Welcome

Lecture #1

CMPSCI 121, Spring 2012

Introduction to Problem Solving with Computers

Prof. Erik Learned-Miller

Please sit all the way at the *inside* of your row.

We' ll start 5 or 10 minutes late today.

Who am I?

- Faculty in Computer Science Department
- Research
  - Computer vision: getting computers to see!
  - Machine learning: getting computers to learn!

Top 10 Reasons to Take This Class

- 1. Have fun!
- 2. Learn how to think rigorously.
- 3. Learn to make things work.
- 4. Gain confidence through demystification. Believe you can understand.
- 5. Become useful to everyone around you.

Top 10 Reasons to Take This Class

- 6. Work in whatever area you want.
- 7. Predict the future.
- 8. Understand the most complex machines ever built.
- 9. Challenge yourself.
- 10. Jobs!

Warning

- This class is hard work!
- If you have never programmed, you will find it is *very unnatural.*
  - Humans are not made to be rigid and precise. But that is what programming requires.
  - It is a very difficult skill to learn.
- Lots and lots of work.
- Lots and lots of new ideas.

But…

- Once you have learned to program, you will find…
  - You think more precisely than you used to
  - You are better at solving problems than you used to be
  - You are *smarter* than you used to be
  - ~~You are more attractive ….~~

You will start to realize you can understand things like…

- how Google works:
  - Writing code to do searches
- how video games work:
  - Writing part of your own video game.
- how a word processor works:
  - Writing code to manipulate text.
- You may start to wonder how other things work….

In summary

- If you stick with this class, I think many of you will be *blown away* by how far you have come in one semester.

Logistics

Getting into this class

- The class is way oversubscribed.
- We are not registering new students unless one of the following holds:
  - You are a computer science major
  - You are in another major which *requires* this class.
  - You are a senior and you have to take this class to graduate.

Staff

- TAs:
  - Grant Sherrick (Section 1)
  - Hee-Tae Jung (Section 2)
  - Manju Narayana (Section 3)
  - Andrew Kae (Section 4)
  - Melissa Frechette (Section 5)

TAs

- Lead sections
- All except Hee-Tae have office hours. Please use them for help!
- You can go to any TA for help, not just your section leader.
- TAs will also respond to emails.

Intro CS Courses

- CS 105: Computer Literacy
- CS 145: Representing, Storing, and Retrieving Information (IT program)
- CS 191P: Python Programming
  - IMPORTANT: If you only want to take ONE programming class, take this one. Not intended for majors.
- CS 121: Intro Programming (THIS CLASS)
  - Assumes literacy but no programming.
  - Potential majors: Take as soon as you can!
  - 4 credits. Moves fast!
- CS 187: Data structures
  - Assumes 121 or equivalent.

What' s the course like?

- You do *not* need previous programming experience.
- Will write lots of simple programs.
- Even if you have programmed before, our approach will give you new ways of thinking.
- The sections will give you additional examples and re-enforce concepts for lecture. The *best* place to ask questions.

What's the course like?

- Not for people simply looking for a breadth requirement they think will be easier than Calculus.
- Lots and lots of material.
- Preparation for the Computer Science major.

Details, details, …

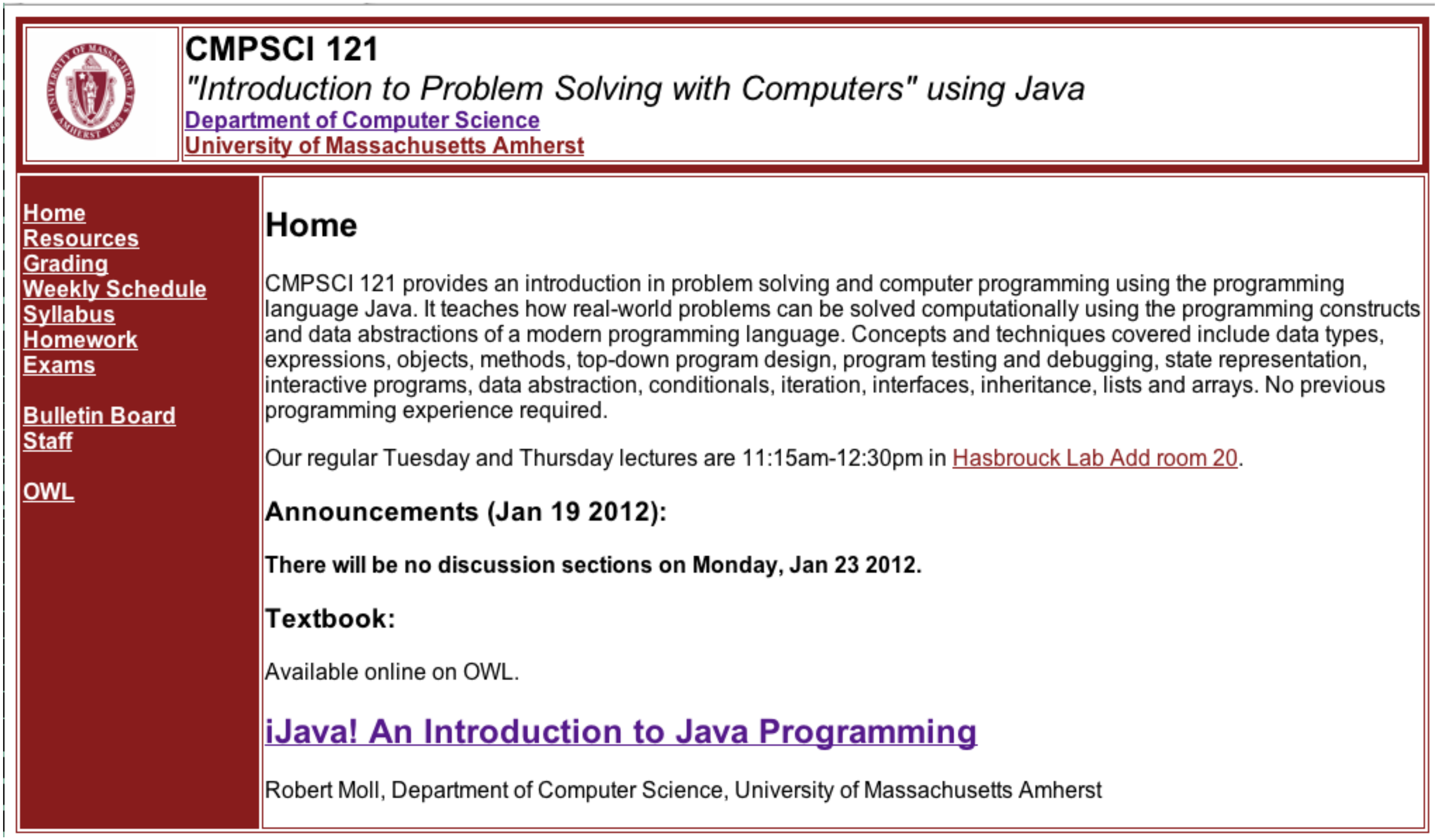
- Everything about syllabus, grading, assignments, exams, labs, textbook, software is or will be at

  http://www.cs.umass.edu/~elm/Teaching/121_S12/index.html

Resources

1. Course Web Page
2. OWL
3. E-book
4. DrJava

1. Course Web Page



CMPSCI 121

"Introduction to Problem Solving with Computers" using Java

Department of Computer Science
University of Massachusetts Amherst

Home

CMPSCI 121 provides an introduction in problem solving and computer programming using the programming language Java. It teaches how real-world problems can be solved computationally using the programming constructs and data abstractions of a modern programming language. Concepts and techniques covered include data types, expressions, objects, methods, top-down program design, program testing and debugging, state representation, interactive programs, data abstraction, conditionals, iteration, interfaces, inheritance, lists and arrays. No previous programming experience required.

