



## **MET CS 503 Extended Syllabus Spring 2008**

### **Course Summary**

Microsoft .NET is a new technology that greatly simplifies application development both for traditional desktop, and for the emerging paradigm of Web-based services. .NET is a complete restructuring of Microsoft's whole operating system's infrastructure and represents a major stepping stone for programmers developing applications on Microsoft platforms. The new platform includes a new programming language C# (C-Sharp) and a major class library, the .NET Framework.

This course will discuss Microsoft's revolutionary C# language in contrast with C++ and Java, the .NET Framework class library and the Common Language Run-time (CLR) services. In-depth exploration of the various .NET services, technologies, pitfalls and best practices for development, debugging, and deployment.

In the first class, we'll examine many features of the C# language such as the multicast and the asynchronous nature of delegates, inheritance, memory management and finalization. As the course progresses, we'll examine, in context, the various and more advanced features of the C# language.

The next topic we'll be covering in detail is the VS.NET development environment, solutions, projects, namespace, dependencies, versioning, common assemblies, and the CLR. We'll dive into a detailed examination of the FCL core classes and components such as diagnostic, logging services, collections, and timers.

For the business tier we'll explore in detail the intricacies of isolation of this tier from the user interface and the data tiers. For the data access tier, we'll explore in detail File I/O using Stream and Binary Readers/Writers for sequential access, File Streams for random and more practical data access. We'll touch on XML Readers and Writers, then we'll dive into Serialization and the benefits it provides for data storage in binary, or SOAP(XML) formats into various storage media such as file system, registry, clipboard, and across the network. We'll examine versioning of Serializable objects for backward compatibility.

Then, we'll start by examining the rules of development of n-tier smart client applications. For the application thin client tier, we'll explore various types of application frameworks such as SDI, MDI, MTI, and service-like applications. In the user interface tier, we'll explore in detail .NET Forms, standard, user and custom controls, components, data binding, data exchange and validation, device I/O handling, visual inheritance, VS.NET extender providers.

We'll examine data services (ADO.NET: Access Data Objects) utilizing SQL Server and MS Access. We'll take a look at Data Readers, Data Adapters, and Datasets and how XML fits into the picture. We'll examine the benefits of using Data Readers versus Datasets, and the benefits of typed and un-typed Datasets.

We'll then examine the presentation layer again in the context of the GDI+ presentation services Graphical Device Interface. We'll examine 2D-vector graphics, imaging, text rendering, including the new features of transparency, gradients, anti-aliasing, double buffering techniques, zooming, off-screen image processing



and rendering.

We'll explore briefly the communications technologies within .NET utilizing the sockets and client communication classes for building unicast-based applications with UDP and TCP in addition to multicasting with UDP. We'll build an instant messenger application to demonstrate the client server versus the peer-to-peer communication model.

We'll then explore miscellaneous topics such drag and drop, performance counters, file system watchers, and much more.

Time permitting; we may examine some of the following bonus topics:

Multithreading: We'll discuss in great detail, the .NET threading models, thread management, thread priorities, thread states, thread local storage, thread pooling, updating UI from worker threads, and inter-thread synchronization. We'll examine the various synchronization mechanisms provided by .NET, examine the pitfalls of the various techniques and distill out some best practices of this complex area.

We'll examine also in some detail the Windows Communication Foundation and compare it with some of the predecessor technologies such as DCOM and CORBA. We'll examine Marshal-by-Value and Marshal-by-Reference techniques. We'll examine HTTP and TCP channels in the contexts of binary and SOAP formatters, hosting in IIS or in your own server, scalability, compatibility, interoperability, performance, fault tolerance, and Load balancing

We'll examine Windows mobile architecture, development techniques, tools and build components, services, and applications for Windows CE, Pocket PC, and Smart Phones and examine the world of occasionally-connected smart applications and form factor issues of those environments. We'll also highlight the major differences between .NET and .NET CF, the pitfalls, as well as memory and power management.

You do not need prior experience in C#, but you should have strong background/experience (See requirements) in an object-oriented language such as C++ or Java.

## Pre- Requisites

MET CS 341: Data Structures in C++ or  
MET CS 565: Java Programming or  
Equivalent

## Grading Policy

The grading criteria and ranges that this course undertakes is the following:  
Grades are assigned on the following discrete scale, not on an average and not on a curve.



