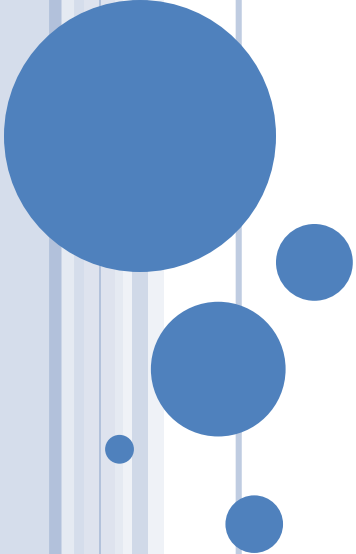




COMP 401

CONCLUDING REMARKS

Instructor: Prasun Dewan



COURSE → FUTURE

Project

Topics

Praxes

Diaries

Quizzes

Exam(s)

Advanced Courses

LAship

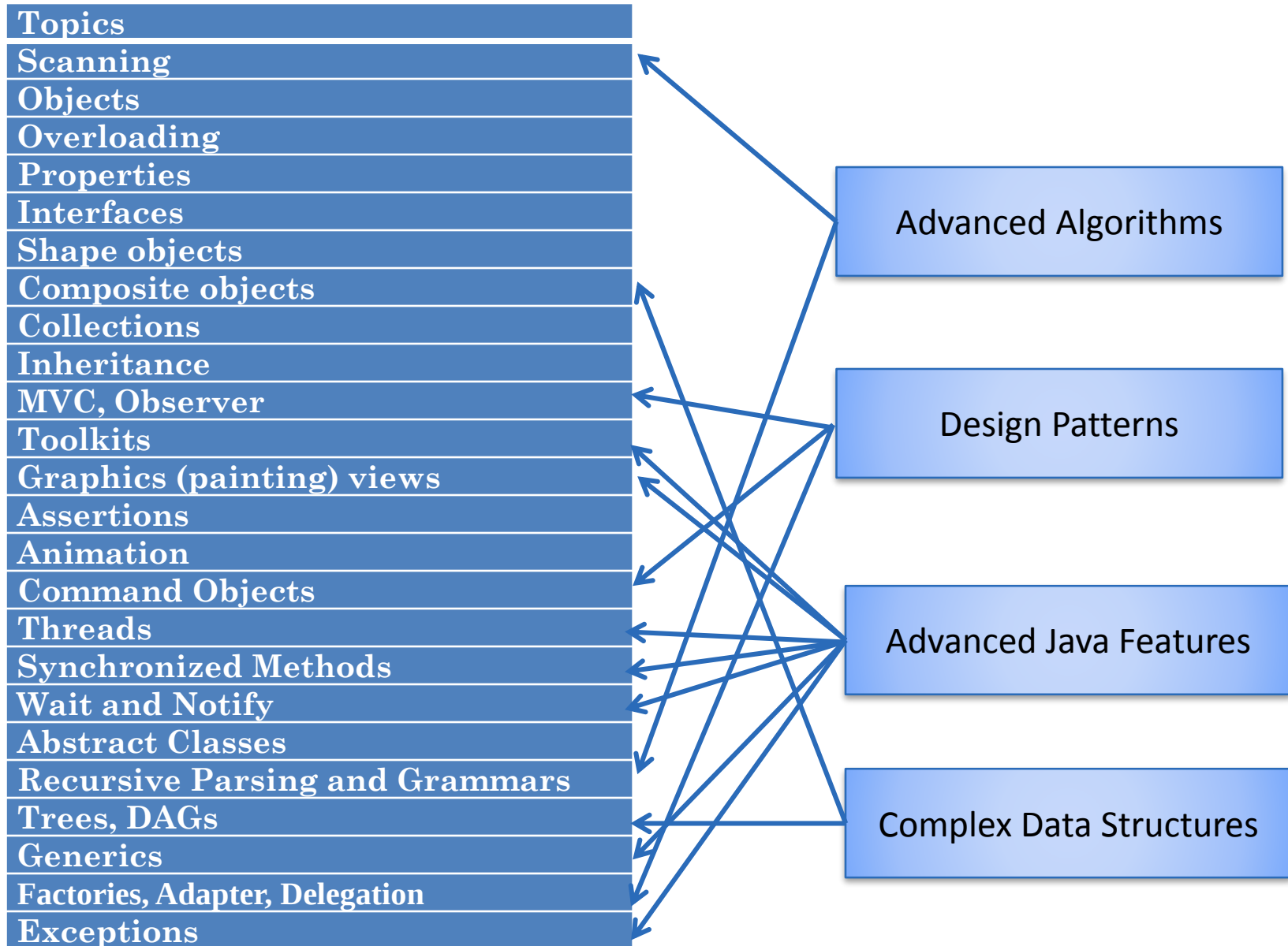
Internships

Research

Building an app



TOPICS



QUICK REVIEW OF DESIGN PATTERNS

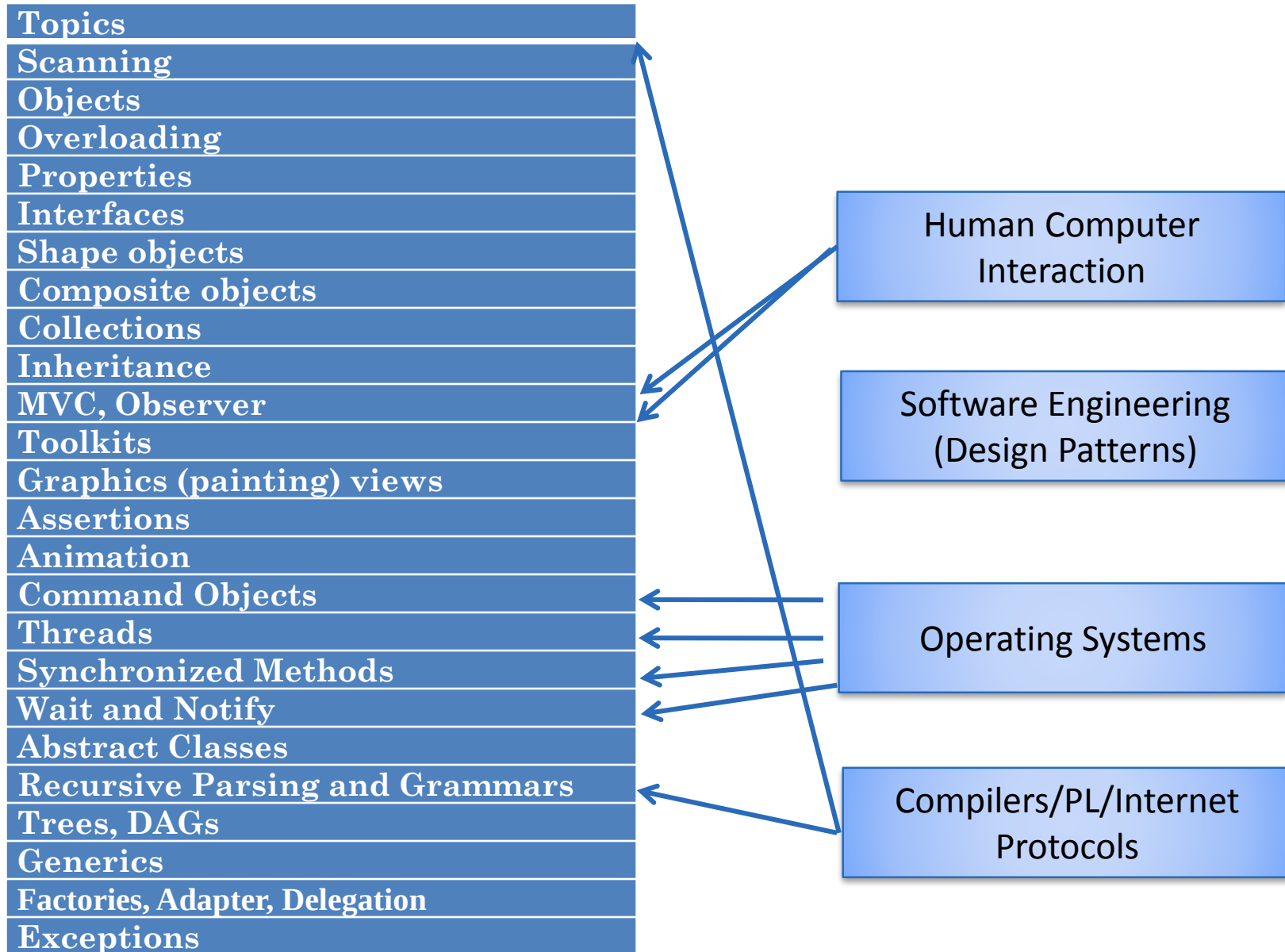