Our regular Tuesday and Thursday lectures are 11:15am-12:30pm in Hasbrouck Lab Add room 20.

Announcements (Jan 19 2012):

There will be no discussion sections on Monday, Jan 23 2012.

Textbook:

Available online on OWL.

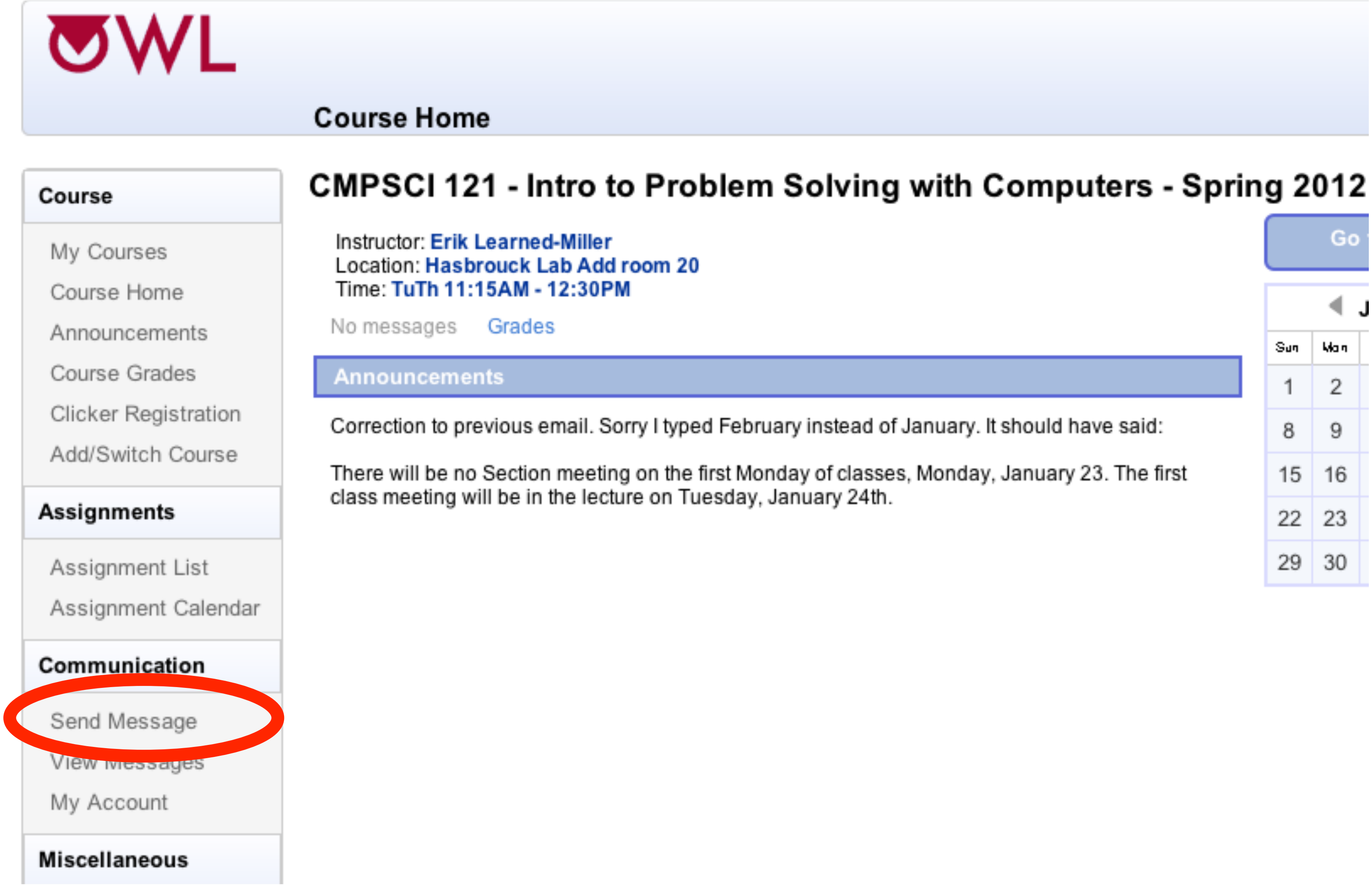
iJava! An Introduction to Java Programming

Robert Moll, Department of Computer Science, University of Massachusetts Amherst

Email for help

- Email through OWL. Do not use professor or TA personal email addresses except in unusual circumstances.
- Dear Staff…
  - After hours of searching the course web site, high and low, this way and that way, I cannot find the answer to my question….

Email in OWL



Course Home

CMPSCI 121 - Intro to Problem Solving with Computers - Spring 2012

Instructor: Erik Learned-Miller
Location: Hasbrouck Lab Add room 20
Time: TuTh 11:15AM - 12:30PM

No messages Grades

Announcements

Correction to previous email. Sorry I typed February instead of January. It should have said:

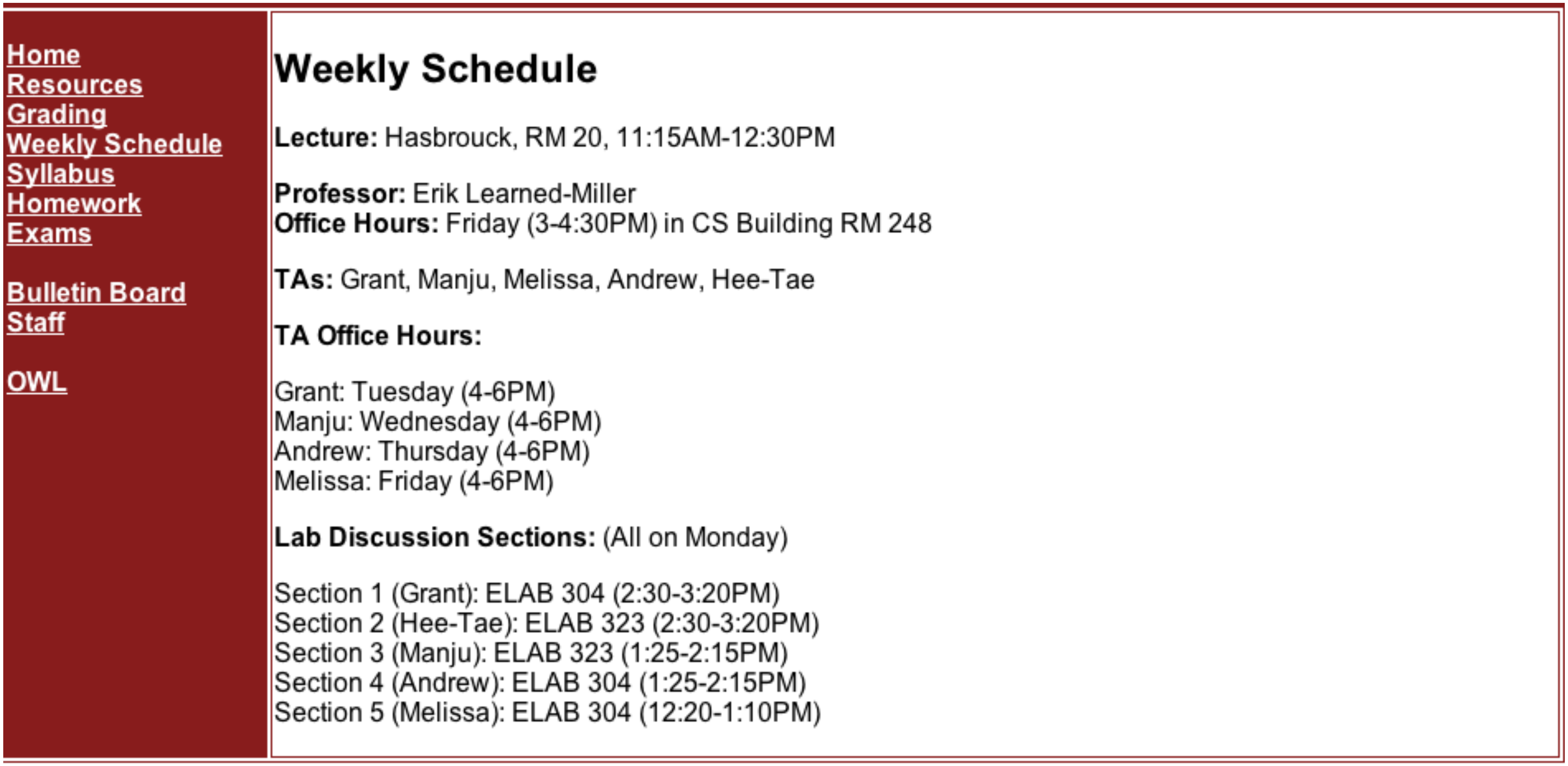
There will be no Section meeting on the first Monday of classes, Monday, January 23. The first class meeting will be in the lecture on Tuesday, January 24th.