100-94 A  
93-90 A-  
89-86 B+  
86-84 B  
83-80 B-  
79-76 C+  
76-74 C  
73-70 C-  
69-66 D+  
66-60 D  
59-56 D-  
55- 0 F

## Attendance (Required)

To master any challenges that windows programming present, you will need more than knowledge of how to do things but also a clear understanding of underlying implementation in the Windows operating system as well as what and why you have to do things one way versus another. Attending lectures is very crucial to accomplishing this level of understanding. In addition, students are encouraged to participate in the classroom by asking questions and making observations.

## Programming Assignments (60%)

Four programs will be assigned. Some assignments will be given a single week for completion, others will be given two weeks. Assignments should be handed in using the Online Student Drop box. Any other form of HW submittal (including emails of assignments) will not be accepted.

HW#1 15%  
HW#2 15%  
HW#3 15%  
HW#4 15%

Assignments will be graded based on 2 the following criteria:

Functionality/Requirements (The program runs and meets all requirements). Points will be deducted for bugs that are found based on the severity of the bug. Usually 2-3 points for minor bugs and 5+ points for major bugs. Missing functionality will be penalized accordingly.

At least 10 points will be taken off, if your project does not compile.

I will not grade (i.e. you will get a zero) if it was missing any files. Make sure to check and recheck your HW submittal. Assignment Solutions will be posted to the web site the night the assignment is due.

Therefore, No Late Assignments will be accepted. No exceptions.



### Final Project (40 %)

The final project is in reality a large programming assignment. You will have about 6 weeks to complete. I will be supplying the project topic during the course of the semester.

NO LATE PROJECTS WILL BE ACCEPTED.

No Incomplete Grades.

## Required Software

- Visual Studio.NET 2005 is required.
  - Any edition of VS.NET or VC# will suffice: Express, Team, Professional, or Standard Edition.
  - Visual Studio.NET 2003 will NOT be accepted.
  - The Express edition of Visual Studio 2005 is free and can be downloaded from Microsoft's web site: <http://msdn.microsoft.com/vstudio/express/visualcsharp/>
- SQL Server 2005 is required.
  - Any edition of SQL Server will suffice: Express, Developer, Enterprise, Standard, etc.
  - SQL Server 2000 or any other version will NOT be accepted.
  - The Express Edition of SQL Server 2005 is free and can be downloaded from Microsoft's web site:  
<http://msdn.microsoft.com/vstudio/express/sql/>
- A comparison of the various Visual Studio.NET editions can be found at the following link (If you care):  
<http://lab.msdn.microsoft.com/vs2005/productinfo/productline/>

## Required Book

Windows Forms 2.0 Programming, 2<sup>nd</sup> Edition

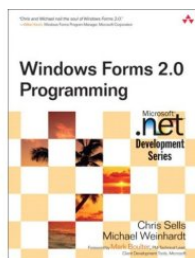
by [Chris Sells](#), [Michael Weinhardt](#)

**Paperback:** 1296 pages

**Publisher:** Addison-Wesley Professional; 2 edition (May 16, 2006)

**Language:** English

**ISBN:** 0321267966





## Recommended Books (Not required):

### Pro C# 2005 and the .NET 2.0 Platform, 3<sup>rd</sup> Edition

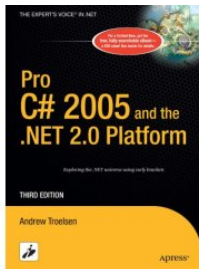
by [Andrew Troelsen](#)

**Hardcover:** 1032 pages

**Publisher:** Apress; 3rd edition (September 8, 2005)

**Language:** English

**ISBN:** 1590594193



### Visual C# 2005 Recipes: A Problem-Solution Approach

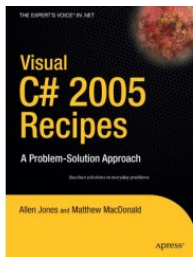
by [Allen Jones](#), [Matthew MacDonald](#), [Rakesh Rajan](#)

**Paperback:** 592 pages

**Publisher:** Apress (January 3, 2006)

**Language:** English

**ISBN:** 1590595890



### CLR via C#, Second Edition (Paperback)

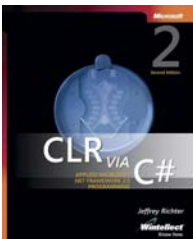
by [Jeffrey Richter](#)

