Comp 401
Course Overview

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

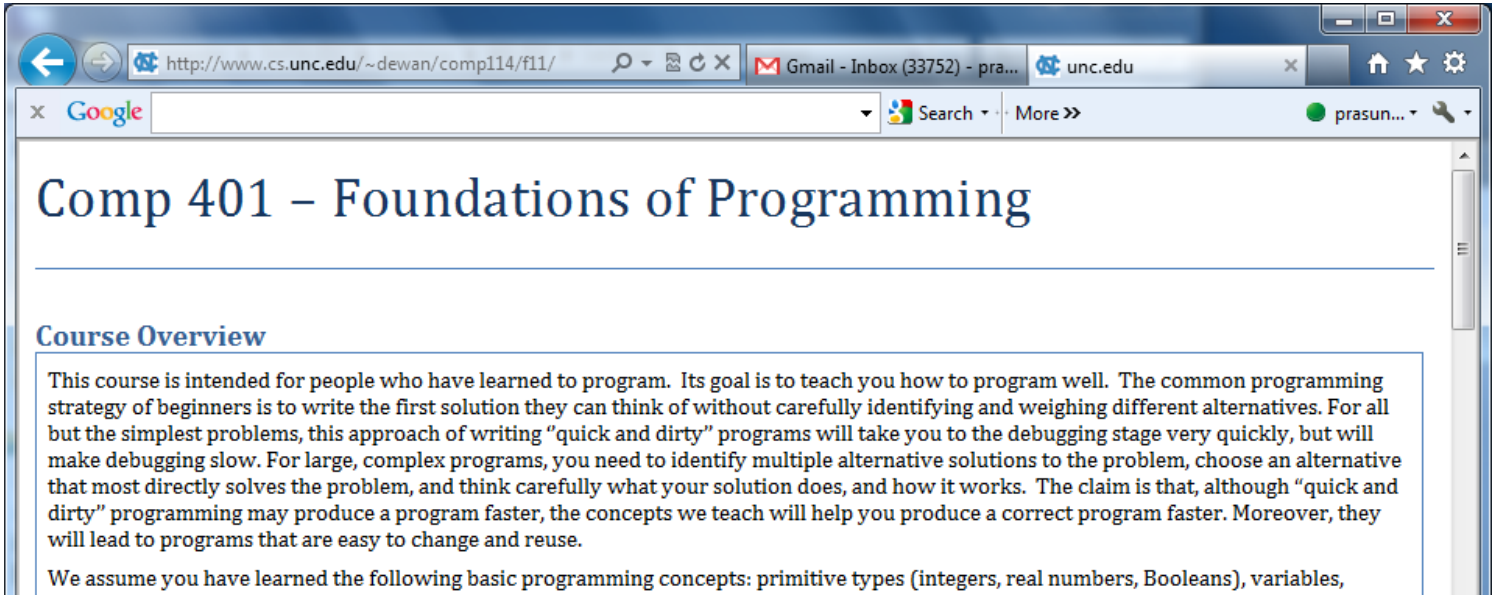
Course page:
http://www.cs.unc.edu/~dewan/comp401/current/

Not in Course?

About 20% of class drops in 2 weeks

Put your name, PID and email in one of the passed sheets (even if you are in the course)

Course Page



Linked from my home page (google my name to find it) and UNC course pages

Detailed Overview: Course Syllabus

Course Overview

This course is intended for people who have learned to program. Its goal is to teach you how to program well. The common programming strategy of beginners is to write the first solution they can think of without carefully identifying and weighing different alternatives. For all but the simplest problems, this approach of writing quick and dirty programs will take you to the debugging stage very quickly, but will make debugging slow. For large, complex programs, you need to identify multiple alternative solutions to the problem, choose an alternative that most directly solves the problem, and think carefully what your solution does, and how it works. The claim is that, although quick and dirty programming may produce a program faster, the concepts we teach will help you produce a correct program faster. Moreover, they will lead to programs that are easy to change and reuse.

