Comp 401-F16
Course Overview

Instructor: Prasun Dewan (FB 150, help401-002-f16@cs.unc.edu)

Getting Started

Course page:
http://www.cs.unc.edu/~dewan/comp401/current/

Contents

Google (dewan comp401) to find page.
Also linked from my home page (google "dewan")

Use index and local web search to find parts

Piazza

Piazza link available from course page (Search piazza)

Sign up on Piazza asap, as all announcements will be made there

Assignments and Associated Resources

First assignment is due in a week!

Search (assignments) for table column with assignments or go to link "Resources by Topics"

Downloads

Resources by Topics

Resources by Topics

Unit	PPT Slides and Exported PDF and Videos	Chapters	Algorithm Videos	Assignments (Please scroll to see the first assignment)	Source
Course Information (8/18, 8/20)	PowerPoint-1 PDF-1 YouTube-1 Mix-1 PowerPoint-2		Bridge Scene - 1st day (long) Bridge Scene - 2nd day (short)	-	
Scanning	PowerPoint PDF YouTube Mix	Docx PDF Drive	Scanning Visualization	Number Scanner Checks File	lectures.scanning Package

Before Next Class

JDK Download	PowerPoint PDF			-	-
Eclipse Install & Use	PowerPoint PDF Installing JDK on Mac	PDF		-	-
Checkstyle with UNC Checks Install	PowerPoint PDF			-	-
ObjectEditor Install	PowerPoint PDF	PDF		-	-
Importing Git Project (JavaTeaching)	PowerPoint PDF			-	-
Specifying Main Args in Eclipse	PowerPoint PDF	PDF		-	-
Debugging in Eclipse	PowerPoint PDF	PDF		-	-
Relevant Java Syntax	PowerPoint PDF	PDF		-	lectures.java_syntax_overview Package
Scanning	PowerPoint PDF YouTube Mix	Docx PDF Drive	Scanning Visualization	Number Scanner Checks File	lectures.scanning Package

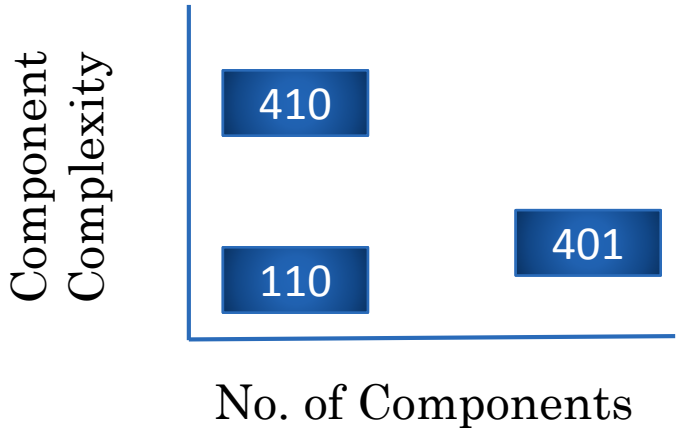
Background?

- Java vs. Non-Java?
  - Course is not about Java
  - Expected to use only those Java features taught in class.
- Object-Oriented vs. Conventional Programming
  - Assume background in conventional programming: Types, variables, assignment , constants, expression, conditionals and loops, input and output, arrays and/or strings, procedures/methods.
  - Weeding out first assignment.
  - Will teach all aspects of object-oriented programming.
  - Repetition for those who know object-oriented programming.

Course Content

110	Introductory Programming	**Small** number of **simple** code fragments connected by you
401	Intermediate Programming	**Large** number (~40) of **simple** code fragments connected by you
410	Data Structures	**Small** number of **complex** code fragments connected by you

Only the optional compiler course will involve more components!



Layered Assignments = Project

Course Information (8/18)			Bridge Scene - 1st day (long) Bridge Scene - 2nd day (short)		

Assignment 11

Assignment 2

Assignment 1

Weekly assignments will build on each other to create a semester project

Each assignment with its own due date and points

Extra Credit

Extra Credit

Allow (a) a number to be succeeded or preceded by a variable number of blanks as in " 2 456 25 3000 ", (b) an arbitrary number of numbers in a line. Do not terminate the program after encountering the first illegal (unexpected) character. Print the illegal character and continue scanning assuming the character had not been input.

Students have varying interests and abilities
Make up or insurance against bad grade in other assignments or exams
Better to give early without extra credit than late with
But if you are already late, might as well do extra credit to make up for late points

Constraints

Constraints

1. Java has libraries that make the writing of this program trivial. The only library functions you should use are the Character.isDigit(), substring() and the Integer.parseInt() functions. Character.isDigit() is like Character.isUppercase() except that it tells us whether a character is a digit rather than whether it is an uppercase letter. substring(), applicable to any string, is explained in the class material. Intreger.parseInt()takes a string consisting of digits and converts into an

Forbid use of certain Java libraries

Goal is not to teach Java and its libraries

It is to teach you how to build these libraries

Usually Java features not covered in class will be banned

Correctness is only one of the goals

Require use of certain programming techniques

Program must also be efficient and well crafted

Early Reward

Assignment 1:
Writing a Number Scanner

Date Assigned: Tue Aug 23, 2016

Completion Date: Fri Sep 2, 2016 (11:55 pm)

Early Submission Date: Wed Aug 31, 2016 (11:55 pm)

5% Extra credit if submitted early on a Wednesday

Normal submission date is a Friday

If you shoot for Wednesday you should be ready by Friday with TA help

End of day is 11:55pm for regular and late deadlines

Late Penalty

5% late if submitted next Monday, 10% late if submitted next Friday

20% late if submitted any day after that, **but no manual grading points**

0% for manual grading component – fairly high in later assignments

No correction of auto grading results

Why Small Late Penalty?

Big difference between getting code working and almost working

Big differences in grades also

Very little partial credit if program not working

Errors will accumulate because of layered assignments

Better late than not working

But being more than 1 week late for multiple assignments is recipe for failing

Coding vs Design/Debugging

The TAS and I are here to help you debug and design (but not write code)

Assignments may contain solution in English (read only if stuck)

Implementation Hints

Read this only if you have trouble developing your own solution. The small print is encouraging you to first think of the problem on your own.

As in the class example, you should define a variable that keeps trick of the index of the start of each token. In the class example, the size of the token was constant (1) and there were no spaced in between tokens. This means that the startIndex of a token was always one more than the startIndex of the previous token. Now the tokens are of variable size. This means that in changing the startIndex, you must take into account the end of the variable-length token and spaces) in between tokens. Given a start index, the end of the token can be computed using the indexOf(() function As in the class example, make sure the startIndex does not go beyond the length of the string.