Paperback: 736 pages

Publisher: Microsoft Press; 2nd edition (February 22, 2006)

Language: English

ISBN: 0735621632





## Detailed Syllabus

Week/Assignments	Topics
<b>Week 1: Jan 21<sup>st</sup> 2008</b>	<ul style="list-style-type: none"> <li>• <b>Course Details: Objectives, requirements, Assignments &amp; Logistics</b></li> <li>• <b>Introduction to C# through the eyes of a C++ or Java programmer.</b></li> <li>• <b>Building the first windows application, while introducing VS.NET IDE for design, development, and debugging</b></li> </ul> <p><b>Install Developer Studio.NET, Start working on Assignment #1</b></p>
<b>Week 2: Jan 28<sup>th</sup> 2008</b>	<ul style="list-style-type: none"> <li>• <b>Anatomy of a Windows Form-based application, inheritance, properties, events and delegates.</b></li> <li>• <b>N-tier rich and smart client architectures.</b></li> <li>• <b>Menus shortcuts, accelerators, toolbar button ImageLists, Splitters, Panels, Anchoring, Docking.</b></li> </ul>
<b>Week 3: Feb 4<sup>th</sup> 2008</b>	<ul style="list-style-type: none"> <li>• <b>More C#, CLR, .NET framework</b></li> <li>• <b>Memory management and finalization.</b></li> <li>• <b>Application framework, UI tiers. Standard, user and custom controls, dialogs, load and close events.</b></li> </ul> <p><b>Assignment #1 Due, Start working on Assignment #2</b></p>
<b>Week 4: Feb 11<sup>th</sup> 2008</b>	<ul style="list-style-type: none"> <li>• <b>More UI.</b></li> <li>• <b>More C#, CLR, .NET framework</b></li> <li>• <b>File I/O</b></li> </ul>
<b>Monday Feb 18th (Holiday Classes Suspended)</b>	<b>Holiday. Class is held tomorrow instead.</b>
<b>Week 5: Tuesday Feb19<sup>th</sup> 2008</b>	<p><b>Come in Today. Substitute Monday Schedule of Classes</b></p> <ul style="list-style-type: none"> <li>• <b>Registry I/O, Windows Clipboard and application data exchange.</b></li> <li>• <b>Binary and Soap Formatters.</b></li> <li>• <b>Serialization.</b></li> </ul>
<b>Week 6: Feb 25<sup>th</sup> 2008</b>	<ul style="list-style-type: none"> <li>• <b>Introduction to Database concepts using SQL Server 2000.</b></li> <li>• <b>Data Access: ADO.NET. Designing a simple database using SQL Server's Enterprise Manager.</b></li> </ul> <p><b>Assignment #2 Due, Start working on Assignment #3</b></p>
<b>Week 7: Mar 3<sup>rd</sup> 2008</b>	<ul style="list-style-type: none"> <li>• <b>Typed and untyped Datasets.</b></li> <li>• <b>Storing and retrieving textual and binary data to SQL server.</b></li> <li>• <b>Object-relational database layer.</b></li> </ul>
<b>Saturday, March 8th, through Sunday, March 16<sup>th</sup> 2008</b>	<b>Spring Break</b>
<b>Week 8: Mar 17<sup>th</sup> 2008</b>	<ul style="list-style-type: none"> <li>• <b>Graphics Device Interface of .NET known as GDI+.</b></li> <li>• <b>Painting foreground and background. 2-D vector drawing. Printing</b></li> </ul>