Our First Concept from Computer Science!

- Constant time algorithms vs. linear time algorithms
  - Let n be the number of students in a course.
  - Preparing a web page for a course is constant-time, also written as O(1).
    - 10 students: 1 hour to prepare web page
    - 100 students: 1 hour to prepare web page
  - Answering email for a course is a linear-time algorithm, also written as O(n).
    - 10 students: 300 minutes of email per semester
    - 100 students: 3000 minutes of email per semester!

Weekly schedule

(TA office hours subject to change.)



Weekly Schedule

Lecture: Hasbrouck, RM 20, 11:15AM-12:30PM

Professor: Erik Learned-Miller
Office Hours: Friday (3-4:30PM) in CS Building RM 248

TAs: Grant, Manju, Melissa, Andrew, Hee-Tae

TA Office Hours:

Grant: Tuesday (4-6PM)
Manju: Wednesday (4-6PM)
Andrew: Thursday (4-6PM)
Melissa: Friday (4-6PM)

Lab Discussion Sections: (All on Monday)

Section 1 (Grant): ELAB 304 (2:30-3:20PM)
Section 2 (Hee-Tae): ELAB 323 (2:30-3:20PM)
Section 3 (Manju): ELAB 323 (1:25-2:15PM)
Section 4 (Andrew): ELAB 304 (1:25-2:15PM)
Section 5 (Melissa): ELAB 304 (12:20-1:10PM)

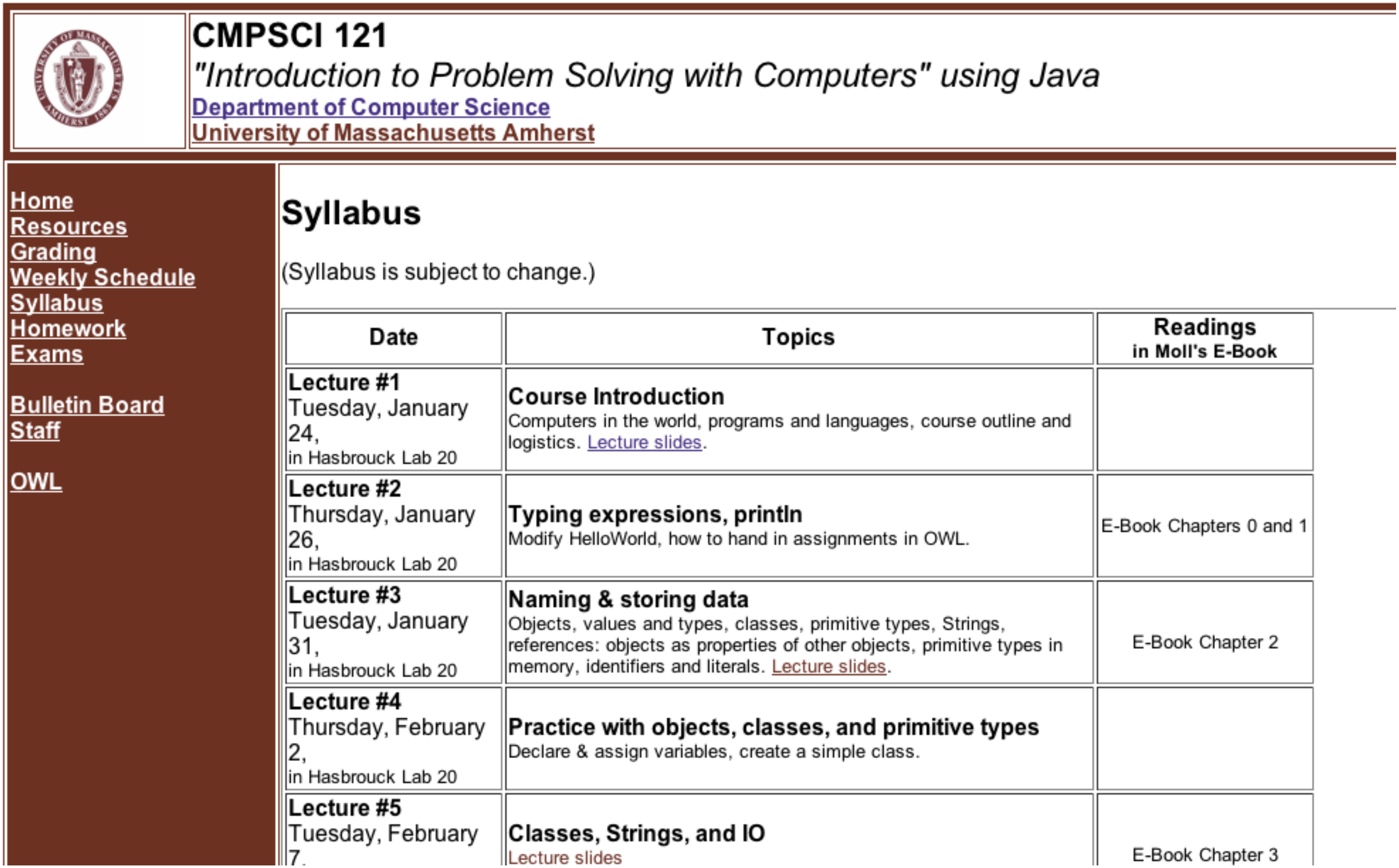
My office hours

- Starting next week (none this week)
- Friday, 3:00-4:30
- Use ‘em!
  - Student: “*Professor, I’ve been having problems all semester, and I want you to fix all my problems today.”*
  - *Me: “How come this is the first time you’re coming to talk to me?”*

Extra Help

- Learning Resource Center
- Tutorials start in a couple of weeks.

