



COMP 401-LEARNING RESOURCES: FROM SLIDE PDFS TO PRAXES

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



TEXT BOOK

No textbook for this unusual course.

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 arguments. 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 down



POWERPOINT OF SLIDES

The slide displays the following Java code with annotations:

```
D:\deuan_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. 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();
    }
}
```

Annotations and callouts:

- Must have this procedure header in executable program**: Points to the `public static void main(String[] args) {` line.
- Predefined internal library operations**: Points to `System.out.println()` and `System.out.print()`.
- Print on new vs. previous line**: Points to the `System.out.println();` line at the end of the `while` loop.

Slide navigation sidebar (left):

- 13: [Thumbnail]
- 14: [Thumbnail]
- 15: [Thumbnail]
- 16: [Thumbnail]
- 17: [Thumbnail - Current Slide]
- 18: [Thumbnail]
- 19: [Thumbnail]

Bottom status bar: Click to add notes



SLIDE SHOW → SYNCHRONIZED RECORDING AND ANIMATIONS



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)



POWERPOINT SLIDES WITH UNSYNCHRONIZED RECORDINGS AND MEDIA CONTROL

The slide displays the following Java code with annotations:

```
D:\devan_hackin\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. 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();  
    }  
}
```

Annotations and callouts:

- Must have this procedure header in executable program**: Points to the `public static void main(String[] args) {` line.
- Predefined internal library operations**: Points to `System.out.println()` and `System.out.print()`.
- Print on new vs. previous line**: Points to the `System.out.println();` line at the end of the `main` method.

Slide navigation and controls:

- Slide thumbnails on the left: 13, 14, 15, 16, 17 (highlighted), 18, 19.
- Slide 17 thumbnail: **AUTH PROGRAM EXECUTER**
- Slide 18 thumbnail: **AUTH PROGRAM EXECUTER**
- Slide 19 thumbnail: **RELATION SHIPS**
- Bottom status bar: SLIDE 17 OF 36
- Bottom right: NOTES, COMMENTS, and media controls (play, volume, 00:00.00).



RECORDED YOUTUBE VIDEOS

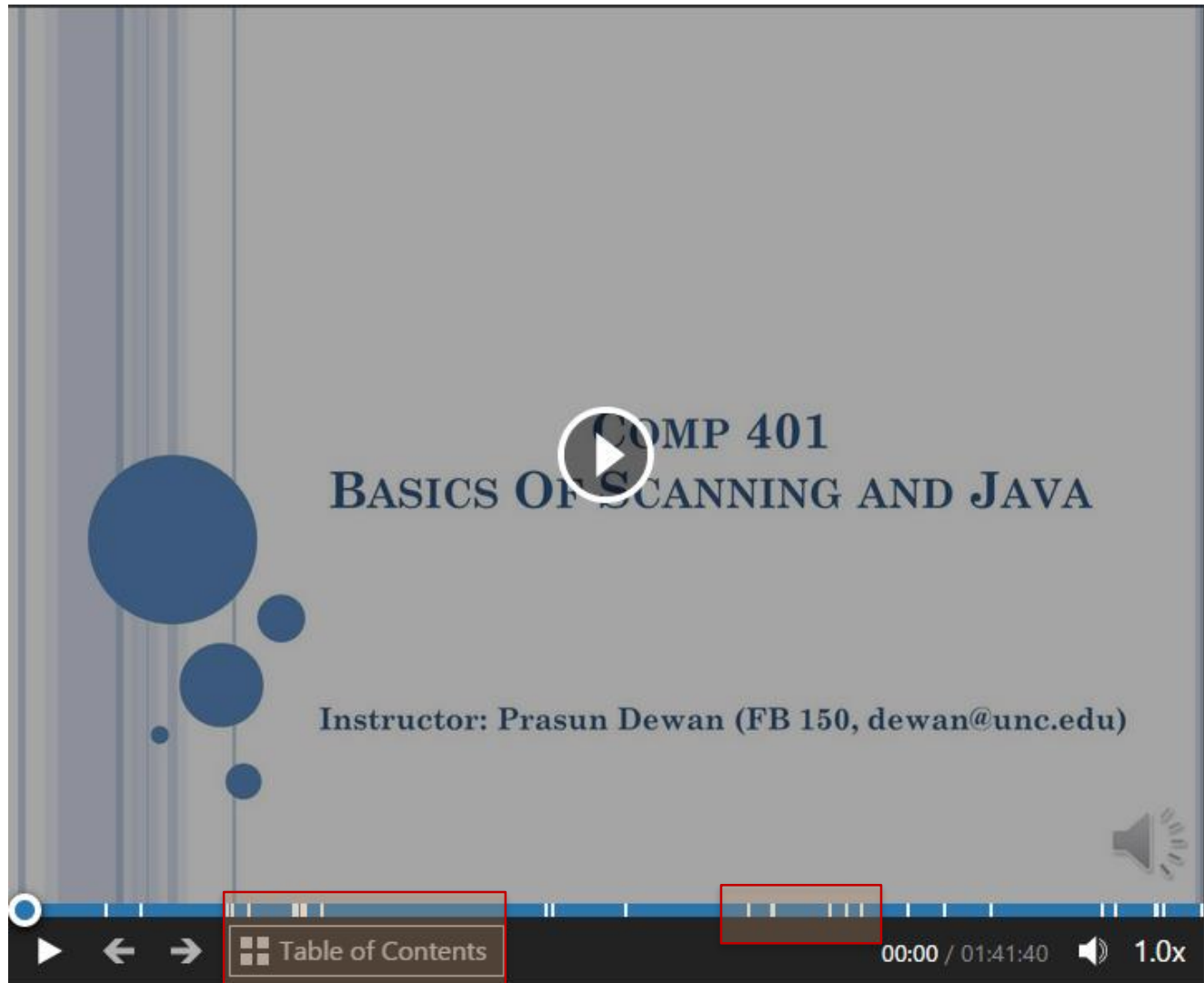
Play 2X,
rewind, pause, fast-forward to match understanding pace

