Computer Science Department
cs.salemstate.edu

**CSC 325 Advanced Programming Techniques** **4 cr.**

**Instructor:** TBA **Office:** location **Phone:** (978) 542-extension
**email**: TBA@salemstate.edu **Office Hours:** days and times

| Section | Time | Room | Final Exam |
|---|---|---|---|
| nn | days and times | location | date and time |
| Lnn | days and times | location | |

**Catalog description**:

Presents state-of-the-practice software development techniques such as Web based computing, application data exchange, frameworks for managing and securing robust systems, developing multi-tier software systems. Topics will be illustrated by applications to such areas as remote objects, communication with remote components, multithreading, reflection, security and Web Services. Specific topics will be chosen based on current software industry trends. Fundamental programming language paradigms, type systems, and memory allocation and management strategies are presented and discussed, followed by comparative analysis of the language utilized in this course and its prerequisite. Three lecture hours and three hours of scheduled laboratory per week, plus extensive programming work outside of class.

**Prerequisite**: CSC 260.

**Goals:**

CG01: to enhance students' skills in problem analysis and program design and implementation through the presentation of complex applications;
CG02: to present a unified discussion of the important ideas and techniques involved in the use of modern programming systems and tools in software design;
CG03: to guide the student through one or more large-scale projects employing these tools;
CG04: to provide additional software development experience, normally using a programming language or environment other than one that the students have already studied.
CG05: to provide a basic level of understanding of fundamental programming language concepts

**Objectives:**

Upon completion of this course the student will have demonstrated the ability to:

CO01: select or design appropriate algorithms, code libraries, and language features for the solution of a complex problem, and use these ingredients effectively to obtain a solution to the problem;
CO02: employ a clear, consistent, readable style in the implementation of the problem solution;
CO03: produce clear documentation for the problem and its solution.
CO04: understand and employ the various types of code reuse in object-oriented design;
CO05: understand and use a variety of components from popular standard software libraries;
CO06: understand and carry out software development related to the specific topics presented in the course.
CO07: design a suite of test data for a particular problem and perform thorough testing and debugging of the generated code.
CO08: demonstrate knowledge of fundamental programming language paradigms, type systems, and memory allocation and management strategies

**Program Outcome vs. Course Objectives matrix**

| Program Objective (condensed form) | CO01 | CO02 | CO03 | CO04 | CO05 | CO06 | CO07 | CO08 |
|---|---|---|---|---|---|---|---|---|
| **PO-A:** apply knowledge of computing and math | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| **PO-B:** analyze a problem and define its computing requirements | ✓ | | | | ✓ | ✓ | ✓ | |
| **PO-C:** design, implement and evaluate applications | ✓ | | | | ✓ | ✓ | ✓ | |

| Program Objective (condensed form) | CO01 | CO02 | CO03 | CO04 | CO05 | CO06 | CO07 | CO08 |
|---|---|---|---|---|---|---|---|---|
| **PO-D:** function effectively in teams to accomplish a common goal | | ✓ | ✓ | ✓ | | | | |
| **PO-E:** professional, ethical, and social responsibilities | | | | | | | ✓ | |
| **PO-F:** communicate effectively with a range of audiences | | ✓ | ✓ | | | | | |
| **PO-G:** local and global impact of computing on people and society | | | | | | | | |
| **PO-H:** need for continuing professional development | | | | | | | | |
| **PO-I:** use current techniques, skills, and tools | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| **PO-J:** apply theory and principles to model and design systems | | | | | | | | ✓ |
| **PO-K:** apply design and development principles in constructing software | | | | | | | | ✓ |

note - full statements of the Program Outcomes (program objectives) for the Computer Science Major can be found in the document *Computer Science Major Program Educational Objectives and Program Outcomes* on the Assessment page of the Computer Science Major (cs.salemstate.edu)

**Topics will be selected from the following:**

- standards for algorithm design and programming style (review)
- overview of software engineering principles **SE1(1)**
- object-oriented design **PL6(1)**
- object oriented application frameworks **SE2(2)**, **PL2(1)**
- design patterns **PF2(2)**, **PL6(1)**
- profilers and other debugging aids **SE3(2)**
- computer simulation and modeling (general discussion & examples) **IM3(1)** CN3(1) (not core)
- numeric applications **PF2(1)** CN1(2) (not core)
  - multiple-precision arithmetic
  - sparse matrices
- applications to memory management **AR4(2)** AR8(1) (not core)
  - garbage collection
  - reference counters
  - compaction
- concurrent programming and multithreaded applications **PL5(2)** SE9(1) (not core)
  - thread safety
  - thread synchronization
  - thread cooperation
- object serialization **PL5(1)**, **NC2(1)**
- object collections, their implementations and usage **SE2(2)**, **PF2(2)**
- in-code documentation design and usage **SE5(1)**
- design and deployment of Web applications **PF5(2)** AR9(1), NC5(2), HC5(1)
- object serialization and sockets programming **AR5(2)** AR9(1) (not core)
- remote object activation **NC2(2)** NC6(1) (not core)
- describing and discovering Web services communication protocols **NC4(2)** NC5(1) (not core)