The design
patterns taught
in Comp 401

[Prasun Dewan, Teaching Inter-Object Design Patterns to Freshmen.
Proceedings of ACM SIGCSE. 2005](#)

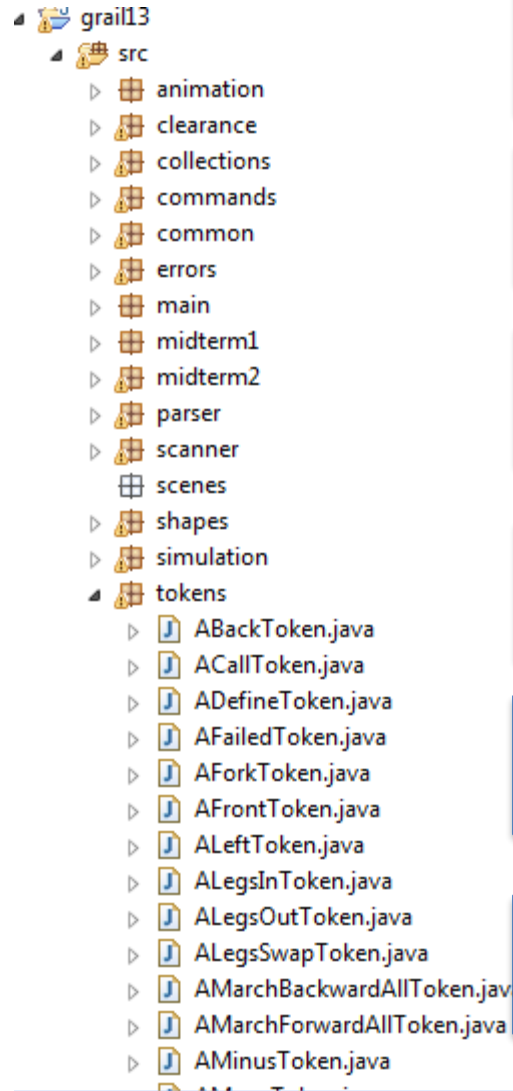
[PPT](#)



FUTURE OPTIONAL COURSES



LARGE-SCALE OBJECT-ORIENTED PROGRAMMING!



Large in terms of number of number
of classes/types

Defined by you (Programmer-
defined)

Count them!

Army of “paint listeners”

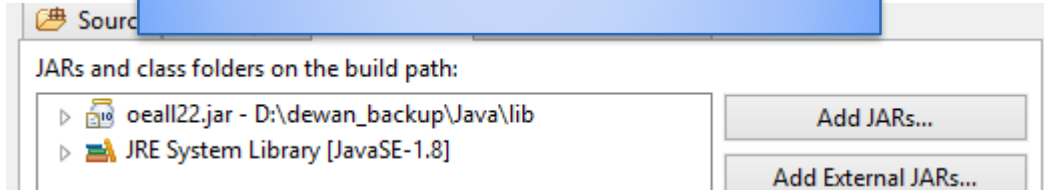
Largest (in terms of components)
you may write at UNC or elsewhere

Refactoring



PROGRAMMING PROCESS AND TOOLS

Class path and libraries (Eclipse)



Following Requirements

Comp 401 - Assignment 12: Wait Notify, Generics, Etc. (Last One!)