We assume you have learned the following basic programming concepts: primitive types (integers, real numbers, Booleans), variables, constants, assignments, comments, expressions, arrays, loops, arrays, and procedures/functions/methods. These concepts are taught in most if not all introductory programming courses regardless of whether they teach conventional or object-oriented programming. This course will teach you the next-level programming concepts. These include objects, classes, interfaces, packages, inheritance, delegation, design patterns, exceptions, assertions, pointers, and formal correctness. These concepts will not help you solve new problems; rather, they will help you solve problems in new ways. The skills that will enable you to use these concepts will form a large part of the challenge you face in this course. After this course, you will have a much deeper understanding of the programming and learn some of the ideas that can make programming a science. We will be using Java as a vehicle for learning these concepts.

Course Syllabus in UNC Format

Fall 2012 Comp 401 Offering

Fall 2007 Comp 110 (Using Object Editor) Offering by Sasa Junuzovic

Fall 2011 Comp 110 (with ObjectEditor and Eclipse Helper) Offering by Jason Carter

Class Material

Audience	Unit (LAST YEAR Start Date)	Recorded PPTX Slides	PDF Slides	Chapters (Not always current with PPT)	Videos	LAST YEAR Assignments	Source Code of Examples (Java Package)
401	Course Information (8/24	PowerPoint	PDF	Course Overview			
110 and 401	Eclipse Install and Basic Use (Look on your own)	PowerPoint	PDF	Warm-up Chapter			
110 and 401	ObjectEditor	PowerPoint	PDF	Warm-up			
401							
401							
401	programming in Java for those who know conventional programming			Chapter			kage

No book: a Java reference might help but is not necessary

Each unit has PPT slides (with recordings), Word document, and assignment

Both background (110) and class material

Source Code Linked to Class Material

Object-first Introduction to Programming	PowerPoint	PDF	Objects Chapter			lectures.objects Package

Package lectures.objects
ABMICalculator
ABMICalculator.1
ASquareCalculator
FunctionsDriver
SquareCalculatorDriver

```
package lectures.objects;
import util.annotations.WebDocuments;

@WebDocuments({"Lectures/Objects.pptx", "Lectures/Objects.pdf", "Videos/Objects.avi"})
public class ABMICalculator  {
    public double calculateBMI(double height, double weight) {
        return weight/(height*height);
    }


}
```

Comp 110/401
Objects

Instructor: Prasun Dewan

Special Source Code Browser

Source Code of Class Examples

Zipped Directory
JavaToHTML

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

All Classes
ABMIAndOverweightSpreadsheet
ABMICalculator
ABMICalculator.1
ABMICalculatorWithErrors
ABMISpreadheetWithPublicVariables

ABMICalculator.1.java

```
package lectures.objects;
import util.annotations.WebDocuments;
@WebDocuments({"Lectures/Objects.pptx", "Lectures/Objects.pdf", "Videos/Ob
public class ABMICalculator  {
    // weight is in Kgs, height in metres
    public double calculateBMI(double height, double weight) {
        return weight/(height*height);
    }
}
```

Linked Assignments

401	Scanning (8/24)	PowerPoint	PDF	Warm-up Chapter		Number List Processor Number and Command Scanner	lectures.scanning Package

First assignment is due next week, so start on it ASAP (on the web site)

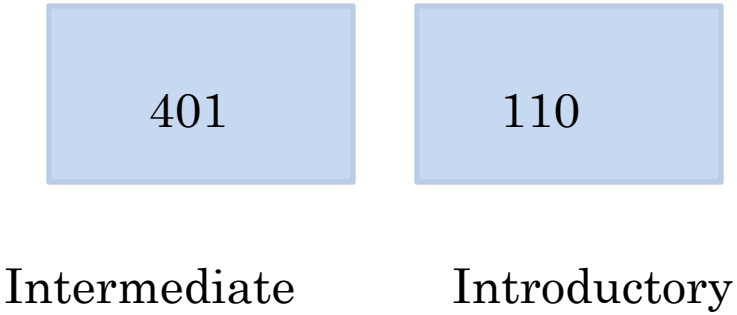
Will help you decide if you belong in this class

