Instructor

Dr. Markus Lumpe
113 Atanasoff Hall
294-2410
lumpe@cs.iastate.edu

TA's

Oksana Yakhnenko

Meeting Place and Times

MWF 4, Science 152
R 4 (Sec A), Sweeney 1126
R 3 (Sec B), Sweeney 1126

Course Description

The programming tools and techniques used in industry are constantly evolving. Com S 430, as a senior elective course, is intended to bring the student to the state-of-the-art cutting edge of what is happening at or near the time of his/her graduation. This course will expose students to the latest and most powerful techniques that have or are likely to become industry standards, but are not covered in other courses in a computer science curriculum. The techniques listed in the catalog description represent topics to be covered in the first offering in fall 1997. The topics are expected to naturally evolve with changing software/hardware technologies.

The catalog description of the course is as follows:

Topics in advanced programming tools and techniques widely used by industry (e.g., event-driven programming and graphical user interfaces, standard libraries, client/server architectures and techniques for distributed applications). Emphasis on programming projects in a modern integrated development environment. Oral and written reports. Nonmajor graduate credit.
Textbooks

In the last two years several .NET related books have been published. However, none of those books covers the .NET framework and the C# language completely. Therefore, I have decided to make only recommendations. However, I will use Andrew Troelsen’s book most of the time. Additionally, the midterm will be an open book test. So, you should consider getting one of the books below. If you have any questions concerning the textbooks (contents, should I buy it), please do not hesitate to contact me.

- Andrew Troelsen, *C# and the .NET Platform, Second Edition*
- Jesse Liberty, *Programming C#, Third Edition*
- Ingo Rammer, *Advanced .NET Remoting*
- Ian Griffiths and Matthew Adams, *.NET Windows Forms in a Nutshell*
- Peter Drayton, Ben Albahari, and Ted Neward, *C# in a Nutshell*
- Erik Brown, *Windows Forms Programming with C#*
- Jeffrey Richter, *Applied Microsoft .NET Framework Programming*
- Tom Archer, *Inside C#*
- David Conger, *Remoting with C# and .NET*
- Neil Bradley, *The XSL Companion*
- Michael Kay, *XSLT, 2nd Edition*
- Fergal Grimes, *Microsoft .NET for Programmers*
- H. M. Deitel, P. J. Deitel, and S. E. Santry, *Advanced Java 2 Platform - How to Program*
- Cay S. Hortmann and Gary Cornell, *Core Java 2, Volume I - Fundamentals*
- Cay S. Hortmann and Gary Cornell, *Core Java 2, Volume II - Advanced Features*

Prerequisites

The formal prerequisite in the Iowa State catalog is successful completion of Com S 311, 361, Engl 105, Sp Cm 212.

The skills relevant to Com S 430 include the ability to:

- Design and implement data structures and algorithms
- Use a make facility
- Collect requirements for an application, specify a design, plan implementation and test, document all development phases
- Understanding of the principle structure of programming languages

If you do not have this background, especially if you are interested in research in programming languages, you should take Com S 311, 331, 361. Mere reading of texts on these subjects is not enough.
Computer Accounts

We will be using the computing facilities (Windows workstations) of the Department of Computer Science for all course-related assignments. The reference system for all assignments is Visual Studio .NET 2003.

If you do not already have a login on the departmental computer systems, you should get one (for details, please consult the department’s web page http://www.cs.iastate.edu).

The Computation Center and the Computer Science Department hold tutorials that are designed to help a new user to get familiar with their facilities. Please contact them for a schedule and information on signing up for one of these tutorials.

In order to debug applications within Visual Studio .NET your account need to be a member of the Windows “Debugger Users” group. We will arrange this for you for the fall semester. Furthermore, debugging from a networked file system may not be possible due to some security constraints. If you want your application, please copy the project to a local directory. Do not forget to copy it back, at the end of your session!

The course web page located at http://www.cs.iastate.edu/~cs430/ will be used to post course materials including assignments, solutions to problem sets, lecture notes, etc.

As a user of the ISU computer facilities you have to familiarize yourself with the policies that apply (see http://www.cs.iastate.edu/documents/policies.html).

Objective

At the end of this course you will be able to:

- Use C# features like classes, interfaces, delegates, inheritance, virtual functions in a modern software development environment.
- Develop applications using a modern, integrated software development environment (Visual Studio .NET).
- Effectively use the class libraries of .NET.
- Develop event-driven, GUI-based applications.
- Develop distributed applications using the remoting features of .NET.
- Use a modern IDE to develop GUI’s.
- Use XML to exchange data between applications.
- Use the .NET multithreading abstractions to develop concurrent applications.
- Develop a multi-tier application.
- Develop Web-Services.
Assignments, Examinations, and Grading

There will be regularly scheduled problem sets and programming assignments to help you learn the material and to allow us to evaluate your progress.

Problem sets will be handed out roughly every week or two. Most assignments will require laboratory work. You should expect to work on a problem set between two and four hours. If you have trouble finding a solution, ask for help! All assignments are fair and reasonable. No problem set will require more than four hours. The necessary software packages will be provided on the department’s Windows workstations.

Absolutely no credit for late homework will be given during the last week of classes (or later!), or for homework turned in later the next lecture meeting after the due date. If you are consistently late with homework, we may stop accepting your late homework. All problem sets are posted on the course web page. So, even if you have not received a homework handout, your can download the problem set from the course web page.