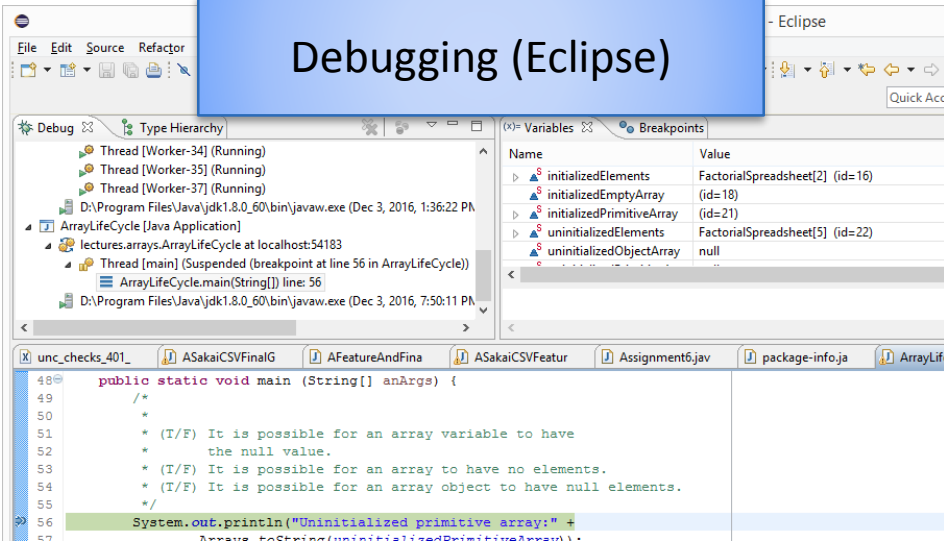
Date Assigned: Tue Nov 8, 2016

Completion Date: Wed Dec 7, 2016 (No late submission!)

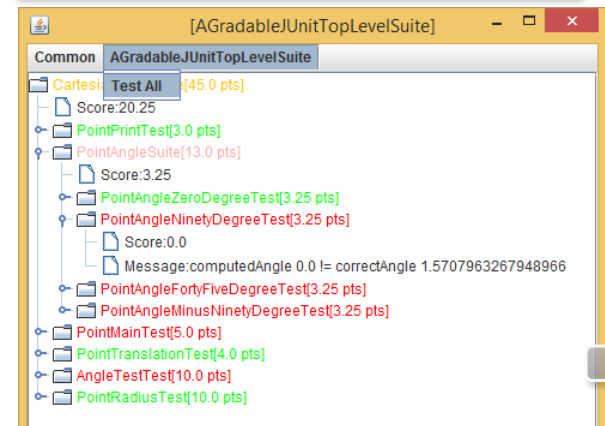
Early Completion Date: Mon Dec 5, 2016 (Monday)

assignment will change as we cover new material. The required part addresses wait notify and generics. There is extra credit for exceptions, recursive descent, undo and tries.

Debugging (Eclipse)



Using a Test Suite (JUnit)



STYLE CHECKS (CHECK STYLE) → SECURITY

```
package main;
```

Multiple markers at this line

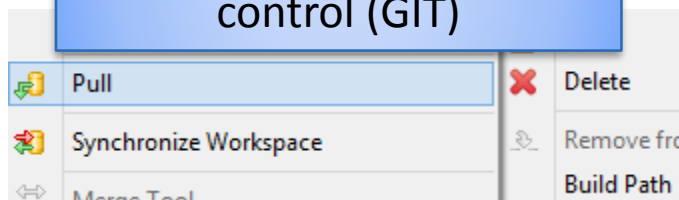
- expectedDeclaredSignatures: (Assignment1.java:3) In type Assignment1, missing declared signature: processInput:->void
- typeDefined: (Assignment1.java:3) Class/Interface Assignment1 matching tag main.Assignment(.*?) defined