Syllabus Web Page



CMPSCI 121

"Introduction to Problem Solving with Computers" using Java

Department of Computer Science
University of Massachusetts Amherst

Syllabus

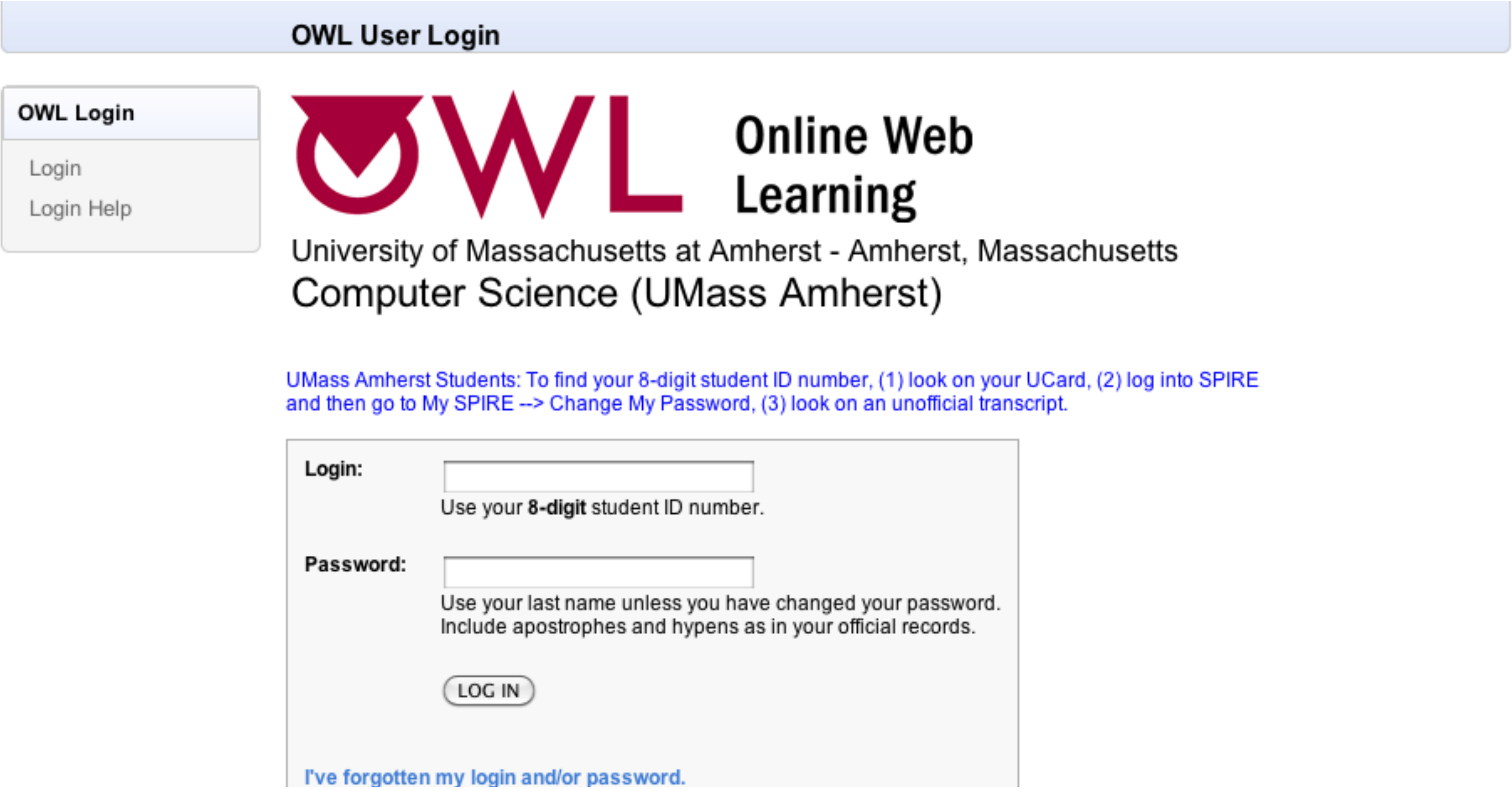
(Syllabus is subject to change.)

Date	Topics	Readings in Moll's E-Book
Lecture #1 Tuesday, January 24, in Hasbrouck Lab 20	**Course Introduction** Computers in the world, programs and languages, course outline and logistics. Lecture slides.	
Lecture #2 Thursday, January 26, in Hasbrouck Lab 20	**Typing expressions, println** Modify HelloWorld, how to hand in assignments in OWL.	E-Book Chapters 0 and 1
Lecture #3 Tuesday, January 31, in Hasbrouck Lab 20	**Naming & storing data** Objects, values and types, classes, primitive types, Strings, references: objects as properties of other objects, primitive types in memory, identifiers and literals. Lecture slides.	E-Book Chapter 2
Lecture #4 Thursday, February 2, in Hasbrouck Lab 20	**Practice with objects, classes, and primitive types** Declare & assign variables, create a simple class.	
Lecture #5 Tuesday, February 7,	**Classes, Strings, and IO** Lecture slides	E-Book Chapter 3

2. OWL

- On-line Web-based Learning
  - Access electronic textbook through OWL
  - Embedded textbook exercises down through OWL.
    - Automatically graded.
  - Chapter Exercises (short answer homeworks) done in OWL.
    - Automatically graded.
  - Programming problems. Turned in with OWL.
    - Graded by hand.

Logging in to OWL



University of Massachusetts at Amherst - Amherst, Massachusetts

Computer Science (UMass Amherst)

UMass Amherst Students: To find your 8-digit student ID number, (1) look on your UCard, (2) log into SPIRE and then go to My SPIRE --> Change My Password, (3) look on an unofficial transcript.

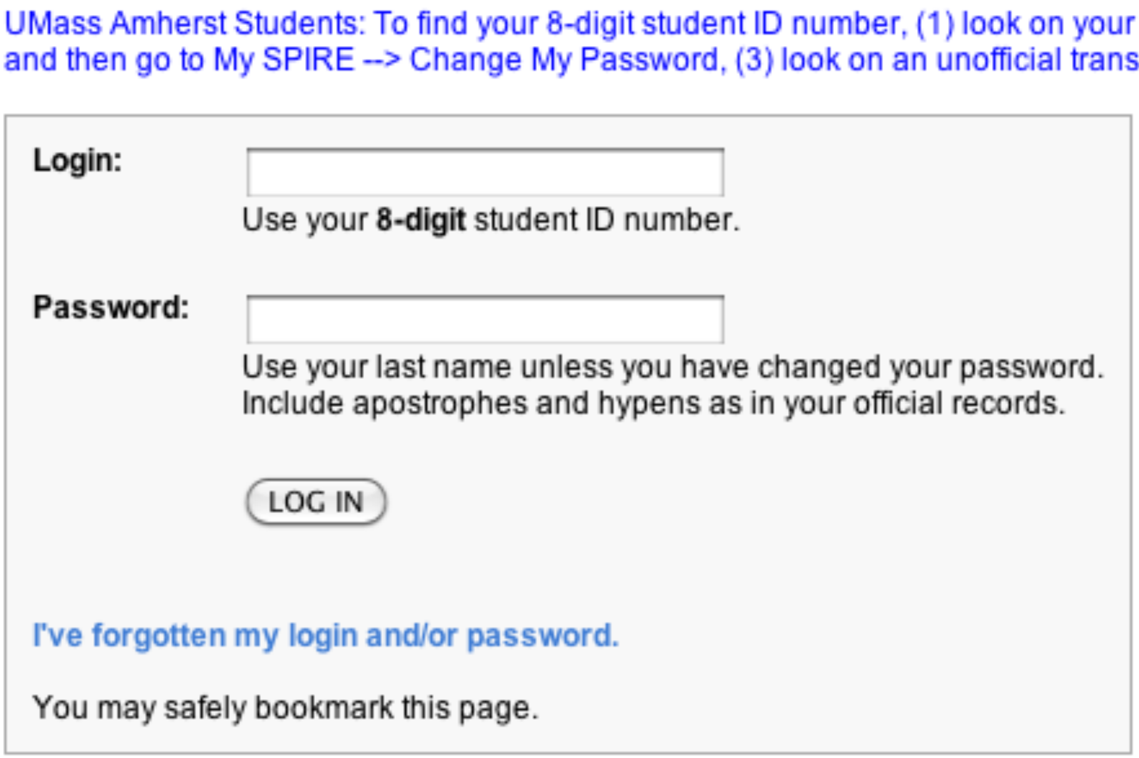
Login:

Use your **8-digit** student ID number.