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

PPT modes allow slide-based browsing but requires
downloading PPT



OFFICE MIX



The image shows a video player interface. The main content area displays a slide with the following text:

COMP 401
BASICS OF SCANNING AND JAVA

Instructor: Prasan Dewan (FB 150, dewan@unc.edu)

The video player controls at the bottom include a play button, a progress bar, a volume icon, and a speed control set to 1.0x. A red box highlights a section of the progress bar, and a tooltip labeled "Table of Contents" is visible over it.

Slide markers



SLIDE-BASED BROWSING

The screenshot displays a slide-based browsing interface with four slides visible. The interface has a dark background with a close button (X) in the top right corner. The slides are numbered 1 through 4.

Slide 1: Titled "COMP 401 BASICS OF SCANNING AND JAVA". It includes the instructor's name: "Instructor: Frason Dewan (FB 150, dewan@unc.edu)".

Slide 2: Titled "PROGRAMMING OVERVIEW THROUGH EXAMPLE". It lists the following topics:

- o Problem
- o Algorithm
- o Representation
- o Code

Slide 3: Titled "SCANNING PROBLEM". It lists the following topics:

- o Scanning image for text.
- o Scanning frequencies for radio stations.
- o Finding words in a sentence
- o Finding identifiers, operators, in a program

Slide 4: Titled "SCANNING PROBLEM". It shows a code editor interface with the following elements:

- Input field: "Print argument or option to the program"
- Program name field: "Program name"
- Code area:

```
import java.util.Scanner;
public class Scanner {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        scanner.next();
    }
}
```
- Output field: "output"

At the bottom of the interface, there is a large watermark text: "COMP 401 AND JAVA" and "Instructor: Frason Dewan (FB 150, dewan@unc.edu)".



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

ring.pdf

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

Visited Getting Started help.unc.edu MyUNC The University of Nort...