Can help each other with design and debugging as long as it does not lead to code sharing

Honor Code

Sharing of code is honor code violation
More or less same project as last time, but do not look at past solutions
Automatic grader may be extended with plagiarism detector

Why We Lie : TED Radio Hour : NPR

Out of Office Help

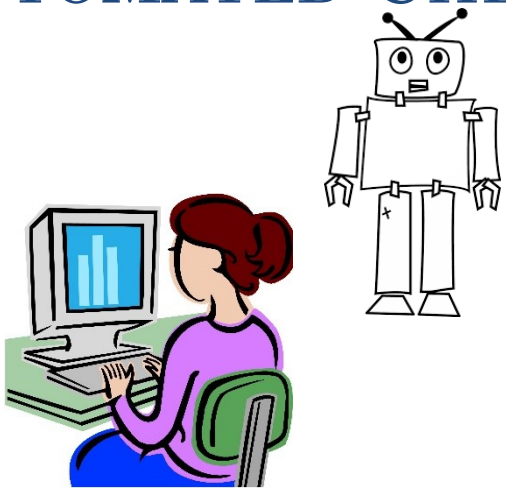
Do

- Use Piazza for out of class questions
- Use help401-002-f16@cs.unc.edu for grades and other private queries

Don't

- Send mail to individual instructors

Automated Checking

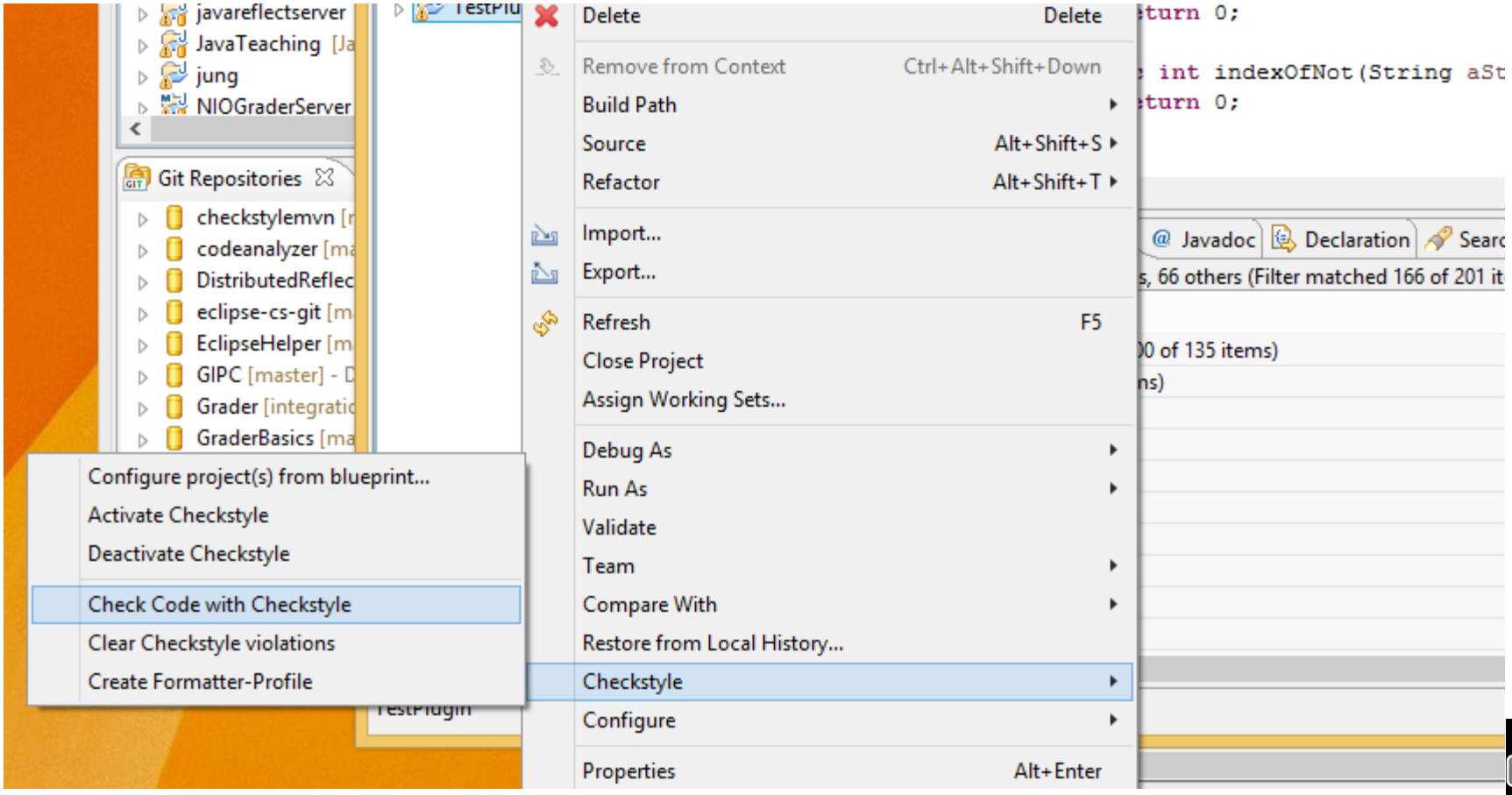


Warn against requirements not met

Indicate potential sources of error

Automatic Checking: Code Analyzer

Code analyzer runnable from Eclipse



Code Analyzer: In-Place Messages

```
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

Code Analyzer: Problems Pane

Problems | Javadoc | Declaration | Search | Console | Send Assignment

0 errors, 135 warnings, 66 others (Filter matched 166 of 201 items)

Description
illegalMethodCall: (IllegalMethodCalls.java:123) called disallowed method bar-->String.split
illegalMethodCall: (IllegalMethodCalls.java:126) called disallowed method bar-->String.split

Assignment Specification

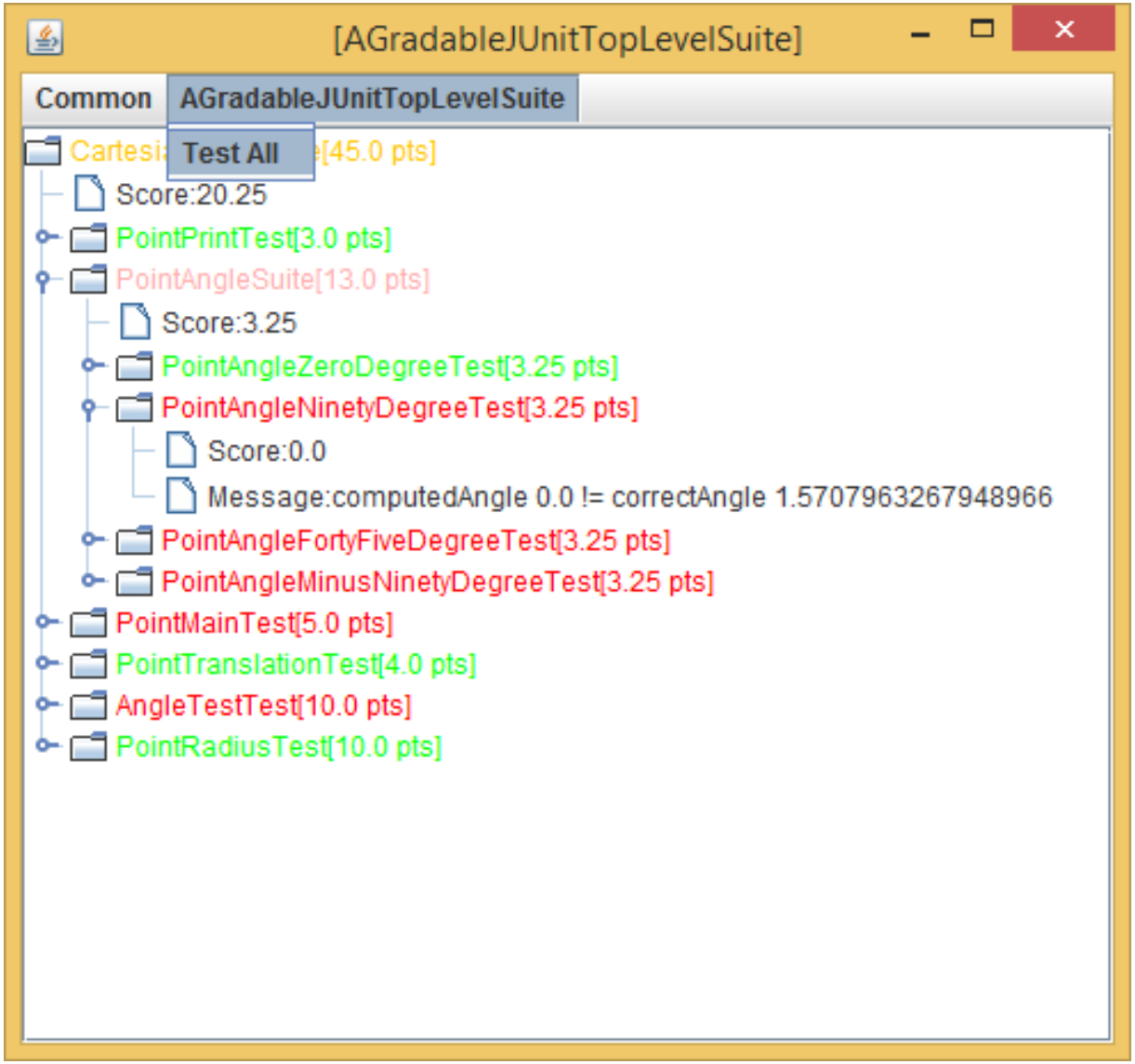
Number Scanner

Checks File

Checks File

```
<property name="expectedTypes" value=" main.Assignment(.*), mp.scanner.Scan
/module>
odule>
dule name="unc.tools.checkstyle.ANonCachingTreeWalker">
property name="severity" value="warning"/>
nodule name="ExpectedDeclaredSignatures">
<property name="severity" value="warning"/>
<property name="expectedSignatures" value=" main.Assignment1 = processInput
indexOfNot:String;char;int->int//EC, grail.scanner.ScanningIterator= indexO
/module>
nodule name="MissingMethodCall">
<property name="severity" value="warning"/>
<property name="expectedCalls" value=" main.Assignment1 = processInput:->vo
