Computing IV - 91.204

- Instructor: John Sieg
- Course website: www.cs.uml.edu/~john/91.204/
  (includes links to anonymous suggestion box, discussion board, and “grades so far”)
- Prerequisite: Computing III
- Please e-mail me “Add to 91.204 distribution list” message.

Prof. John Sieg

- Research in query optimization and scheduling transactions
- Office: Olsen 211
- E-mail: john@cs.uml.edu
- Home page (including office hours, etc.): www.cs.uml.edu/~john

Course Content (Approximate)

- Event-based programming, application frameworks (Visual Studio .NET / C++)
- Relational databases (when to use, SQL)
- The ER model and data design
- Object-oriented software development
- Lexical scanning, regular languages, automata
- Context-free languages and pushdown automata
- The Internet and the Web
- XML

91.204 - Grading

- Assignments 30%
- Project 20%
- Midterm Quiz 25%
- Final Quiz 25%

Quizzes

- midterm and final (non-cumulative with possible exceptions)
- closed book
- Except for documented emergencies, quizzes must be taken on (before) quiz date.
  Please see syllabus for complete policy.
- Samples at course website

Homework

- Web page linked to www.cs.uml.edu/~john/91.204/
- Strict lateness policy: Submit before I start lecturing!
- Teaching assistant is in charge of all aspects of homework.
- First assignment: Read Sections 1.1-1.9 for Monday.
Visual Studio: Why Bother?

- Support for event-based programming
- Reusable components (supporting design-time interfaces to customize functionality)
- Visual Studio = An application framework, leveraging the power of class inheritance and components to enable reuse

Event-Based Programming

- When was the last time you heard of a shrink-wrap program that grabbed its input from standard input and wrote its output to standard output?
- Instead, real programs react to events, e.g., mouse clicks on buttons or menu items (user events) or system shut-down (system events)

The Events Loop

do while (TRUE)
{  
get an event for this program;
handle the event with a designated handler;
}

Responsibilities

- Developer: Build event handlers (functions) and bind them to appropriate events (e.g., Click) on appropriate resources (e.g., a particular button)
- Run-time environment (e.g., MFC, .NET, Java Swing): Implement events loop and find appropriate handlers.

Evolution of Windows Programming

- C code against Windows API (= torture)
- MFC programming using Visual Basic, Visual C++, etc. (= painful)
- .NET Programming (= almost fun)

What’s New in .NET

- Simplified interface to forms, controls, etc., similar to Visual Basic and Java
- Interface is shared by different languages (but not different OSs, unlike Java)
- Same interface for server code, implying unified model for Web services
- Security model inherited from Java
- Simplified events model similar to that of Java