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);
        }
    }
}
```

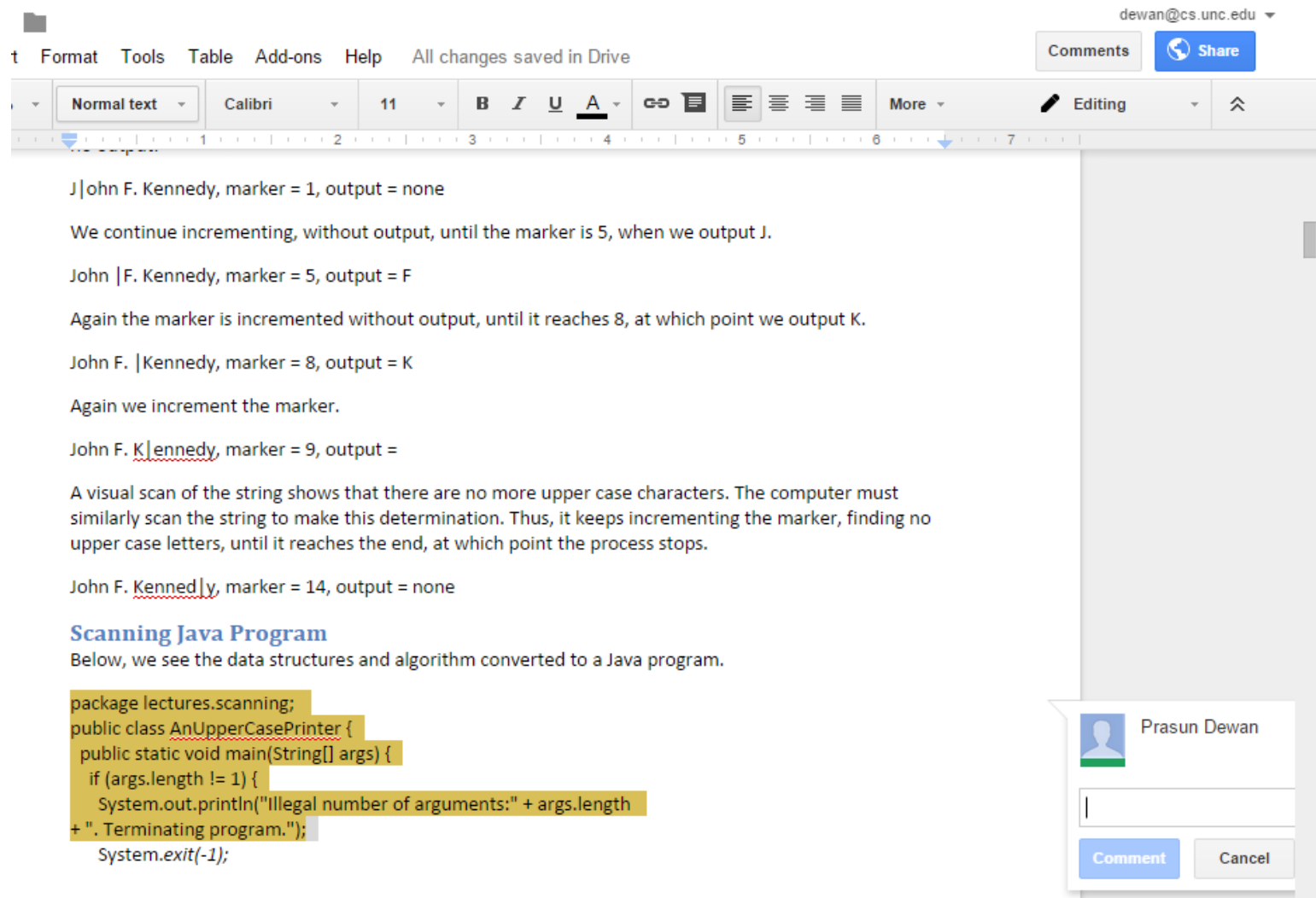
fox automatically sends some data to Mozilla so that we can improve your experience. Choose Wh

Little graphics,
designed for mobile
reading on mobile
computers

Lots of (obvious)
mistakes



PUBLICLY COMMENTABLE GOOGLE DOCS



The screenshot shows a Google Docs interface. At the top right, the user's email is 'dewan@cs.unc.edu'. The menu bar includes 'Format', 'Tools', 'Table', 'Add-ons', and 'Help'. A status bar indicates 'All changes saved in Drive'. The toolbar shows 'Normal text', 'Calibri', '11', and various formatting options like bold, italic, underline, and text color. The main text area contains several paragraphs of text, some with red squiggly lines under the words 'Kennedy' and 'y'. A comment box is open on the right side, showing the user's profile picture and name 'Prasun Dewan', a text input field, and 'Comment' and 'Cancel' buttons.

John 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);
        }
    }
}
```



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](#)