Homework problems that are late receive points based on the following table:

<table>
<thead>
<tr>
<th>When turned in</th>
<th>Percentage penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>If answer(s) are given in class</td>
<td>25% (or more due to time)</td>
</tr>
<tr>
<td>By 5pm of the next weekday</td>
<td>15%</td>
</tr>
<tr>
<td>At the next lecture meeting</td>
<td>25%</td>
</tr>
<tr>
<td>Later or during last week of classes</td>
<td>100%</td>
</tr>
</tbody>
</table>

All problem sets are equally weighted. In general every problem set is worth a different number of points. The maximum number of points equals 100%.

You receive one grade for all problem sets, which calculated as follows:

\[
\text{Homework grade} = \frac{\text{sum of all homework's}}{\text{number of homework's}}
\]

Example:
- Problem set 1: 50 out of 75 = 67%
- Problem set 2: 67 out of 80 = 84%
- Problem set 3: 89 out of 90 = 99%

Grade: 67% + 84% + 99% / 3 = 83%

Occasionally, you may earn some extra credits for a problem set, if this problem set is particular challenging. Therefore, the final homework grade may be greater than 100%. Extra credit can help you to improve your final grade.

Homework problems are due at the date and time specified on the handouts. In general, problem sets are in general due one just before the class starts. If you have problems with a particular assignment, talk to the instructor before the deadline.
You have to submit your solutions on paper in class (preferable printed using Visual Studio .NET). You also need to submit your solution electronically. You need to logon to a CS department UNIX workstation and use the course's “turnin” script.

The solution of a problem set will be discussed in the recitation following the due date.

There will be a midterm to test your theoretical knowledge. In the second half of the semester you will work on a group project (a multi-tier Internet game). Every student has to turn in a written report that documents all phases of the project. In order to pass this course you must have turned in 80% of the problem sets, you must have taken the midterm, turned in a complete project, and a written report.

The final grade is calculated as follows:

<table>
<thead>
<tr>
<th></th>
<th>Percentage Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final grade:</td>
<td>20% homework grade</td>
</tr>
<tr>
<td></td>
<td>20% midterm</td>
</tr>
<tr>
<td></td>
<td>30% project</td>
</tr>
<tr>
<td></td>
<td>30% written report</td>
</tr>
</tbody>
</table>

Your grade is independent of anyone else's grade in this class. That is, we do not grade on a curve, and everyone can get an A. Our purpose in grading is to uphold a standard of quality and to give you feedback: it is not to rank students.

Instead of using curve grading as a final defense against problems that are too hard, we use the following policy. If a problem on the homework or a test is so hard that most students do not "get it", then we will eliminate it from the test or homework grading. If this problem was appropriate, then we will teach how to solve problems like it, and give a similar problem on another test or homework. If it was not appropriate, then we will ignore it. If you detect such a problem, let us know about it as soon as possible, as it will save us all a lot of work.

Although we will not always make fine distinctions in points the nominal minimum standards are given by the following table.

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>A</td>
</tr>
<tr>
<td>85%</td>
<td>A-</td>
</tr>
<tr>
<td>80%</td>
<td>B+</td>
</tr>
<tr>
<td>75%</td>
<td>B</td>
</tr>
<tr>
<td>70%</td>
<td>B-</td>
</tr>
<tr>
<td>65%</td>
<td>C+</td>
</tr>
<tr>
<td>60%</td>
<td>C</td>
</tr>
<tr>
<td>55%</td>
<td>C-</td>
</tr>
<tr>
<td>50%</td>
<td>D+</td>
</tr>
<tr>
<td>45%</td>
<td>D</td>
</tr>
<tr>
<td>40%</td>
<td>D-</td>
</tr>
<tr>
<td>Less</td>
<td>F</td>
</tr>
</tbody>
</table>
Cheating

The simple rule of thumb is:

**Never give or use someone else’s code or written answers.**

Such exchanges are definitely cheating and not cooperation. We will take action if we catch you cheating on a test or exchanging code or written answers. Read the section on *Academic Dishonesty* in the section on *Academic Regulations* in the *Iowa State University Bulletin*.

If you honestly believe that certain problems are too much busy work, then bring it to the instructor's attention; or failing that, only do the part of the problem that you think you need to do to learn the material and explain that to us.

Special Accommodation

If you have a documented disability and anticipate needing accommodations in this course, please make arrangements to meet with me soon. Please request that a Disability Resources staff sends a SAAR (Student Academic Accommodation Request) form verifying your disability and specifying the accommodation you will need.

Active Duty

Students who are in ROTC/military may be called on Active Duty and/or their two-week annual training. For active duty, ISU has a policy, which is available at:

[http://www.iastate.edu/~registrar/info/act-duty.html](http://www.iastate.edu/~registrar/info/act-duty.html)

Grades, incompletes, or administrative drops all require approval of the instructor.

The two-week annual training is not covered by the procedures outline above. All requests have to be made in writing to the instructor.

Syllabus

The following gives a tentative list of topics not necessarily in the order in which they will be covered in the course. Specific reading assignments will be announced in class. Brief lecture outlines will be placed on the course homepage periodically.

- What is .NET?
- Introduction to C#
- Understanding types and assemblies
- A multi-tier application
- XML, XSL, XSLT (optional)
- ADO.NET (optional)
- Petri Nets
- .NET Remoting
- XML Web services
- Windows Forms user interfaces
- Web Forms
- Mobile applications