indexOfNot:String;char;int->void//EC AND (.*)!hasNext:->boolean//EC AND (.*
>void AND indexOfNot:String;char;int->void//EC "/>
/module>
```

Automatic Checking: Local Code Executor



Automatic Checking: Server Code Analyzer and Code Executor

```
//OEFrame editor2 = ObjectEditor.edit(interpreterView);
OEFrame editor = ObjectEditor.edit(bridgeScene);
bridgeScene.setOEFrame(editor);

editor.hideMainPanel();
editor.setSize(800, 500);
pm.stepComplete();

sleep(2000);
```

Send Assignment

COMP 401-038

Assignment

Preconditions, Commands, Threads, Animation (Assignment 10)

Onyen vitkus

Password ••••••••••••••••••

Automatic Checking: Server Code Analyzer and Code Executor

Assignment10.java | Preconditions, Commands, Threads, Animation (Assignment 10) Grading

Grading response for Preconditions, Commands, Threads, Animation (Assignment 10)

LOOKUP STATISTIC UPLOAD

Grading:

Requirement	% Autograded	Points	Possible
Precondition methods	100.0	6	12
Console view shows precond events	0.0	0	12
Say & move cmd objects	100.0	5	5
Move cmd constructor	100.0	3	3
Say cmd constructor	100.0	3	3
Say and move parsers	100.0	5	5
Command object invoked	100.0	0	5
Animating methods	100.0	5	5
Methods start new threads	100.0	0	10
Animating command classes	0.0	0	20
Animator classes	0.0	0	20
Guard animation	100.0	2	5
Precondition buttons	0.0	0	10
Awesome demo	0.0	0	5

Notes:

- Command object invoked
  - Command objects returned from say and move are invoked.
    - Couldn't find a parser invoker that called run()

Send Assignment

COMP 401-038

Assignment

Preconditions, Commands, Threads, Animation (Assignment 10)

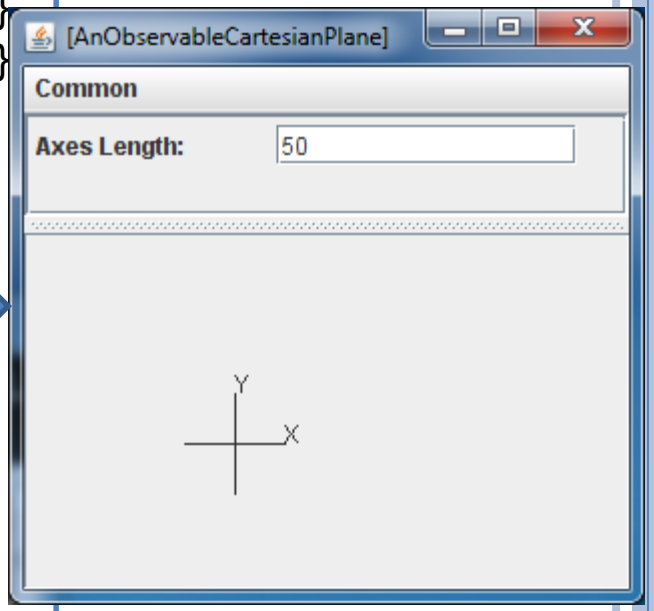
Onyen vitkus

Password ••••••••••••••••

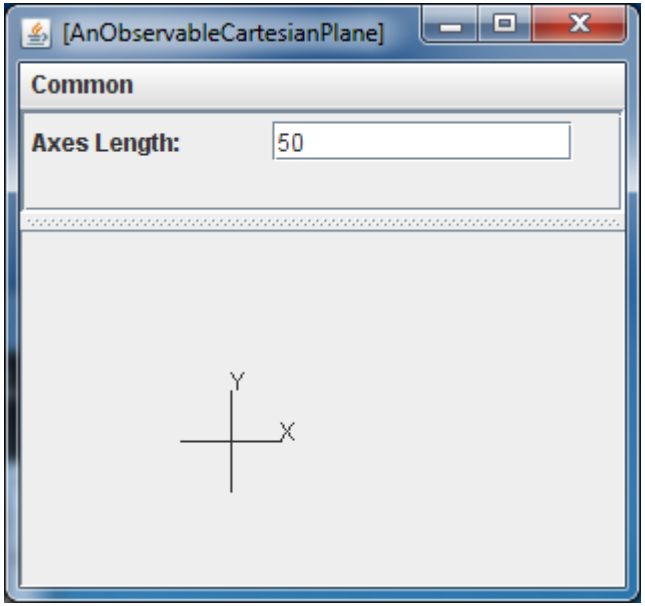
Send Assignment

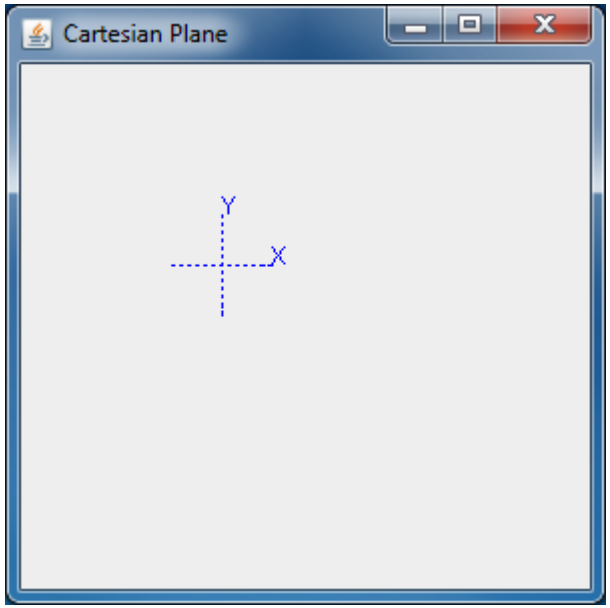
ObjectEditor: Code Analyzer, Executor, Visualizng Checker