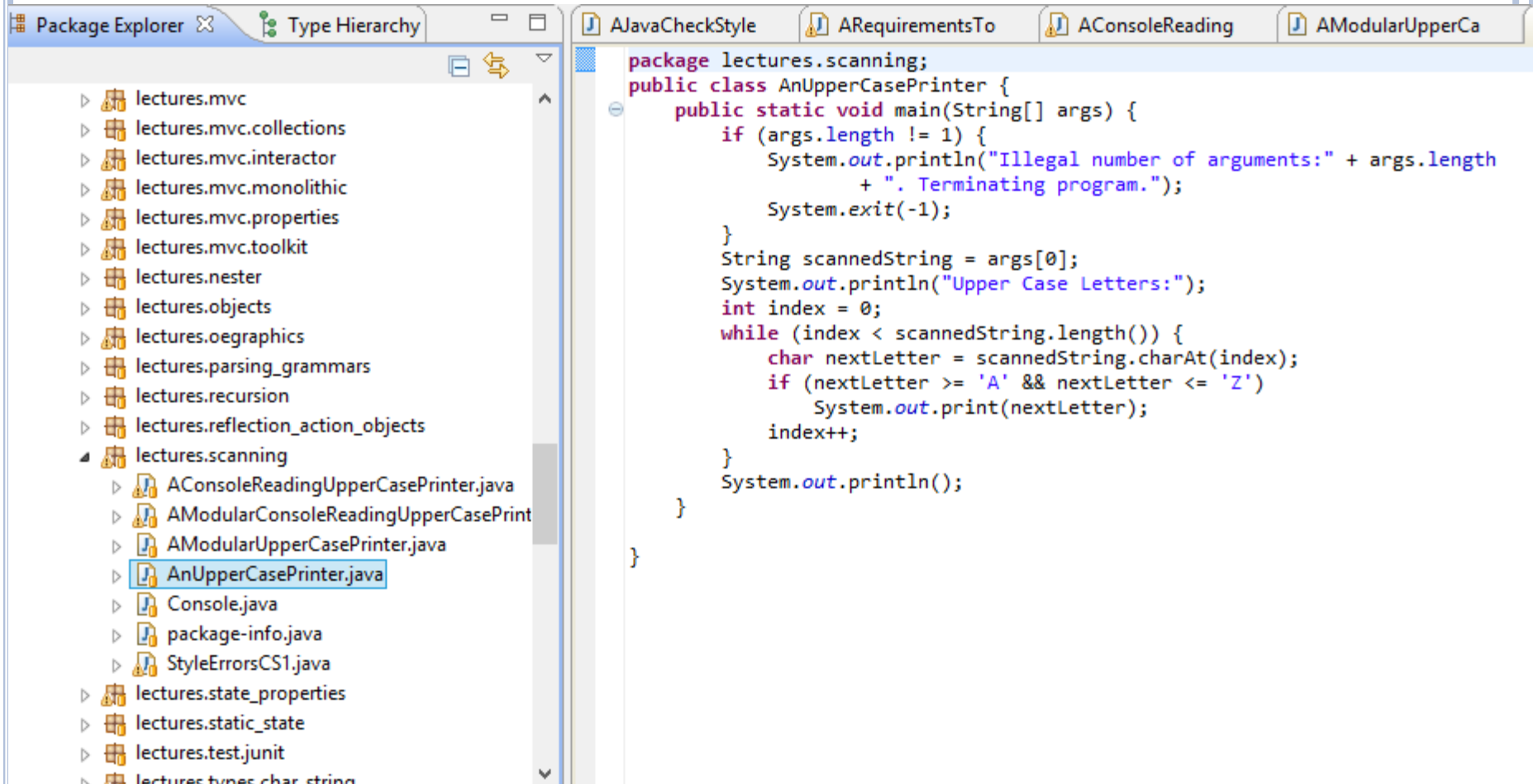
AnUpperCasePrinter.java

```
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



The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows a project structure with several packages under 'lectures.mvc' and 'lectures.scanning'. The 'lectures.scanning' package is expanded, showing files like 'AnUpperCasePrinter.java' which is selected. The main editor window shows the code for 'AnUpperCasePrinter.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();
    }
}
```