Comp 401 vs. 110

401	110
CS Majors	Psychology, Biology, …
Object-Oriented	Functional, Imperative, …
Java	C++, Python, …

- Majors vs. Non Majors?
  - Majors usually start with 401
  - But many 110 students become majors.
- Object-oriented vs. Conventional?
  - Both 401 and (current) 110 focus on objects.
- Java vs. Non-Java?
  - 110 and 401 are both in Java
  - Language is not the issue
  - Expected to use only those Java features taught in class
  - Course is not about Java

Comp 401 vs. 110

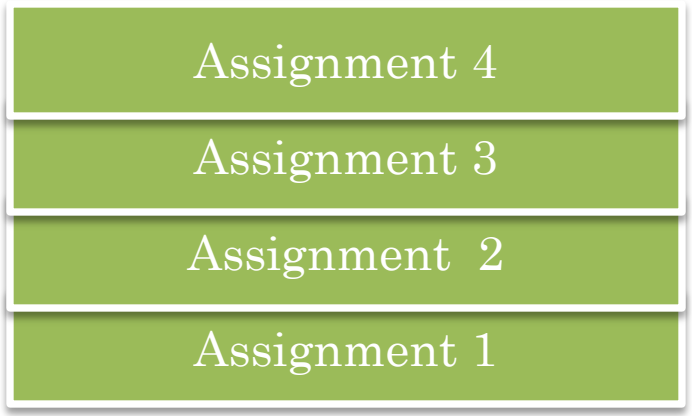


- "Intermediate" vs. "introductory" programming
  - Introductory may be object-oriented
  - Introductory may be conventional
  - *Assume background in conventional programming and will teach Java syntax for it.*
  - Repetition for those who know object-oriented programming.

Introductory Conventional Programming

- Types, variables, assignment , constants, expression
- Conditionals, loops.
- Input and output
- Arrays/Strings
- Procedures/Functions/Subroutines/Methods
- Comments
- Program vs. algorithm

Layered Assignments = Project



Assignments will build on each other to create a semester project

Due dates normally separated by a week (holidays, exams can cause more separation)

Working vs. Almost Working

- 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

Grade Distribution

Exams (Two midterms, no final)	44%
Assignments (Home work)	46%
Recitation (Class work)	10%
Fudge Factor (Class participation, other factors)	10%

No final!

Two Submission Dates

Comp 401 - Assignment 1: Writing a Number Scanner

Date Assigned: Tue Aug 20, 2013

Completion Date: Fri Aug 30, 2013 (11:59 pm)

Early Submission Date: Wed Aug 28, 2013 (11:59 pm)

In this assignment, you will revise your programming skills by doing an assignment involving the use of many of the concepts that are a pre-requisite for this course, which

Extra credit if submitted early on a Wednesday

This program reads input in a loop until the user enters a special "end of input" line.

Normal submission date is a Friday

Extra Credit Programming

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.

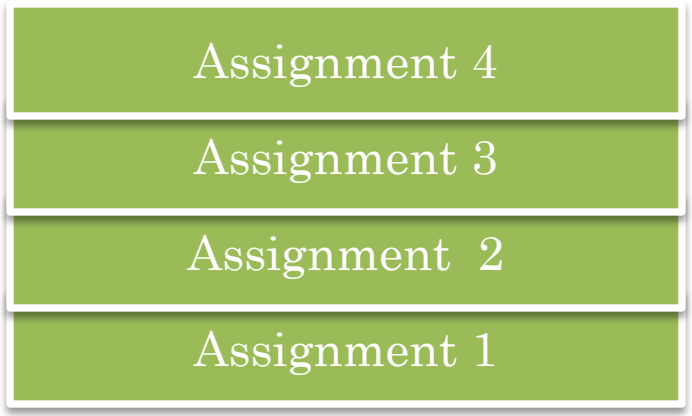
Students have varying interests and abilities