```
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());
  …
 }
```

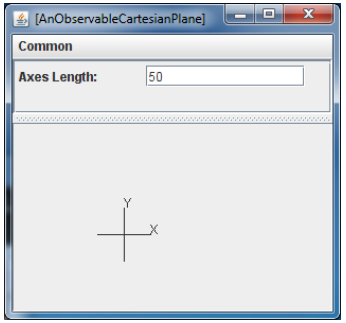


ObjectEditor: Training Wheels





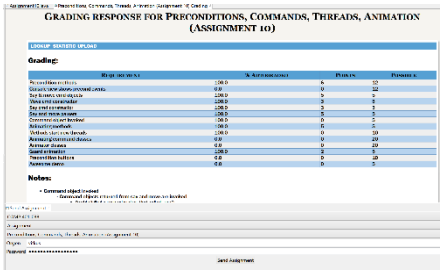
Research vs Required Tools

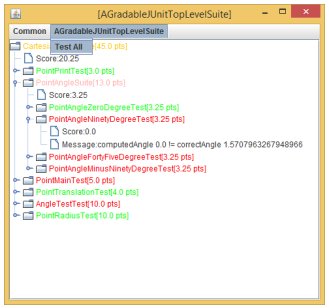


✓

Only ObjectEditor is required

Others are optional, research tools with consent form






```
package main;
```

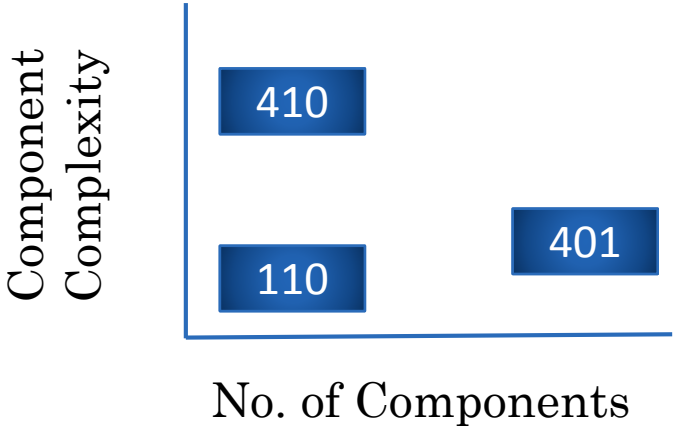
Multiple markers at this line
- expectedDeclaredSignatures: (Assignment1.java:3) In type Assignment1, missing declared signature: processInput:->void
- typeDefined: (Assignment1.java:3) Class/Interface Assignment1 matching tag main.Assignment(.*) defined

Downloads and Consent Form

Downloads

ObjectEditor Version 3 (used in comp110)	oeall3
ObjectEditor Version 19 (used last year)	oeall19
ObjectEditor Version 21	oeall21
ObjectEditor Version 22 (latest, use this unless it fails on you)	oeall22
Checkstyle	UNCChecks_6.5.0.jar Checkstyle 6.5 zip
GraderBasics	GraderBasics
Consent Form	ConsentForm
Images	images.zip

Unusual Course



Only the optional compiler course will involve more components!

Some creativity in number and nature of components

Unusual course – no textbook at this level covers such large programs

Learning Resources

No textbook!

Alternatives?

PDF of Slides

www.cs.unc.edu/~dewan/comp401/f15/Lectures/Scanning.pdf

Java Program Structure

```
D:\dewan_backup\Java\JavaTeaching\bin>java lectures.scanning.AnUpperCasePrinter
"John F. Kennedy"
Upper Case Letters:
JFK
```

```
package lectures.scanning;
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of argum
+ ". Terminating program.");
      System.exit(-1);
    }
    String scannedString = args[0];
    System.out.println("Upper Case Letters:");
    int index = 0;
    while (index < scannedString.length()) {
      char nextLetter = scannedString.charAt(index);
      if (Character.isUpperCase(nextLetter))
        System.out.print(nextLetter);
      index++;
    }
    System.out.println();
  }
}
```

Must have this procedure header in executable program

Predefined internal library operations

Print on new vs. previous line

Course.pptx

Show all dow

PowerPoint of Slides

Java Program Structure

```
D:\dewan_backup\Java\JavaTeaching\bin>java lectures.scanning.AnUpperCasePrinter
"John F. Kennedy"
Upper Case Letters:
JFK
```

```java
package lectures.scanning;
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of argum
+ ". Terminating program.");
      System.exit(-1);
    }
    String scannedString = args[0];
    System.out.println("Upper Case Letters:");
    int index = 0;
    while (index < scannedString.length()) {
      char nextLetter = scannedString.charAt(index);
      if (Character.isUpperCase(nextLetter))
        System.out.print(nextLetter);
      index++;
    }
    System.out.println();
  }
}
```

Must have this procedure header in executable program

Predefined internal library operations

Print on new vs. previous line

Click to add notes

SLIDE 17 OF 36 NOTES COMMENTS

Slide Show → Synchronized Recording and Animations

PowerPoint Slide Show - [Scanning.pptx] - PowerPoint

Java Program Structure