Password:

Use your last name unless you have changed your password. Include apostrophes and hypens as in your official records.

Logging in to OWL



UMass Amherst Students: To find your 8-digit student ID number, (1) look on your UCard, (2) log into SPIRE and then go to My SPIRE --> Change My Password, (3) look on an unofficial transcript.

Login:

Use your **8-digit** student ID number.

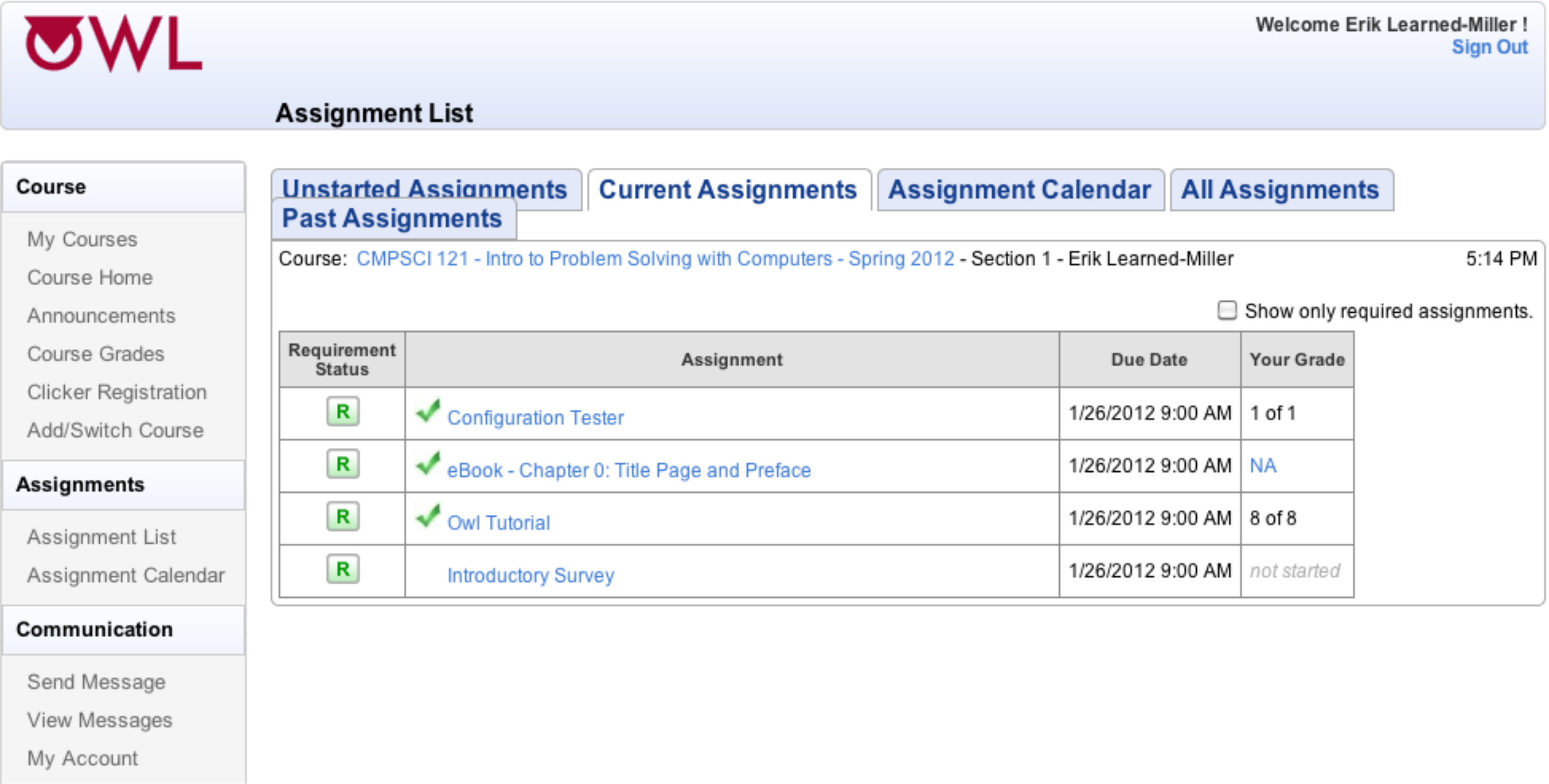
Password:

Use your last name unless you have changed your password. Include apostrophes and hypens as in your official records.

I've forgotten my login and/or password.

You may safely bookmark this page.

OWL “Course”



Welcome Erik Learned-Miller !
Sign Out

Assignment List

Course

- My Courses
- Course Home
- Announcements
- Course Grades
- Clicker Registration
- Add/Switch Course

Assignments

- Assignment List
- Assignment Calendar

Communication

- Send Message
- View Messages
- My Account

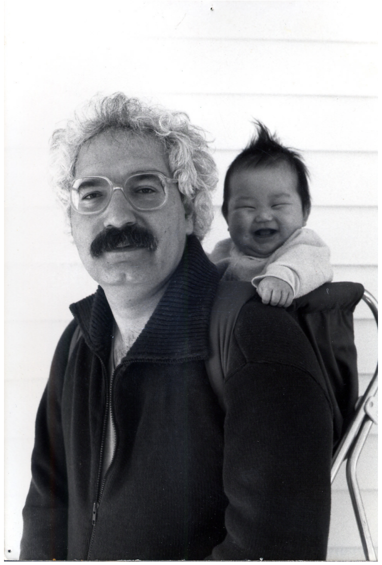
Unstarted Assignments | Current Assignments | Assignment Calendar | All Assignments | Past Assignments

Course: CMPSCI 121 - Intro to Problem Solving with Computers - Spring 2012 - Section 1 - Erik Learned-Miller 5:14 PM

□ Show only required assignments.