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

Layered Assignments = Project



Assignment 4 Due Date

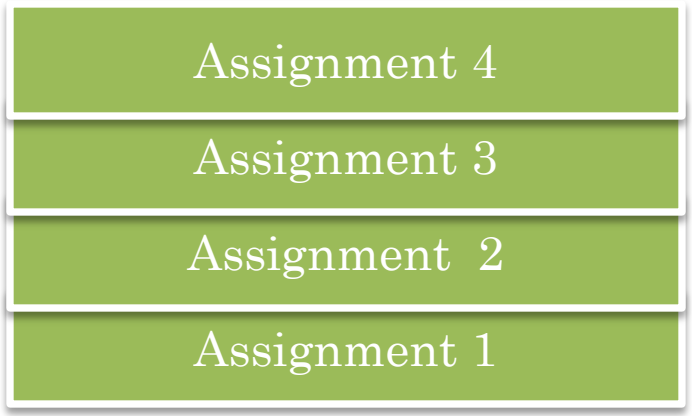
Assignment 3 Due Date

Assignment 2 Due Date

Assignment 1 Due Date

Can submit up to two class days (1 week) late with penalty

Shifting Dates



Assignment 4 Due Date

Assignment 3 Due Date

Assignment 2 Due Date

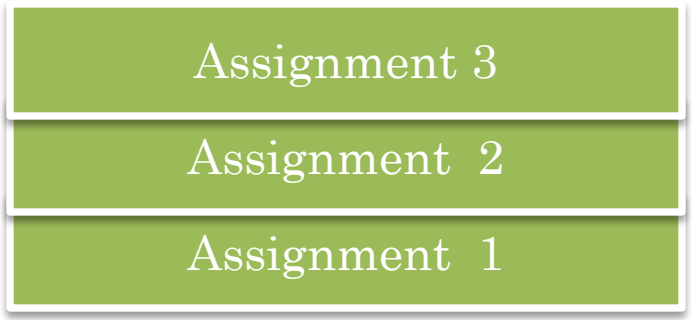
Assignment 1 Due Date

What if you get permanently behind?

Can shift assignment dates N times if last N assignments will not be done.

But you sacrifice the last N assignments, whose scores will go in fudge factor.

Skipping Once



Assignment 4 Due Date

Assignment 3 Due Date

Assignment 2 Due Date

Assignment 1 Due Date

Skipping Twice

Assignment 4 Due Date

Assignment 3 Due Date

Assignment 2 Due Date

Assignment 1 Due Date

Do submit sacrifice assignments if you catch up – will take this work into account in fudge factor

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

2. You should store the numbers you scan into an array for later use. This

Goal is not to teach Java and its libraries

should not write a monolithic main method. This will demonstrate you ability to

It is to teach you how to build these libraries

Usually Java features not covered in class will be banned

Require use of certain programming techniques

Correctness is only one of the goals

Program must also be efficient and well crafted

Getting Help

- Can discuss solutions with each other at a high level
- Not at the code level
- Sharing of code is honor code violation
- Can help each other with debugging as long as it does not lead to code sharing
- Assignments may contain solution in English (read only if stuck)

Piazza for Getting Help and Class Discussion

Getting Help and Class Discussion

We will be using Piazza for class discussion and getting help. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you do not get a response within a day or two on Piazza, please send mail to help401@cs.unc.edu. But try Piazza first. Do not send mail to an individual instructor, as that can overwhelm him - such mail will be ignored.

Before posing a question, please check if this question has been asked before. This will reduce post clutter and reduce our burden. Repeat questions will be ignored by the instructors.

Piazza allows anyone to respond. So if you see a question that you think you can respond to, please do so, as that will reduce our burden and help you "teach" your fellow students.

This will be a form of class participation that will be noted when I allocate my fudge points!

Hope it works well

If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: https://piazza.com/unc/fall2013/comp401

Dos and Don'ts

Do	Don't
Attend class even though material is online	Look at your laptop while in class unless it is to take notes
Look at class material if you have to miss class for extenuating circumstances or did not understand	Come to class late **or leave class early**
Use Piazza for questions	Send mail to individual instructors
Use help401@cs.unc.edu for grades and other private queries