due Monday, Dec 11

- Incentive to fill out evaluations

- If 60% fill out, 1% Extra Credit on “Individual programming and written homework assignments”
    - Additional 1% for every additional 10% (~2 students) who complete; max: 5%

- Next week’s Office Hours – by appointment