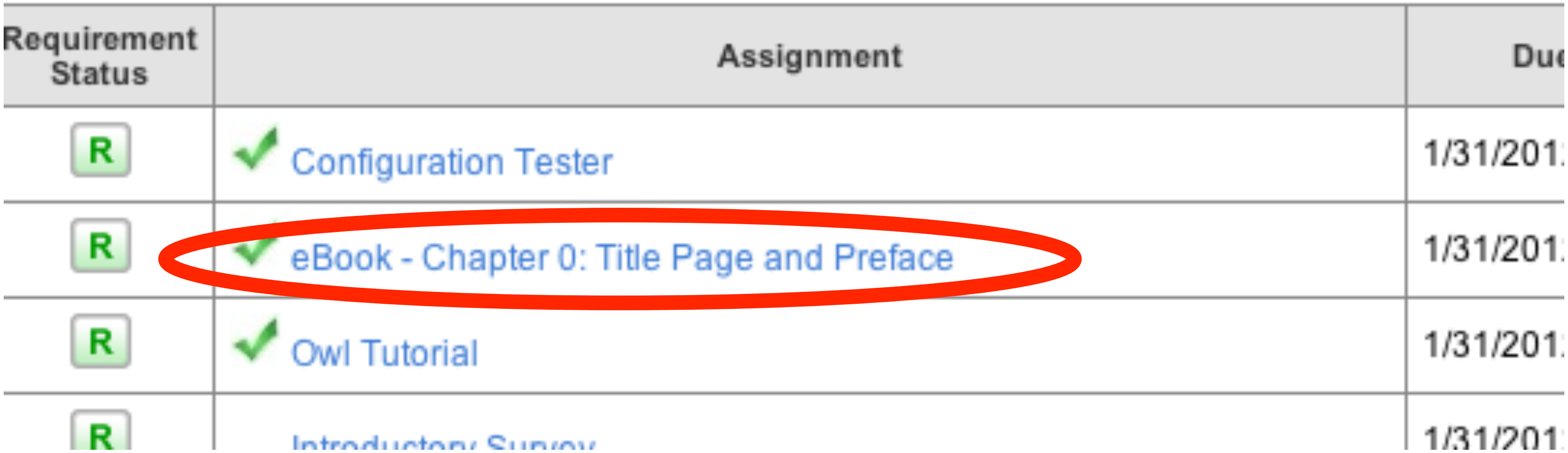
Requirement Status	Assignment	Due Date	Your Grade
R	✓ Configuration Tester	1/26/2012 9:00 AM	1 of 1
R	✓ eBook - Chapter 0: Title Page and Preface	1/26/2012 9:00 AM	NA
R	✓ Owl Tutorial	1/26/2012 9:00 AM	8 of 8
R	Introductory Survey	1/26/2012 9:00 AM	*not started*

3. Online Textbook



- iJava!
  - An Introduction to Java Programming
  - by Robert (Robbie) Moll
- On-line and FREE!!!
- Available through OWL from course web page
- Embedded exercises.
- Read Chapter 0 for next Tuesday!!!
  - (go ahead and read Chapter 1 while you’re at it.)

Accessing On-line textbook in OWL

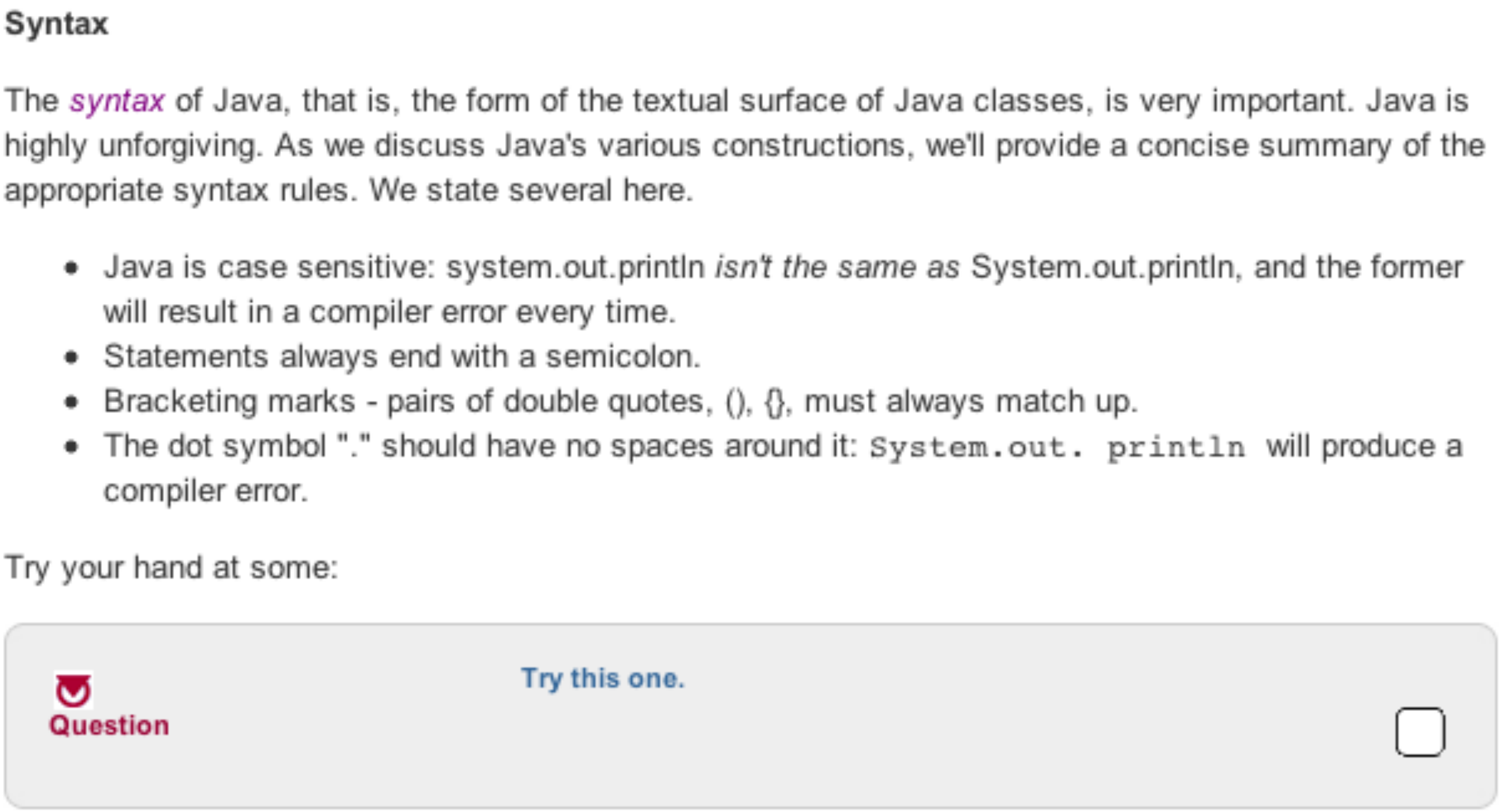


Requirement Status	Assignment	Du
R	✓ Configuration Tester	1/31/201
R	✓ eBook - Chapter 0: Title Page and Preface	1/31/201
R	✓ Owl Tutorial	1/31/201
R	Introductory Survey	1/31/201

Embedded Exercises

- Allows author to embed exercises in the text.
- Your exercises are automatically evaluated before you proceed.