```
D:\dewan_backup\Java\JavaTeaching\bin>java lectures.scanning.AnUpperCasePrinter
"John F. Kennedy"
Upper Case Letters:
JFK
```

```
package lectures.scanning;
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of arguments:" + args.length
+ ". Terminating program.");
      System.exit(-1);
    }
    String scannedString = args[0];
    System.out.println("Upper Case Letters:");
    int index = 0;
    while (index < scannedString.length()) {
      char nextLetter = scannedString.charAt(index);
      if (Character.isUpperCase(nextLetter))
        System.out.print(nextLetter);
```

Can escape out into unsynchronized or no audio mode (WPS Office on Android will play synchronized audio)

17

SLIDE 17 OF 36

PowerPoint Slides With Unsynchronized Recordings and Media Control

Java Program Structure

```
D:\devan_backup\Java\JavaTeaching\bin>java lectures.scanning.AnUpperCasePrinter
"John F. Kennedy"
Upper Case Letters:
JFK
```

```
package lectures.scanning;
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of argume
+ ". Terminating program.");
      System.exit(-1);
    }
    String scannedString = args[0];
    System.out.println("Upper Case Letters:");
    int index = 0;
    while (index < scannedString.length()) {
      char nextLetter = scannedString.charAt(index);
      if (Character.isUpperCase(nextLetter))
        System.out.print(nextLetter);
      index++;
    }
    System.out.println();
  }
}
```

Must have this procedure header in executable program

Predefined internal library operations

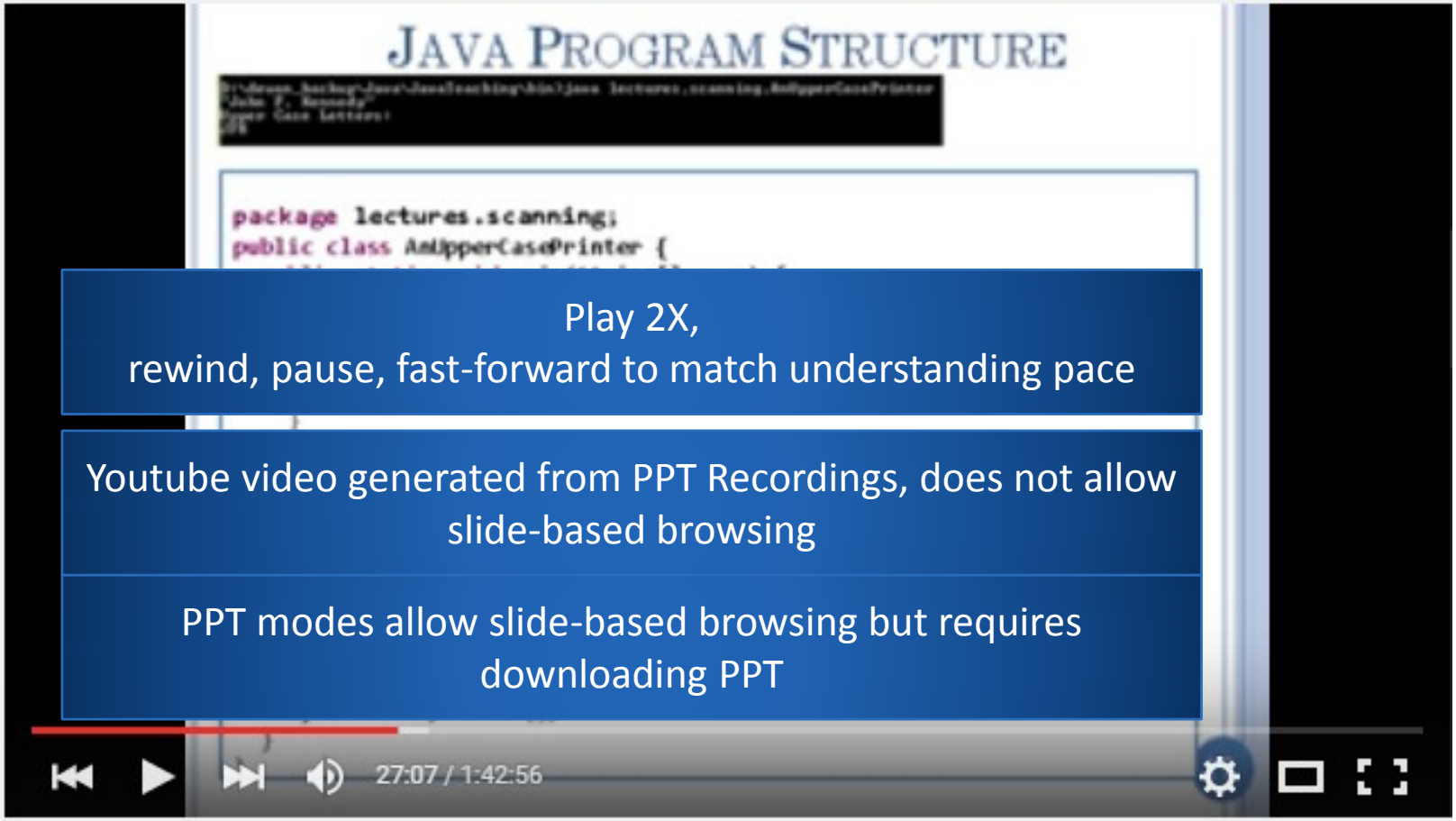
Print on new vs. previous li

Click to add notes

SLIDE 17 OF 36