PRAXIS: ACTIVE LEARNING, MAINTENANCE, VERSION CONTROL

```
public class AConsoleReadingUpperCasePrinter {  
  
    /**  
     * MAIN METHOD HEADER  
     * Syntax of main method shown below.  
     * Methods correspond to procedures and functions in other languages.  
     * Method names should be camel case starting with lowercase letter.  
     * Everything before the first curly brace is the method header.  
     */  
    public static void main(String[] args)  
    {  
        /**  
         * What happens if you use the following header instead, can you execute the program?  
         * Comment out the header above and uncomment the following to see what happens?  
         * What is the difference between the two headers?  
         */  
        // public static void main(String args) {  
        /**  
         * METHOD BODY  
         * The code between the outermost curly braces is the method body.  
         */  
    }  
}
```

Sharing code and version
control (GIT)



INTERPRETER OF PROGRAMMING LANGUAGE

The screenshot displays a software interface for an interpreter of a programming language, featuring a simulation environment and several configuration windows.

Simulation Environment (AnInterpretedHGSimulationDriver): The main window shows a blue background with four stick figures. Each figure has a head image: a man, a woman, a man, and a woman. The figures are arranged in a 2x2 grid. A circle is drawn around the bottom-right figure. A line connects this circle to the 'ASimulationInterpreterWithCannedProgram' window.

Configuration Windows:

- [ASimulationInterpreterWithCannedProgram]:** Contains a text field labeled 'thread beats'.
- [ACollectionBasedParser]:** Contains a text field labeled '1: thread beats'.
- [ATable]:** Contains several configuration fields:
 - guardArmsIn:** guardArmsIn
 - guardArmsIn:** 1: rotateleftarm -3, 2: rotaterightarm 3
 - guardArmsOut:** guardArmsOut
 - guardArmsOut:** 1: rotateleftarm 3, 2: rotaterightarm -3
 - beat:** beat
 - beat:** 1: call guardArmsIn, 2: proceedall, 3: sleep 500
 - beats:** beats
 - beats:** repeat 5 call beat
- [ACollectionBasedScanner]:** Contains a text field labeled 'Scanned String: thread beats' and a list of tokens:
 - 1 thread
 - 2 beats

Bottom Left Panel: A small window labeled 'Comm AClearanceMa' with buttons 'Proceed' and 'Proceed All'. Below it, a desktop icon labeled 'Kindle' is visible.

Bottom Center Panel: A text field labeled 't h r e a d b e a t s' with a cursor pointing to the end of the string.