	<p>and print previewing.</p> <p><b>Assignment #3 Due, Start working on Assignment #4</b></p>
<b>Week 9: Mar 24<sup>th</sup> 2008</b>	<ul style="list-style-type: none"> <li>• Graphics continued: Typography, Anti-aliasing, double buffering and GDI+ Transforms.</li> <li>• Zooming, scrolling</li> </ul>
<b>Week 10: Mar 31<sup>st</sup> 2008</b>	<p>Drag and Drop, MDI, Threading</p> <p>Client application models and frameworks (SDI, MDI, MTI).</p> <p><b>Assignment #4 Due</b></p> <p><b>Project Proposal Due, Start working on Project</b></p>
<b>Week 10: Apr 2<sup>nd</sup> 2008</b>	<ul style="list-style-type: none"> <li>• Miscellaneous topics + catch up.</li> </ul>
<b>Week 11: Apr 9<sup>th</sup> 2008</b>	<ul style="list-style-type: none"> <li>• .Net Networking</li> <li>• TCP, UDP and HTTP</li> <li>• Sockets</li> </ul>
<b>Monday Apr 21<sup>st</sup> 2008 (Holiday Classes Suspended)</b>	<b>Do NOT come in on Monday. Come instead on Wednesday instead.</b>
<b>Week 12: Wednesday Apr 23<sup>rd</sup> 2008</b>	<b>Come in Today. Substitute Monday Schedule of Classes</b> <b>Catch up, Tools, Documentation, Best Practices, etc</b>
<b>Week 14: Monday Apr 28<sup>th</sup> 2008</b> <b>Last day of instruction for this class.</b>	<ul style="list-style-type: none"> <li>• Multithreading</li> <li>• Communication Foundation</li> <li>• NET Framework 3.0</li> <li>• WCF, WPF, etc</li> </ul>
<b>Friday May 3<sup>rd</sup> By 6 PM</b>	<b>Final Projects Due. . No exceptions.</b>

## Rules and Policies

Students are allowed to share ideas, techniques, and thoughts for solving problems. Students are NOT allowed to share any code. Code is private to each person, and any sharing, e-mailing, or posting of any code will be considered a violation of the Boston University academic conduct. Some examples of academic misconduct are cheating on exams; plagiarism (copying someone else's work of any kind and submitting it as your own work); unauthorized collaboration on homework or computer assignments; forgeries; theft of assignments or lab reports; and grade tampering. Conduct that is allowed in one course may be academic misconduct in another course.



I will reserve the right to reject any assignment if I have evidence that collaboration or copying of other people's work was involved, whether the copying was done from students currently attending the course or from any other source. These are the course regulations that I have set forth. By taking this course you are agreeing to the above rules and regulations.

Refer to the BU Metropolitan College Rules of Academic Misconduct:

[http://www.bu.edu/met/metropolitan\\_college\\_people/student/resources/conduct/code.html](http://www.bu.edu/met/metropolitan_college_people/student/resources/conduct/code.html)

## Class Cancellations Due to Storms or Emergencies

For the Charles River campus, are broadcast on WBUR(90.9) FM, WBZ (1030 AM), WEEI (590 AM), WHDH (850 AM); or by telephone, (617)353-SNOW

## BU E-mail accounts, applications, and e-mail forwarding

This course requires an ACS account. This account is the only means by which you can retrieve course information such as syllabus and required material, homework assignments, sample code, and solutions to assignments. Also, you'll have access to an online discussion board where you can ask questions and get answers. Once you've obtained an ACS account you'll get a BU e-mail address. This is the e-mail account that I'll be using for class updates and announcements.

This account can be obtained by following these steps:

### 1- Obtain a BU ID#.

All Boston University students are assigned a nine-character, computer-generated I.D. number. This University I.D. number or 'U' number replaces the old Boston University I.D. number or student's social security number and will be used for all University records including registration, class adjustments, and access to the Terrier Phone.

### 2- Obtain an ACS BU account. This requires your BU ID#.

Go to the following URL:

[http://www.bu.edu/htbin/cr/custom.pl?template\\_extension=acs&\\_hostname=acs](http://www.bu.edu/htbin/cr/custom.pl?template_extension=acs&_hostname=acs)

Follow the online instructions.

Your account will normally be available within twelve hours.

- **If you need help, contact the Office of Information Technology at 617/353-2780 or [it@bu.edu](mailto:it@bu.edu)**

For setting up mail accounts, using mail programs, acs web-based mail, etc, refer to the following URLs:



<http://www.bu.edu/webmail/>  
<http://www.bu.edu/pcsc/email/manage/webmail/faqs.html>

- For forwarding your BU e-mail to an e-mail account of your choice:

**Do not use the e-mail forwarding capability of CourseInfo. Even though that may seem to work, it will be reset every 24 hours and you will not get your e-mail forwarded. Instead use the following link:**

You MUST follow the instructions outlined in the following URL:

<http://www.bu.edu/computing/email/forwarding/>