NOTES COMMENTS

Recorded YouTube Videos



Play 2X,
rewind, pause, fast-forward to match understanding pace

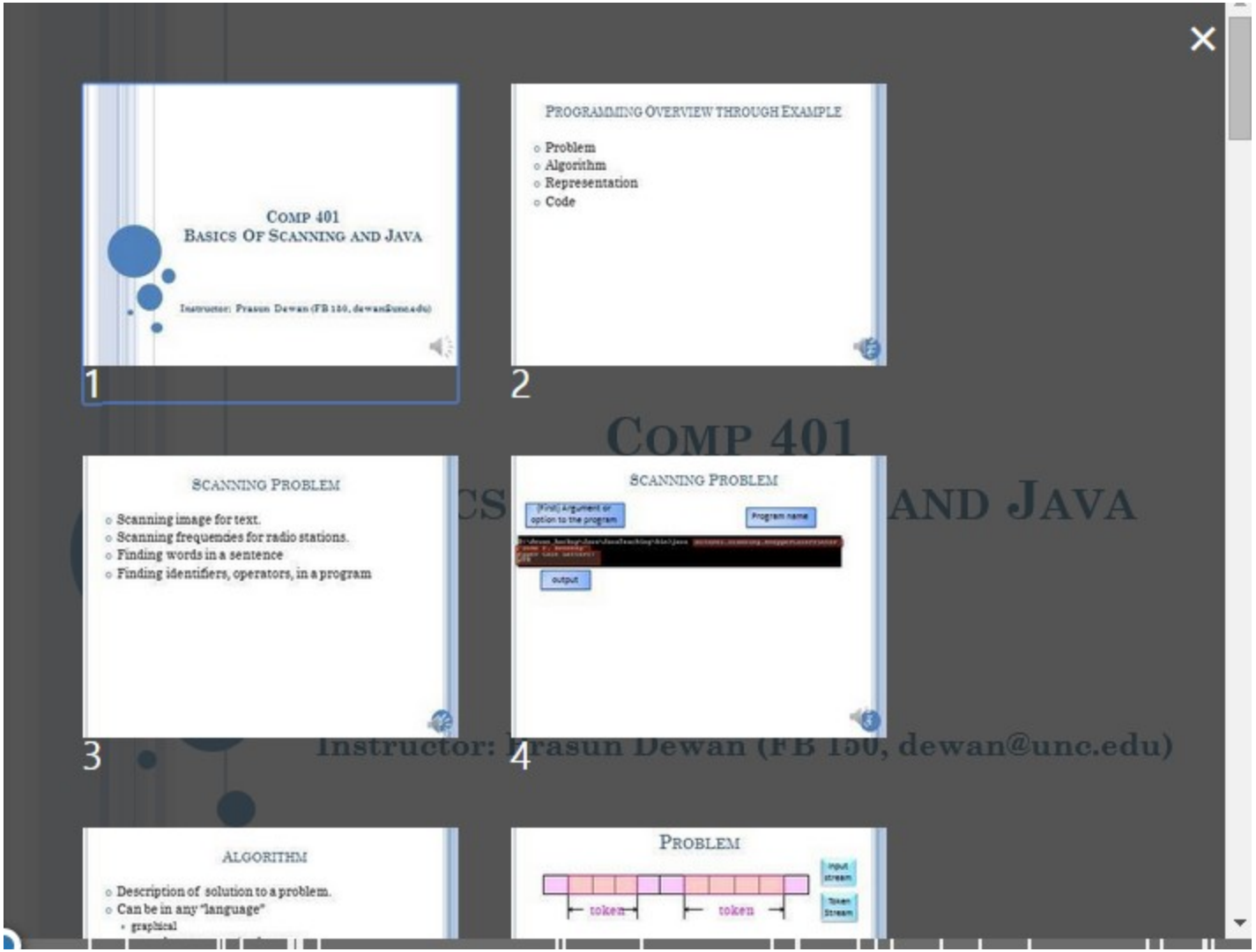
Youtube video generated from PPT Recordings, does not allow slide-based browsing

PPT modes allow slide-based browsing but requires downloading PPT

Office Mix

Slide markers

Slide-Based Browsing



Video Nature

Long pauses, you may know the answer

Cannot hear student answer

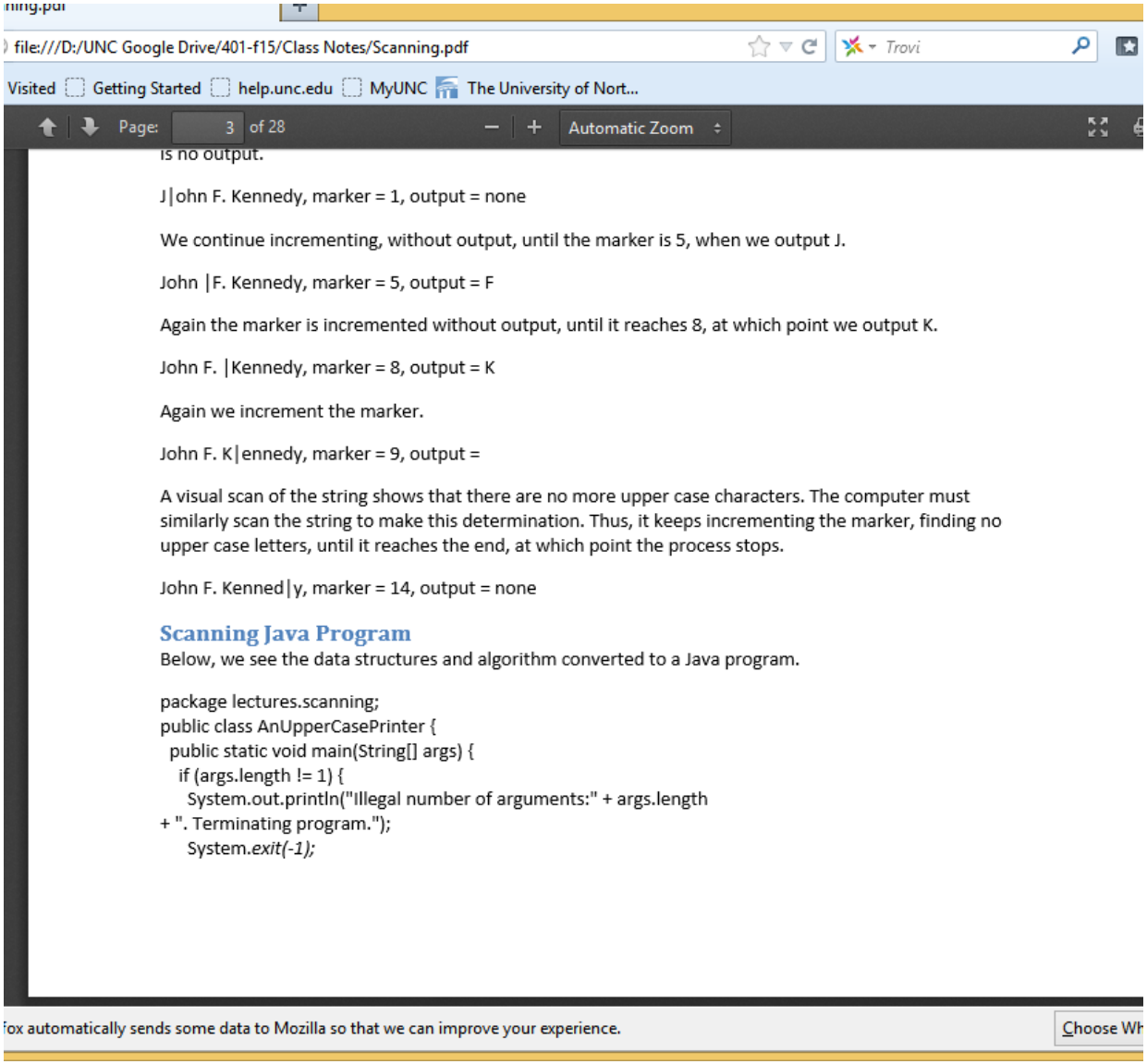
Audio is not the fastest way to get information, specially when studying for an exam

Recordings of live lectures with q/a rather than 15 minute lessons

Can fast forward

You can get a clue from my answer

PDF of Word Documents



file:///D:/UNC Google Drive/401-f15/Class Notes/Scanning.pdf

Page: 3 of 28 Automatic Zoom

is no output.

J|ohn F. Kennedy, marker = 1, output = none

We continue incrementing, without output, until the marker is 5, when we output J.

John |F. Kennedy, marker = 5, output = F

Again the marker is incremented without output, until it reaches 8, at which point we output K.

John F. |Kennedy, marker = 8, output = K

Again we increment the marker.

John F. K|ennedy, marker = 9, output =

A visual scan of the string shows that there are no more upper case characters. The computer must similarly scan the string to make this determination. Thus, it keeps incrementing the marker, finding no upper case letters, until it reaches the end, at which point the process stops.

John F. Kenned|y, marker = 14, output = none

Scanning Java Program

Below, we see the data structures and algorithm converted to a Java program.