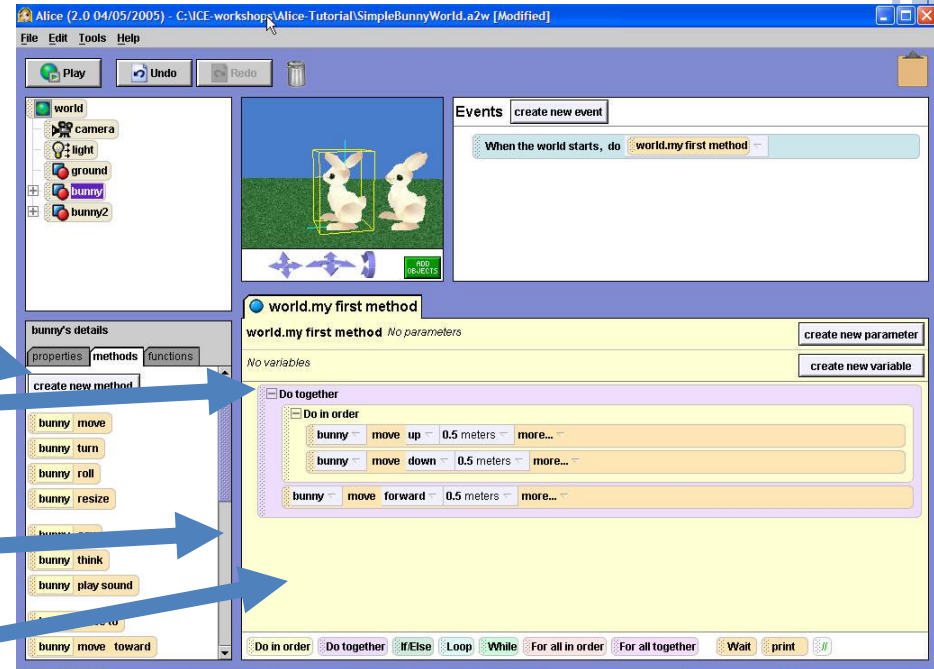


USE METHODS

- Drag method names from the details window Methods area
 - to world.my first method

- Can Group Methods

- Do in order
 - One after the other
- Or Do together
 - At the same time



You built an Alice-like visual programming environment that can be used by others to program

Beginning programmers learn how to use Alice



BUILDING AN APP

Handling of User and Internal Errors

Concurrency

Flexible, Multiple User Interface

Scanning

Parsing

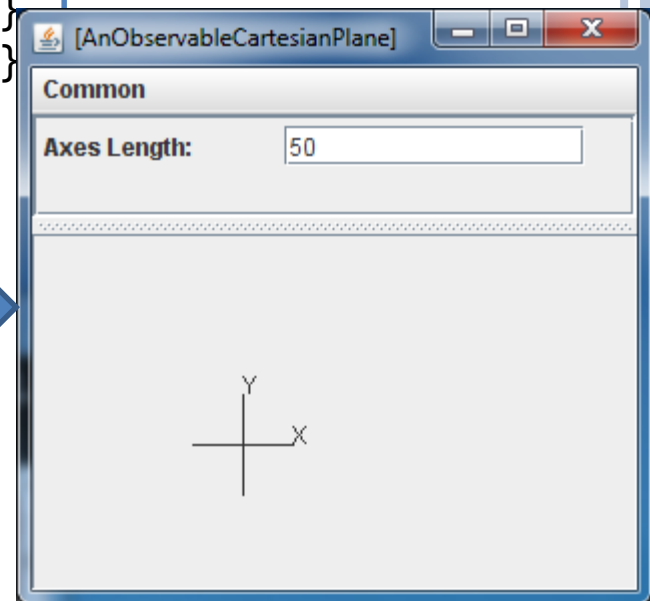
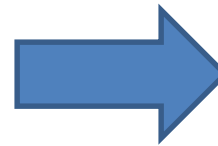
Geometry Processing

Use of Window Systems and Toolkits



LOGICAL DATA STRUCTURE VISUALIZER, STYLE-BASED TOOL, TRAINING WHEELS (OBJECTEDITOR)

```
public Line getXAxis() {return xAxis;}
public Line getYAxis() {return yAxis;}
public StringShape getXLabel() {return xLabel;}
public StringShape getYLabel() {return yLabel;}
public int getAxesLength() {return axesLength;}
public void setAxesLength(int anAxesLength) {
    axesLength = anAxesLength;
    xAxis.setWidth(axesLength);
    yAxis.setHeight(axesLength);
    xAxis.setX(toXAxisX());
    xAxis.setY(toXAxisY());
    yAxis.setX(toYAxisX());
    ...
}
```



DIARIES AND Q/A

Written and oral skills

Abstraction



QUIZZES AND EXAMS

Make you (and me) think about and identify what you have programmed



MODULARITY

Class Decomposition into Methods: Recursive descent programming

Program Decomposition into Classes: Design Patterns

Process Decomposition into Threads: wait, notify, synchronized methods, command objects



WHAT DID YOU NOT LEARN

Distributed Systems

Efficient and Complex Data Structures and Algorithms
(Implemented by Java)

How a computer system works on regular and mobile
computing (Architecture, OS, Compilers)

Proving things about what your program can (not) do

Team Programming