Embedded Exercises



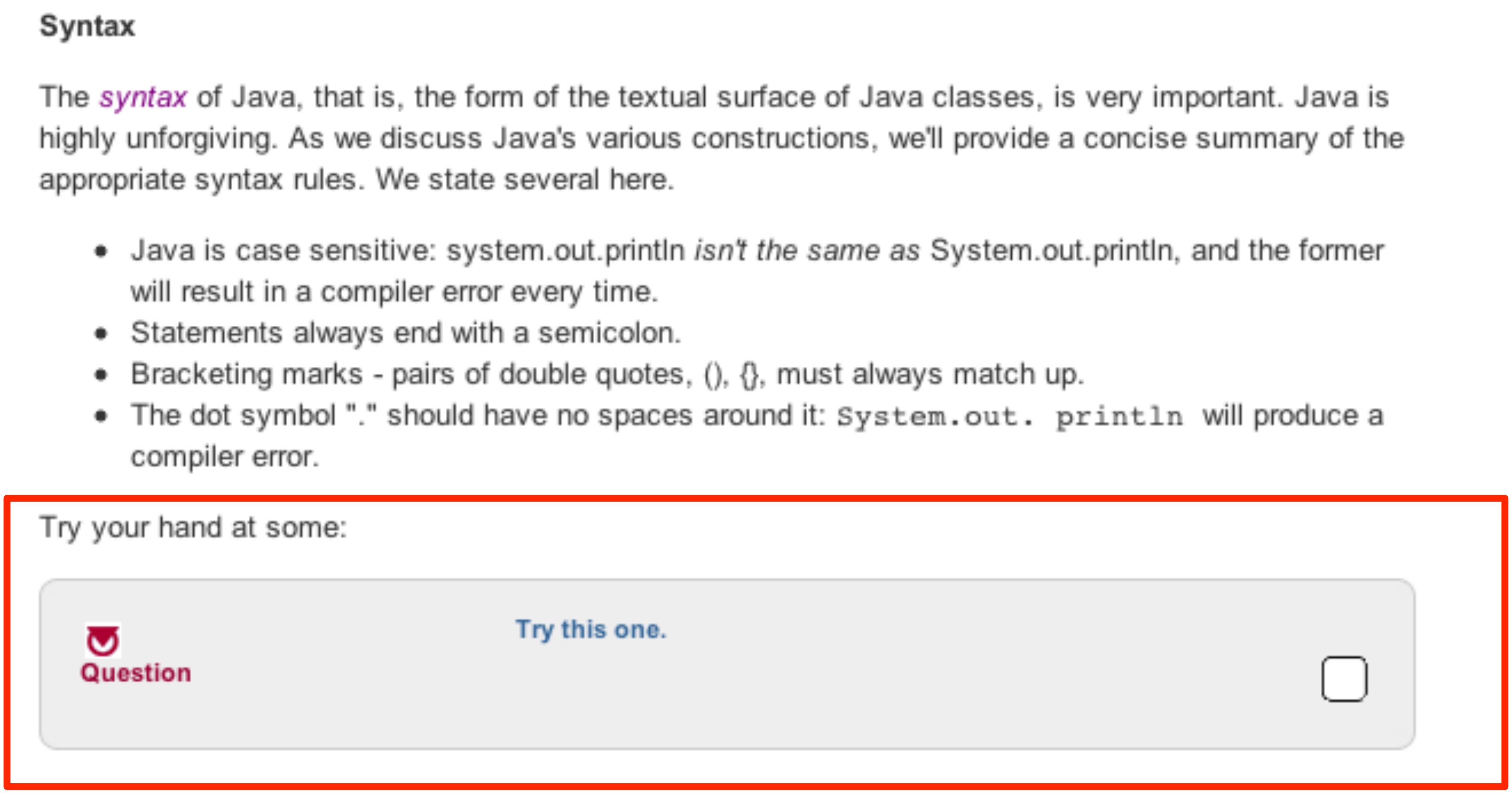
Syntax

The *syntax* of Java, that is, the form of the textual surface of Java classes, is very important. Java is highly unforgiving. As we discuss Java's various constructions, we'll provide a concise summary of the appropriate syntax rules. We state several here.

- Java is case sensitive: system.out.println *isn't the same as* System.out.println, and the former will result in a compiler error every time.
- Statements always end with a semicolon.
- Bracketing marks - pairs of double quotes, (), {}, must always match up.
- The dot symbol "." should have no spaces around it: `System.out. println` will produce a compiler error.

Try your hand at some:

Embedded Exercises



Syntax

The *syntax* of Java, that is, the form of the textual surface of Java classes, is very important. Java is highly unforgiving. As we discuss Java's various constructions, we'll provide a concise summary of the appropriate syntax rules. We state several here.

- Java is case sensitive: system.out.println *isn't the same as* System.out.println, and the former will result in a compiler error every time.
- Statements always end with a semicolon.
- Bracketing marks - pairs of double quotes, (), {}, must always match up.
- The dot symbol "." should have no spaces around it: `System.out. println` will produce a compiler error.

Try your hand at some:

Question

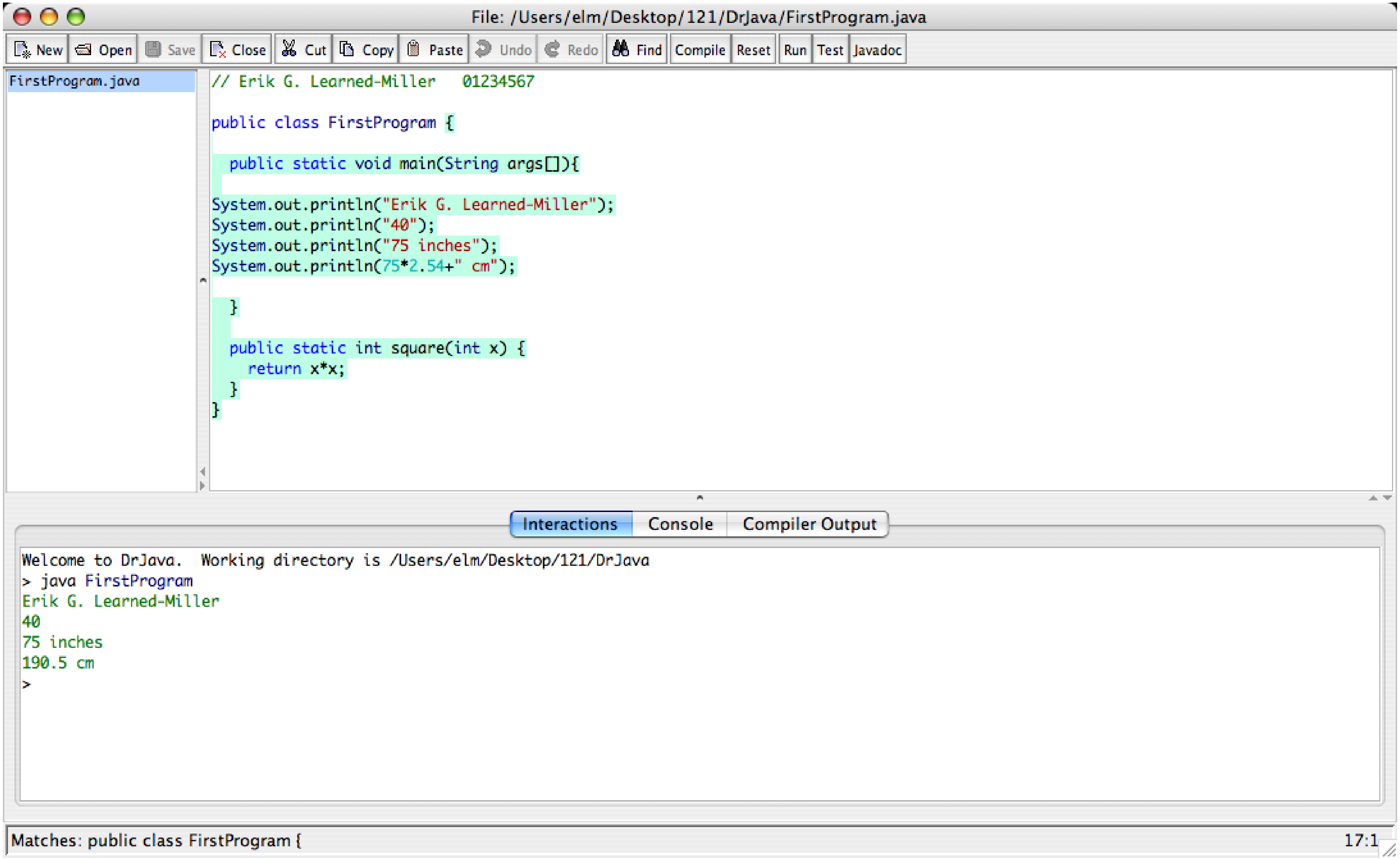
Try this one.