WEB SITE LINKS

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



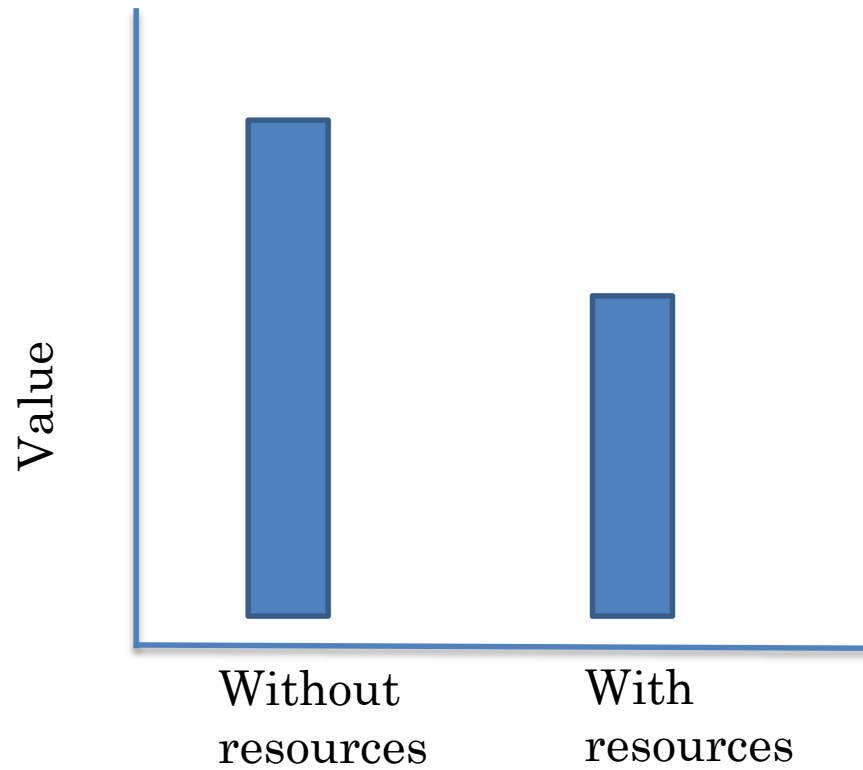
REGULAR AND FLIPPED CLASS

- In regular class
 - concepts are formally taught by instructor during lecture time.
 - external learning resources are used to solidify/clarify concepts.
- In flipped class:
 - concepts are learnt by students through external learning (books and online) resources.
 - lecture time is used for hands-on activities.

A variety of learning resources available!



USEFULNESS OF LIVE LECTURE



WHAT DO WE DO IN FLIPPED CLASS?

Problems related to homework?

Works when complexity of work is high

Recitations aim to provide templates for assignment



WHAT DO WE DO IN FLIPPED CLASS?

Homework?

Deep thinking done solo?

Limited discussion with classmates?



TRADEOFFS IN REGULAR CLASS STYLES

Distill concepts in PPT or whiteboard.

Do live programming

JAVA PROGRAM STRUCTURE

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

The screenshot shows an IDE window with the following code and annotations:

```
148 System.out.println("Please enter a string with uppercase and lower case
149 /* Use of library, nothing conceptual here, just some syntax */
150 /*
151 * The next statement needs to be executed once per console reading prog
152 */
153 Scanner scanner = new Scanner(System.in);
154 /*
155 * The next statement gives next line input by the user on the console.
156 * hover on method name to see its description
157 */
158 String scannedString = scanner.nextLine();
159
160 int index = 0; // variable names should be camel case starting with lower
161 /**
162 * SCANNING AND WHILE LOOPS
163 * (???) A token produced by scanning a string can have non consecutive
164 * characters of the string.
165 * (???) A string scanned by a scanner is a concatenation of all the tokens
166 * detected by the scanner.
167 *
168 * Syntax of Java while loop shown below
```

Annotations include: "Use of library, nothing conceptual here, just some syntax", "The next statement needs to be executed once per console reading program", "The next statement gives next line input by the user on the console. hover on method name to see its description", "variable names should be camel case starting with lower", "SCANNING AND WHILE LOOPS", "A token produced by scanning a string can have non consecutive characters of the string.", "A string scanned by a scanner is a concatenation of all the tokens detected by the scanner.", "Syntax of Java while loop shown below".

Faster paced, concept intensive, use animation and graphics, advanced classes

Slower pace, more laborious, learn programming environment commands, intro classes



CHANGING LIVE PROGRAMMING

I code in class, you watch.

You learn more as a driver than passenger



PRAXIS: FLIP CODE AND CONCEPT PRESENTATION

```
public class AConsoleReadingUpperCasePrinter {
```

```
/**
```

```
 * MAIN METHOD HEADER
```

```
 * Syntax of main method shown below.
```

```
 * Methods correspond to procedures and functions
```

```
 * Method names should be camel case starting with
```

```
 * Everything before the first curly brace is the
```

```
 */
```

```
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.
```

JAVA PROGRAM STRUCTURE

The screenshot shows a Java code editor with the following code and annotations:

```
package lectures_scanning;
public class AnUpperCasePrinter {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Illegal number of arguments");
            System.out.println("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();
    }
}
```

Annotations in the image:

- A box labeled "Must have this procedure header in executable program" points to the package and class declarations.
- A box labeled "Predefined internal library operations" points to `System.out.println()` and `System.exit(-1)`.
- A box labeled "Print on new vs. previous line" points to the `System.out.println()` at the end of the `main` method.

The script followed by instructor (including questions) whole doing live programming is embedded in the program template

Concept presentation in code rather than code in concept presentation

Sakai quiz to record answers derived in open-book discussion with peers and instructors

