CGN 6426
Advanced Computing in Civil & Environmental Engineering

Course Syllabus
Fall 2000

General Information

Instructor Prof. Fang Zhao
Office EAS 3607
Classes Mondays & Wednesdays 5:40 - 6:55 p.m.
Classroom EAS 2830 (EIC lab)
Hours Tuesdays 4:30 - 6:00 p.m.
Thursdays. 4:30 - 5:30 p.m.
Friday 2 am - 5 p.m., other hours by appointment
Phone 348-3821
Fax 348-2802
Email fang@eng.fiu.edu (preferred mode of communication)

Catalog Information

Advanced computer modeling and programming techniques for civil and environmental engineering applications including data modeling, engineering database design, object-oriented programming, and user interface design.

Course Objectives

The objectives of this course are to introduce a disciplined approach to problem-solving methods and algorithm development, with a focus on more complex data structures, database applications, user interface design, and the integration of applications code into an operating system environment such as Windows 98/NT; to teach program design, coding, debugging, testing, and documentation using good programming style; and to teach problem solving using computer programming tools in various civil and environmental engineering disciplines.

Class Policy

Class Participation

Students are expected to participate in the lectures, and are responsible for submitting assignments and projects on time. If for a legitimate reason (sickness and emergencies, but not vacations) a student cannot attend a class, or submit an assignment on time, the instructor must be consulted in advance or as soon as possible and an arrangement made. Students will sign an attendance sheet at the beginning of each
class and class attendance will be considered when determining the final grades.

**Homework**

While a certain amount of collaboration is expected and often necessary, there is a fine but clear line between collaboration and cheating. Students may help each understand new concepts and techniques, discuss design and programming issues, and exchange information on bugs in programs. This does NOT include duplication of programs or designs. All programming assignments are individual assignments. You must design, code, debug, and execute the programs on your own. Copying of code, regardless of the number of editor changes, renaming, and/or retyping is considered cheating, and a student's inability to describe the function of an assignment will be considered clear evidence of cheating. Providing another student with part or all of a solution is also certainly cheating. Cheating will result in failing the course, along with possible expulsion from the University. When an assignment is designed as a team effort, individual efforts must be clearly defined and reported.

All programming assignments must conform to the variable and object naming standards and all other coding and internal (and external) documentation standards established in this class. In general programs will be graded based on meeting assignment specifications, elegance and efficiency of program design (not at the expenses of clarity and generality), use of standard interface design, modular code, code readability, and internal documentation. A very detailed structure chart is due with each assignment.

All assignments must be handed in at the beginning of a lecture on the date they are due. No late homework will be accepted unless an extension has been granted after consulting with the instructor prior to the homework deadline.

Students are responsible for keeping multiple copies of assignments on different disks and backing up files as necessary. Failure to do so thus resulting in loss of a completed assignment (it does happen, often) will not be considered as a legitimate reason for extension of homework submission deadline.

**Computer Account and Network Access**

Electronic mail will be used as a main means for communication between instructor and students. All students should obtain an Engineering Information Center account or other Internet service for email and some of the homework assignments. EIC account applications should be submitted to the EIC.

**Grading**

- **Homework Assignment** 25%
- **Quizzes ( ~ 4)** 25%
- **Project** 50%
  - Design document 10%
  - Implementation 30%
  - Final report & documentation 10%
Final scores are computed based on cumulative scores weighed by percentages above, and a letter grade will be assigned as follows:

- A = 93-100,
- A- = 90-92,
- B+ = 89-87,
- B = 86-83,
- B- = 80-82
- C+ = 79-77,
- C = 76-73,
- C- = 70-72
- D+ = 69-67,
- D = 66-60,
- and F below 60

Final grades cannot be changed once assigned. Any discussion about a grade for a quiz, an assignment, the project, or the final grade must take place within five working days immediately after the grade is assigned. To be fair to all students, no additional work will be given to an individual student for the purpose of improving his/her grades.

