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
Course Overview
  Course Syllabus in UNC Format
  Instructor
  Teaching Assistants
  Lectures Time and Location
  Recitations and Associated Material
Exams
  Exam Schedule (Subject to change)
  Old Exams
  Exam Information
Resources
  Getting Help and Class Discussion
  Course Web Site (From CS)

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

Use index and local web search to find parts
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”

<table>
<thead>
<tr>
<th>Resources by Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unit</strong></td>
</tr>
<tr>
<td>Course Information (8/18, 8/20)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Scanning</strong></th>
<th><strong>PowerPoint PDF YouTube Mix</strong></th>
<th><strong>Docx PDF Drive</strong></th>
<th><strong>Scanning Visualization</strong></th>
<th><strong>Number Scanner Checks File</strong></th>
<th><strong>lectures.scanning Package</strong></th>
</tr>
</thead>
</table>
# Before Next Class

<table>
<thead>
<tr>
<th>JDK Download</th>
<th>PowerPoint PDF</th>
<th></th>
<th></th>
<th></th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse Install &amp; Use</td>
<td>PowerPoint PDF</td>
<td>Installing JDK on Mac</td>
<td>PDF</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Checkstyle with UNC Checks Install</td>
<td>PowerPoint PDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ObjectEditor Install</td>
<td>PowerPoint PDF</td>
<td>PDF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Importing Git Project (JavaTeaching)</td>
<td>PowerPoint PDF</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Specifying Main Args in Eclipse</td>
<td>PowerPoint PDF</td>
<td>PDF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debugging in Eclipse</td>
<td>PowerPoint PDF</td>
<td>PDF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relevant Java Syntax</td>
<td>PowerPoint PDF</td>
<td>PDF</td>
<td></td>
<td></td>
<td>lectures.java_syntax_overview_package ✓</td>
</tr>
<tr>
<td>Scanning</td>
<td>PowerPoint PDF</td>
<td>YouTube Mix</td>
<td>Docx PDF Drive</td>
<td>Scanning Visualization</td>
<td>Number Scanner Checks File</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Component Complexity</th>
<th>No. of Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>Introductory Programming</td>
</tr>
<tr>
<td>401</td>
<td>Intermediate Programming</td>
</tr>
<tr>
<td>410</td>
<td>Data Structures</td>
</tr>
</tbody>
</table>

- **Small number of simple code fragments connected by you**
- **Large number (~40) of simple code fragments connected by you**
- **Small number of complex code fragments connected by you**

Only the optional compiler course will involve more components!
**Layered Assignments = Project**

<table>
<thead>
<tr>
<th>Course Information (8/18)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Bridge Scene - 1st day (long)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bridge Scene - 2nd day (short)</td>
</tr>
</tbody>
</table>

- **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 4 5 6 2 5 3 0 0 0 " (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.

---

<table>
<thead>
<tr>
<th>Students have varying interests and abilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make up or insurance against bad grade in other assignments or exams</td>
</tr>
<tr>
<td>Better to give early without extra credit than late with</td>
</tr>
<tr>
<td>But if you are already late, might as well do extra credit to make up for late points</td>
</tr>
</tbody>
</table>
## 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. `Integer.parseInt()` takes a string consisting of digits and converts into an integer.

<table>
<thead>
<tr>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forbid use of certain Java libraries</td>
</tr>
<tr>
<td>Goal is not to teach Java and its libraries</td>
</tr>
<tr>
<td>It is to teach you how to build these libraries</td>
</tr>
<tr>
<td>Usually Java features not covered in class will be banned</td>
</tr>
<tr>
<td>Correctness is only one of the goals</td>
</tr>
<tr>
<td>Require use of certain programming techniques</td>
</tr>
<tr>
<td>Program must also be efficient and well crafted</td>
</tr>
</tbody>
</table>
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 track 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 spaces 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](mailto: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
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