```
package lectures.scanning;
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of arguments:" + args.length
+ ". Terminating program.");
      System.exit(-1);
```

Fox automatically sends some data to Mozilla so that we can improve your experience.

Little graphics, designed for mobile reading on mobile computers

Lots of (obvious) mistakes

Publicly Commentable Google Docs

dewan@cs.unc.edu

t Format Tools Table Add-ons Help All changes saved in Drive

Comments Share

Normal text Calibri 11 More Editing

J|ohn F. Kennedy, marker = 1, output = none

We continue incrementing, without output, until the marker is 5, when we output J.

John |F. Kennedy, marker = 5, output = F

Again the marker is incremented without output, until it reaches 8, at which point we output K.

John F. |Kennedy, marker = 8, output = K

Again we increment the marker.

John F. K|ennedy, marker = 9, output =

A visual scan of the string shows that there are no more upper case characters. The computer must similarly scan the string to make this determination. Thus, it keeps incrementing the marker, finding no upper case letters, until it reaches the end, at which point the process stops.

John F. Kenned|y, marker = 14, output = none

Scanning Java Program

Below, we see the data structures and algorithm converted to a Java program.

```
package lectures.scanning;
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of arguments:" + args.length
+ ". Terminating program.");
      System.exit(-1);
```

Prasun Dewan

Comment Cancel

Web-Based Browsable Code

Source Code of Class Examples

Zipped Directory
JavaToHTML

Java Source
All Classes
Packages
lectures.animation.loops
lectures.animation.mvc
lectures.animation.threads.synchronized_method
lectures.animation.threads.ui
lectures.animation.threads.wait_notify
lectures.animation.threads_commands
lectures.arrays

AnOperatorPrecedenceDemoer
AnOverWeightSpreadsheet
AnUndoableBMISpreadsheet
AnUndoableCounter
AnUnmodularUndoableBMISpreadsheet
AnUpperCaseIndexBasedScanner
AnUpperCaseIndexBasedScanner
AnUpperCaseIterator
AnUpperCaseIterator
AnUpperCaseIterator.1
AnUpperCasePrinter
AnUpperCasePrinter

AnUpperCasePrinter.java

```
package lectures.scanning;
import util.annotations.WebDocuments;
@WebDocuments({"Lectures/Scanning.pptx", "Lectures/Scanning.pdf", "Vid
public class AnUpperCasePrinter {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Illegal number of arguments:" + args.l
                    + ". Terminating program.");
            System.exit(-1);
        }
        String scannedString = args[0];
        System.out.println("Upper Case Letters:");
        int index = 0;
        while (index < scannedString.length()) {
            char nextLetter = scannedString.charAt(index);
            if (nextLetter >= 'A' && nextLetter <= 'Z')
                System.out.print(nextLetter);
            index++;
        }
        System.out.println();
    }

}
```

May not be up-to date

Eclipse Java Project of Lecture Code on GIT

Package Explorer | Type Hierarchy

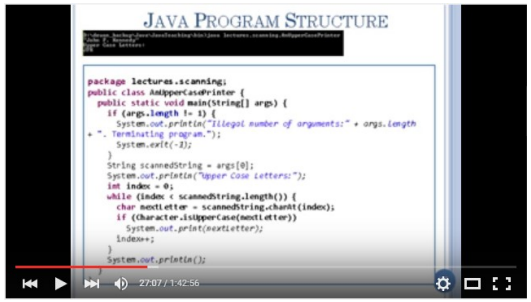
- lectures.mvc
- lectures.mvc.collections
- lectures.mvc.interactor
- lectures.mvc.monolithic
- lectures.mvc.properties
- lectures.mvc.toolkit
- lectures.nester
- lectures.objects
- lectures.oegraphics
- lectures.parsing_grammars
- lectures.recursion
- lectures.reflection_action_objects
- lectures.scanning
  - AConsoleReadingUpperCasePrinter.java
  - AModularConsoleReadingUpperCasePrint
  - AModularUpperCasePrinter.java
  - AnUpperCasePrinter.java
  - Console.java
  - package-info.java
  - StyleErrorsCS1.java
- lectures.state_properties
- lectures.static_state
- lectures.test.junit
- lectures.types.char_string

AJavaCheckStyle | ARequirementsTo | AConsoleReading | AModularUpperCa

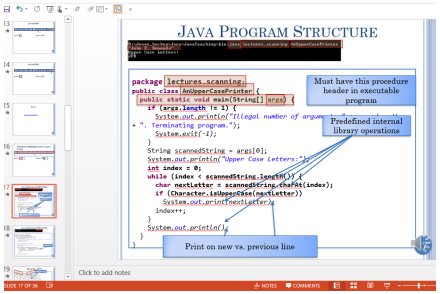
```java
package lectures.scanning;
public class AnUpperCasePrinter {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Illegal number of arguments:" + args.length
                    + ". Terminating program.");
            System.exit(-1);
        }
        String scannedString = args[0];
        System.out.println("Upper Case Letters:");
        int index = 0;
        while (index < scannedString.length()) {
            char nextLetter = scannedString.charAt(index);
            if (nextLetter >= 'A' && nextLetter <= 'Z')
                System.out.print(nextLetter);
            index++;
        }
        System.out.println();
    }