**Course Project**

To allow the practice of good software engineering and to gain software design, implementation, and documentation experience, a course project must be completed. Developing ideas for the project is the responsibility of the students, but must be approved by the instructor before proceeding with the design and implementation of the project. The project should address a specific engineering or scientific problem, and should utilize the software tools learned in this class to the maximum extent. As a minimum, the program should have the following elements:

- A window-based graphic user interface
- Data management capabilities (file management, data manipulation, display)
- Graphic display (charts, curves, objects models, etc.)
- Interface to at least one other Windows-based application (e.g. Excel, Access, Word, S-Plus)
- Create installation package for the Visual Basic application and install it to end-users’ computers

The project is staged as described below.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal</td>
<td>Describe the idea and methodologies</td>
<td>9/18/2000</td>
</tr>
<tr>
<td>Design document</td>
<td>Formal documentation of the software design</td>
<td>10/23/2000</td>
</tr>
<tr>
<td>Implementation</td>
<td>Writing and debugging code</td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td>Testing the program and document the testing</td>
<td></td>
</tr>
<tr>
<td>Final report</td>
<td>Assembly of all the above including the source code</td>
<td>12/13/2000</td>
</tr>
<tr>
<td>Life demonstration</td>
<td>Life demonstration of the software with a brief introduction</td>
<td>12/13/2000</td>
</tr>
</tbody>
</table>

Crashes of program will result in loss of points.

**Objectives and Assessment Strategies**

1. Students will understand problem solving methods and algorithm development, and be able to apply that understanding in writing programs that correctly use programming language features, specifically including selection and repetition control structures, array data structures, and pass-by-reference and
pass-by-value parameters. This objective will be assessed via homework assignments, quizzes, and project. Assignments involving writing programs will be particular relevant.

2. Students will understand how data is represented in a computer. This objective will be assessed via quizzes and project.

3. Students will learn to use the Visual Basic 6 Integrated Development Environment to create, debug, compile and run programs. This objective will be assessed via the use of the IDE during quizzes.

4. Students will understand the concept of variable scope. This objective will be assessed via programming assignments, quizzes, and project.

5. Students will be able to format output appropriately. This objective will be assessed via programming assignments and project.

**Course Outline**

<table>
<thead>
<tr>
<th>Topics</th>
<th>Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>Chapter 1</td>
</tr>
<tr>
<td>2. IDE</td>
<td>Chapter 2</td>
</tr>
<tr>
<td>3. Review of basic programming concepts</td>
<td>Chapter 3</td>
</tr>
<tr>
<td>4. Control structures</td>
<td>Chapters 4, 5</td>
</tr>
<tr>
<td>5. Sub and Function Procedures</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>6. Arrays</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>7. String manipulation, dates and times</td>
<td>Chapter 8</td>
</tr>
<tr>
<td>8. Graphic User Interface (GUI)</td>
<td>Chapters 9, 10</td>
</tr>
<tr>
<td>9. Error handling and debugging</td>
<td>Chapter 13</td>
</tr>
<tr>
<td>10. Input and output</td>
<td>Chapters 12, 14, 15</td>
</tr>
<tr>
<td>11. Object-Oriented programming</td>
<td>Chapter 16</td>
</tr>
<tr>
<td>12. ActiveX</td>
<td>Chapter 17</td>
</tr>
<tr>
<td>13. Databases and Data Management</td>
<td>Chapter 18</td>
</tr>
<tr>
<td>14. Data structures, collections, and dictionary</td>
<td>Chapter 21</td>
</tr>
</tbody>
</table>

**EIC (Engineering Information Center) Lab Information**

Phone: (305) 348-3034  
Fax: (305) 348-4023, (305) 348-6178  
Lab Hours (Fall 2000): Monday through Friday -- 8:30 am to 10:00 pm  
Saturday -- 9:00 am to 5:00 pm

**Internet Resources**

http://www.deitel.com/home.htm  
http://www.fiu.edu/~zhaof (for course-related information)  
http://msdn.microsoft.com/vba/  
http://msdn.microsoft.com/vbasic/  
http://visualbasic.about.com/
Note: this syllabus is subject to minor modifications.