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

- illegalMethodCall: (IllegalMethodCalls.java:123) called disallowed method bar---> String.split
- illegalMethodCall: (IllegalMethodCalls.java:126) called disallowed method bar---> String.split
Number Scanner
Checks File
<property name="expectedTypes" value="main.Assignment(.*, mp.scanner.Scanner)"/>
<module name="unc.tools.checkstyle.ANonCachingTreeWalker">
  <property name="severity" value="warning"/>
  <module name="ExpectedDeclaredSignatures">
    <property name="severity" value="warning"/>
    <property name="expectedSignatures" value="main.Assignment1 = processInput
indexOfNot: String; char; int->int // EC, grail.scanner.ScanningIterator = indexOfNot

module name="MissingMethodCall">
  <property name="severity" value="warning"/>
  <property name="expectedCalls" value="main.Assignment1 = processInput:->void
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

```java
//OFrame editor2 = ObjectEditor.edit(interpreterView);
OFrame editor = ObjectEditor.edit(bridgeScene);
bridgeScene.setOFrame(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

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

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

Notes:
- Command object invoked
  - Command objects returned from say and move are invoked.
  - Couldn't find a parse invoke that called undo.

Send Assignment

COMP 401-038
Assignment
Preconditions, Commands, Threads, Animation (Assignment 10)
Onyen vitkus
Password: ***************
Send Assignment
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());
    yAxis.setY(toYAxisY());
    ...
}
OBJECT EDITOR: 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.Anonymous(,*) defined
# Downloads and Consent Form

<table>
<thead>
<tr>
<th>Downloads</th>
</tr>
</thead>
</table>
| 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

<table>
<thead>
<tr>
<th>Component Complexity</th>
<th>No. of Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>410</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td></td>
</tr>
<tr>
<td>401</td>
<td></td>
</tr>
</tbody>
</table>

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?
Java Program Structure

```java
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
**PowerPoint of Slides**

---

**Java Program Structure**

```java
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.println(nextLetter);
            index++;
        }
        System.out.println();
    }
}
```

- Must have this procedure header in executable program
- Predefined internal library operations
- Print on new vs. previous line
SLIDE SHOW ➔ SYNCHRONIZED RECORDING AND ANIMATIONS

JAVA PROGRAM STRUCTURE

```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 (Character.isUpperCase(nextLetter))
                System.out.print(nextLetter);
            index += 1;
        }
    }
}
```

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

JAVA PROGRAM STRUCTURE

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

Print on new vs. previous line

Must have this procedure header in executable program

Predefined internal library operations
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

COMP 401
BASICS OF SCANNING AND JAVA

Instructor: Prasun Dewan (FB 150, dewan@unc.edu)
SLIDE-BASED BROWSING

Comp 401
Basics of Scanning and Java

Problem
Algorithm
Representation
Code

Programming Overview Through Example

Scanning Problem

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

Algorithm

- Description of solution to a problem
- Can be in any "language"

Problem

Input stream

- tokens
- output stream
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
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. Kennedy, 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. Kennedy, marker = 14, output = none

**Scanning Java Program**

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

```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);
        }
    }
}
```

Lots of (obvious) mistakes

Little graphics, designed for mobile reading on mobile computers
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. Kennedy, marker = 9, output = none

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. Kennedy, marker = 14, output = none

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

```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);
        }
    }
}
```

Prasun Dewan
package lectures.scanning;
import util.annotations.WebDocuments;
@WebDocuments({"Lectures/Scanning.pptx", "Lectures/Scanning.pdf", "Video"})
public class AnUpperCasePrinter {
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println("Illegal number of arguments:" + args[1] + ". 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();
  }
}
Eclipse Java Project of Lecture Code on Git
DESIGN SPACE OF STUDY MODES!

- You Tube video
- Slides without audio
- Slides with unsynchronized audio
- Slides with synchronized audio
- Office Mix
- Word PDFs
- Shared Google Docs
- Eclipse Project
-Browsable Project
# Web Site Links

<table>
<thead>
<tr>
<th>Scanning</th>
<th>PowerPoint</th>
<th>Docx</th>
<th>Scanning Visualization</th>
<th>Number Scanner</th>
<th>lectures.scanning Package</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YouTube</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mix</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What do we do in class?

Live Lecture?

- Value
  - Without resources
  - With resources
Discussion is About Concrete Programs

Java Program Structure

```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.println(nextLetter);
            }
            index++;
        }
    }
}
```

Pace at which you understand lecture in general and code in particular varies
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.
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.
 * In method body, you do all your code.
 */
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

- Come to class

Don’t

- 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]
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
**Grade Distribution (Current Plan)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>22%</td>
</tr>
<tr>
<td>Final</td>
<td>28%</td>
</tr>
<tr>
<td>In-Class Work (Recitations &amp; Lecture Quizzes, Discussion, Attendance)</td>
<td>15%</td>
</tr>
<tr>
<td>Assignments</td>
<td>35%</td>
</tr>
<tr>
<td>Fudge Factor (Special participation, other distinguishing factors), particularly useful for borderline cases</td>
<td>5%</td>
</tr>
</tbody>
</table>