- Programming language concepts (BoK TBD)
  - programming paradigms (two hours)
    - procedural
    - object-oriented
    - functional
    - declarative
    - multi-paradigm
    - emerging and/or specialized paradigms
  - type systems (one hour)
    - "strongly-typed" vs "weakly-typed" vs. "type safe"
    - static (compile-time) type checking
    - dynamic (rim-time) type checking
  - memory allocation and management (one hour)

- static vs. dynamic
- direct vs. indirect
- stack-based
- heap-based

**Programming Assignments:** The purpose of this course is to illustrate a variety of programming techniques in applications areas that do not have file structures and database techniques as their primary focus. Due attention will be paid to implementation details. There will be 4 to 6 programming assignments in which students will be asked to implement selected techniques.

All programs must conform to departmental guidelines for design and implementation, and laboratory reports must conform to the written guidelines supplied by the instructor. Regardless of numeric average or grades on individual assignments or examinations, a student will not be eligible for a passing grade in the course unless he or she has submitted a lab report for every assignment within the time frame specified by the instructor.

**Laboratory exercises:** There will be short programming and design exercises to be completed during weekly scheduled laboratory sessions. Laboratory exercises will concentrate on development tools and framework usage, object-oriented design implementation, and standard software library integration into the code.

The course grade will be determined using the following approximate weights: laboratory exercises - 20%, programming assignments – 40%, examinations (midterm and final) - 30%, written homework -10%.

**Course Objective / Assessment Mechanism matrix**

| | Written Homework | Laboratory Exercises | Programming Projects | Examinations |
|---|---|---|---|---|
| CO01 | | | ✓ | ✓ |
| CO02 | ✓ | ✓ | ✓ | ✓ |
| CO03 | ✓ | ✓ | ✓ | |
| CO04 | ✓ | | ✓ | ✓ |
| CO05 | | ✓ | ✓ | ✓ |
| CO06 | | ✓ | ✓ | ✓ |
| CO07 | | ✓ | ✓ | |
| CO08 | ✓ | | | ✓ |

**Bibliography:**


Dalessandro, Mayo, Joe. **Linq Programming.** McGraw-Hill, 2008.

Deitel, Harvey M.; Deitel. Paul J. **Visual C# 2010: How to Program. Fourth Edition.** Prentice Hall, 2010.

Evjen, Bill; Nagel, Christian; Glynn, Jay; Watson, Karli; Skinner, Morgan. **Professional C# 4.0 and .NET 4**. Wrox Press Inc., 2010.

Hejlsberg, Anders; Torgersen, Mads; Wiltamuth, Scott; Golde, Peter. **The C# Programming Language. Third Edition.** Addison-Wesley, 2009.

Hoffman K. **Microsoft Visual C# .NET 2005 Unleashed**, Sams, 2006.

Hoffman K. and Kruger L. **Microsoft Visual C# .NET Unleashed**, Sams, 2005.

Kanjilal, Joydip. **ASP.NET 4.0 Programming.** McGraw Hill, 2010.

Mayo, Joe. **C# 3.0 with the .NET Framework 3.5 Unleashed. Second Edition.** SAMS, 2008.

Mayo, Joe. **Microsoft Visual Studio 2010: A Beginner's Guide.** McGraw-Hill, 2010.

Michaelis, Mark. **Essential C# 4.0. Third Edition.** Addison-Wesley, 2010.

Millspaugh, Anita C.; Bradley, Julia Case. **Programming in Visual C# 2008.** Career Education, 2009.

Sanders, William. **ASP.NET 3.5: A Beginner's Guide.** Second Edition. McGraw-Hill, 2009.

Schildt, Herbert. **C# 4.0: The Complete Reference.** McGraw-Hill, 2010.

Watson, Ben. C# 4.0: **Real Solutions for C# 4.0 Programmers.** Sams, 2010.

**Academic Integrity Statement:**

"Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives." Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University website (http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity). The formal regulations are extensive and detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and will be dealt with according to the University's formal Procedures. *Copying without attribution is considered cheating in an academic environment - simply put,* ***do not do it!***

**University-Declared Critical Emergency Statement:**

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to www.salemstate.edu for further information and updates. The course attendance policy stays in effect until there is a university-declared critical emergency.

In the event of an emergency, please refer to the alternative educational plans for this course, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

**Equal Access Statement:**

"Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. **Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately.** Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services."

**Note:** This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via email.