}
```

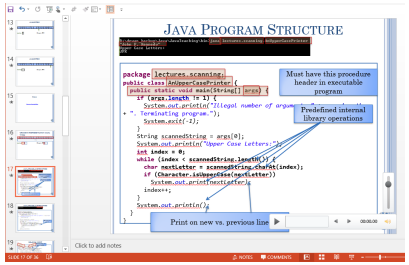
Design Space of Study Modes!



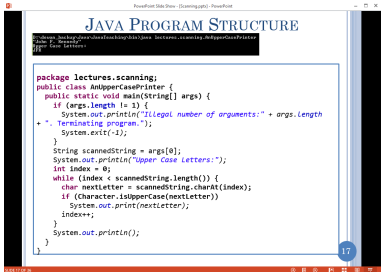
You Tube video



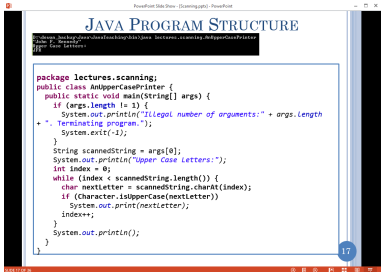
Slides without audio



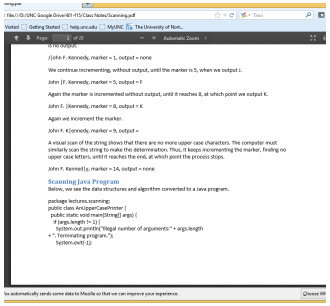
Slides with unsynchronized audio



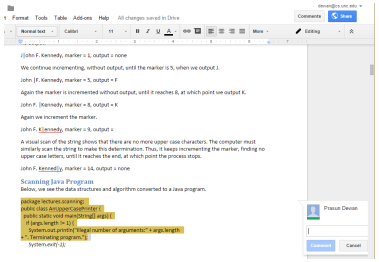
Slideshow with synchronized recording



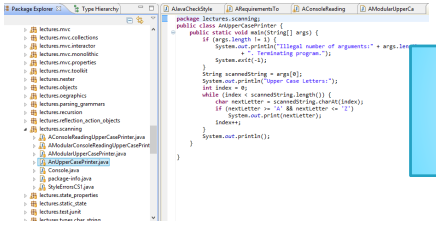
Office Mix



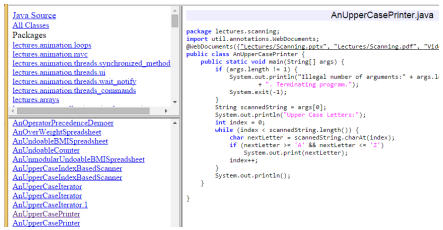
Word PDFs



Shared Google Docs



Eclipse Project



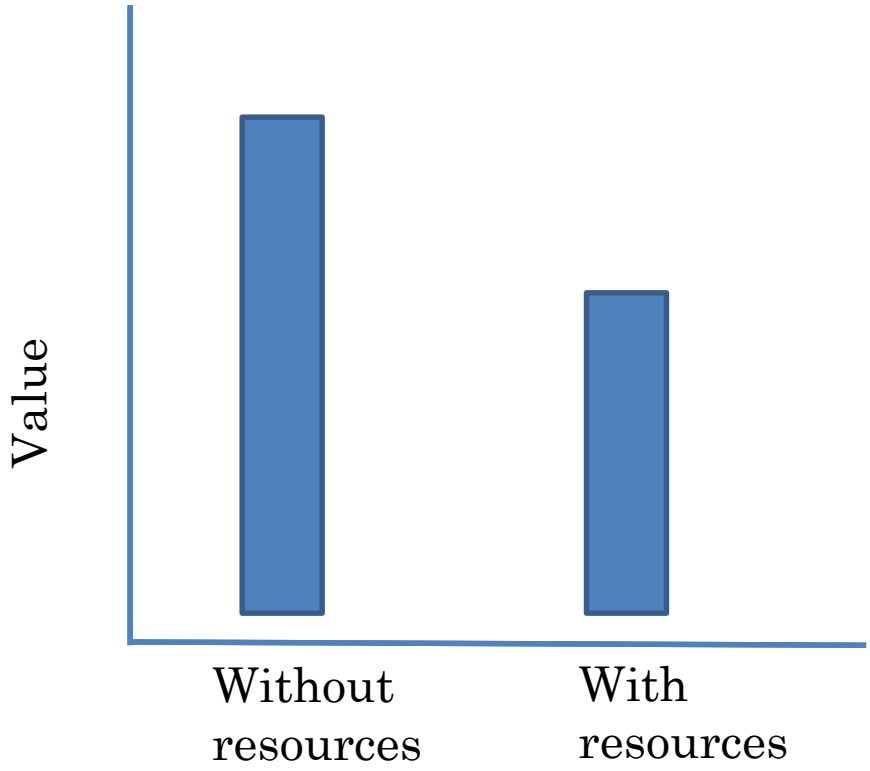
Browsable Proje

Web Site Links

Scanning	PowerPoint PDF YouTube Mix	Docx PDF Drive	Scanning Visualization	Number Scanner	lectures.scanning Package

What do we do in class?

Live Lecture?



Discussion is About Concrete Programs

PowerPoint Slide Show - [Scanning.pptx] - PowerPoint

Java Program Structure

```
D:\dewan_backup\Java\JavaTeaching\bin>java lectures.scanning.AnUpperCasePrinter
"John F. Kennedy"
Upper Case Letters:
JFK
```

```
package lectures.scanning;
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of arguments:" + args.length
+ ". Terminating program.");
      System.exit(-1);
    }
    String scannedString = args[0];
    System.out.println("Upper Case Letters:");
    int index = 0;
    while (index < scannedString.length()) {
      char nextLetter = scannedString.charAt(index);
      if (
        Sy
      inde
    }
    System.
  }
}
```

Pace at which you understand lecture in general and code in particular varies

17

SLIDE 17 OF 36

Pacing Yourself

A variety of sources with different amounts of information

Each source can be browsed at your own pace

What do we do in class?

Homework?

Deep thinking done solo?

Limited discussion with classmates?

What do we do in class?

Quizzes: Test that you understood support material?

One hour means deep testing, puts pressure

May not have discipline to access material

What do we do in class?

I code in class, you watch.

I am not a touch typist!

You learn more as a driver than passenger

Praxis

```
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.
```

Class Structure

Introduction to Praxis

Do praxis with as large a group as possible for 50 minutes and answer associated Sakai quiz

Answer class questions on the material

Finish praxis based on the answers and further understanding

If you ask for help from us, you are pledging that you have done the praxis

Attendance

Do	Don't
Come to class	Come to class late **or leave class early**

Attendance Sakai "Quiz"

Attendance

Table of Contents

Part 1 of 1 -

Question 1 of 3

Identify the dates on which you cannot attend class and give the reason.

Maximum number of characters (including HTML tags added by text editor): 32,000

Count Characters

Show/Hide Rich-Text Editor

Class Discussion "Quiz"

Part 1 of 1 -

Question 1 of 5

Explain how you contributed constructively to the class discussion, giving the date.

Maximum number of characters (including HTML tags added by text editor): 32,000

Count Characters

Show/Hide Rich-Text Editor

Grade Distribution (Current Plan)

Midterm	22%
Final	28%
In-Class Work (Recitations & Lecture Quizzes, Discussion, Attendance)	15%
Assignments	35%
Fudge Factor (Special participation, other distinguishing factors), particularly useful for borderline cases	5%