□

4. DrJava

- An Integrated Development Environment, or IDE.
    - Installation explained on web page.
    - Go to “Resources” page.

DrJava Screen Shot



```
// Erik G. Learned-Miller   01234567

public class FirstProgram {

  public static void main(String args[]){

System.out.println("Erik G. Learned-Miller");
System.out.println("40");
System.out.println("75 inches");
System.out.println(75*2.54+" cm");

  }

  public static int square(int x) {
    return x*x;
  }
}
```

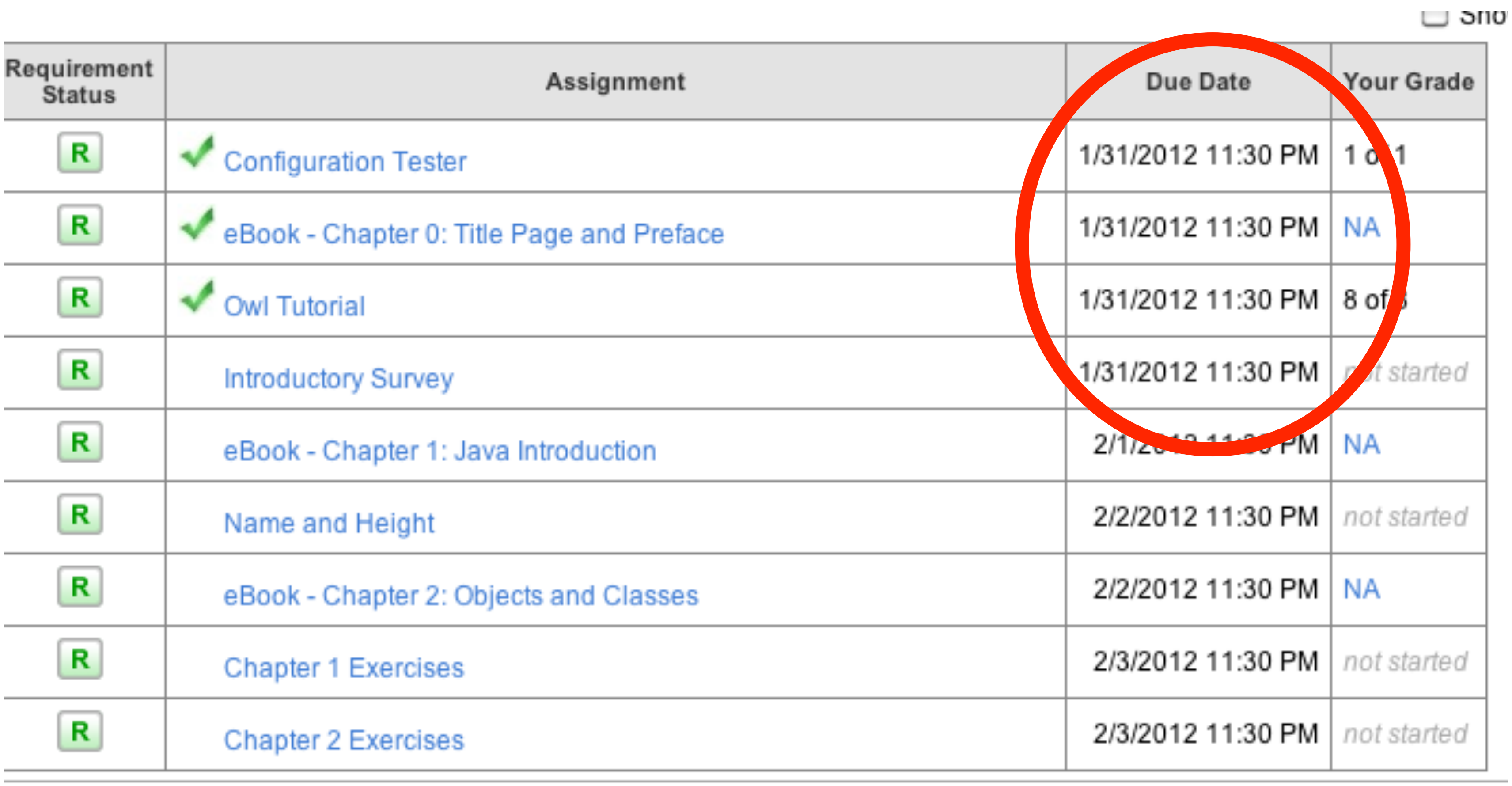
```
Welcome to DrJava.  Working directory is /Users/elm/Desktop/121/DrJava
> java FirstProgram
Erik G. Learned-Miller
40
75 inches
190.5 cm
>
```

Matches: public class FirstProgram {   17:1

Assignments!

- By Next Tuesday at 9am!
  - Log into OWL
    - Access via course web page
  - Do “Configuration Tester” (graded)
  - Do OWL Tutorial (graded)
  - Read Chapter 0 of eBook
  - Download DrJava and start it up.
  - Complete Introductory Survey (graded)

Current OWL view…



□ Sho

Requirement Status	Assignment	Due Date	Your Grade
R	✓ Configuration Tester	1/31/2012 11:30 PM	1 of 1
R	✓ eBook - Chapter 0: Title Page and Preface	1/31/2012 11:30 PM	NA
R	✓ Owl Tutorial	1/31/2012 11:30 PM	8 of 8
R	Introductory Survey	1/31/2012 11:30 PM	*not started*
R	eBook - Chapter 1: Java Introduction	2/1/2012 11:30 PM	NA
R	Name and Height	2/2/2012 11:30 PM	*not started*
R	eBook - Chapter 2: Objects and Classes	2/2/2012 11:30 PM	NA
R	Chapter 1 Exercises	2/3/2012 11:30 PM	*not started*
R	Chapter 2 Exercises	2/3/2012 11:30 PM	*not started*

Let’s get started!

What are Programs?

- Basically just sequences of commands for the computer.

Recipes vs. Programs

Recipe: Bake a ham

- Preheat oven to 450.
- Put ham in a baking pan.
- Put honey glaze on the ham.
- Put ham in oven for 1 hour.

Program: Update savings account

- Type in principal, interest rate, and period of saving.
- Compute total interest earned.
- Add interest to principal.
- Print out new principal amount.

Recipes and Programs… continued

- An *efficient* chef can use an ingredient without worrying about exactly how it is made.
- A *great* chef will ultimately understand how each component is made, to better appreciate the interactions among the parts.

Trade-offs in programming

- Sometimes it is important to accept the code “around you” without delving into it.
- Sometimes you have to understand what is around you.
- This is the art.

Computer Languages

- Medium of communication for programming
  - BASIC
  - C++
  - Java
  - Python
  - Lisp
  - ML
  - HTML

Computer Languages

- Some differences are superficial:
  - BASIC print statement:
    - print
  - Java print statement:
    - System.out.print();
  - C++ print statement:
    - printf();
- Some differences are more significant.

The Good Ol’ Days: BASIC

```
10   Let x=3
20   Let y=5
30   print x+y

RUN

8
```

The Bad New Days: Java

- **`public static void`** `main(String [] args){…`

- `How did we get here?`

Old Programming Styles

- Simpler, but…
- Intended for smaller programs
- Fewer programmers working on the same project
- Had lots of bugs
- Dependent on the computer
- Code was harder to read
- Code was harder to maintain

New Programming Styles

- More complicated in some ways, but…
- Isolate programmer from specific type of computer
- Allow large scale collaborations
- Protect against many types of errors

Readings

- You will learn about computers in general.
- But you will also learn how to program them before you understand everything about them.
- You can drive a car before you understand everything about engines.

DrJava Demo

End of slides