MET CS 503 Extended Syllabus

Spring 2011

Course Summary

This course will discuss Microsoft's.NET (3.5) and the C# language in contrast with C++ and Java, the .NET Base Class Library (BCL) 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.

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

Environment and Runtime Library and framework: 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 BCL core classes and components such as diagnostic, logging services, collections, and timers.

File I/O: we'll explore in detail File I/O using Stream and Binary Readers/Writers for sequential access, File Streams for random data access. We'll touch on XML Readers and Writers, XLINQ, then we'll dive into Serialization and the benefits it provides for data storage in binary, or SOAP 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.

Application: 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.

Database: 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. If time permits, we will cover the new Entity Framework for data access provided by .NET 3.5 SP1.

Graphics: 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.

Networking: 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.

Web Service and Windows Communication Foundation (WCF): 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

.NET CF and Windows Mobile: 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.

Course work

All coursework must be your own individual effort.
Four projects will be assigned. Each project assignment will be given two weeks. Projects should be handed in using the Online Student Drop box. Any other form of project submittal (including emails) will not be accepted.

Project #1	15%
Project #2	15%
Project #3	15%
Project #4	15%
Final Project	30%
Final Exam	10%

Projects will be graded based on 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.

For the final project, you will have about 5-6 weeks to complete. I will be supplying the project topic during the course of the semester. You can propose a different final project, subject to my review and acceptance.

NO LATE PROJECTS WILL BE ACCEPTED.
No Incomplete Grades.

Required Software

BU will provide the following software free of charge. I will facilitate getting everyone access to the download page.

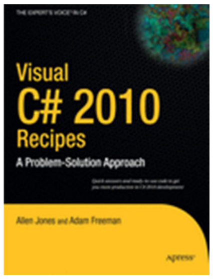
- Visual Studio.NET 2010 is required. **Do not use Express Edition**
  - The following editions of VS.NET or VC# will suffice: Team, Professional, or Standard Edition.
  - Visual Studio.NET 2003/2005/2008 will NOT be accepted.

- SQL Server 2008 is required. **Do not use Express Edition**
  - The following editions of SQL Server will suffice: Developer, Enterprise, or Standard Edition.
  - SQL Server 2005 or any other version will NOT be accepted.

Required Book

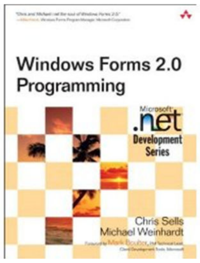
Visual C# 2010 Recipes: A Problem-Solution Approach

- **Paperback:** 1016 pages
- **Publisher:** Apress; 1 edition (March 25, 2010)
- **Language:** English
- **ISBN-10:** 1430225254
- **ISBN-13:** 978-1430225256

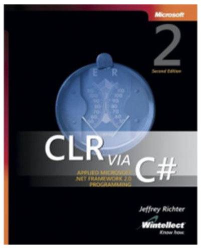


Recommended Books (Not required):

Windows Forms 2.0 Programming, 2nd Edition
by Chris Sells, Michael Weinhardt
Paperback: 1296 pages
Publisher: Addison-Wesley Professional; 2 edition (May 16, 2006)
Language: English
ISBN: 0321267966



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
Jan 20th 2011	• Course Details: Objectives, requirements, Assignments & Logistics • Introduction to C# through the eyes of a C++ or Java programmer. • Building the first windows application, while introducing VS.NET IDE for design, development, and debugging • Start working on Project #1 • Slides: Basic C#, Applications
Jan 27th 2011	• Anatomy of a Windows Form-based application, inheritance, properties, events and delegates. • N-tier Application architectures. • Menus, shortcuts, accelerators, toolbar button Image Lists, Splitters, Panels, User Controls, TreeViews, ListView, Comboboxes, and Windows Timers. • Slides: BasicC#, Basic.NET, Applications
Feb 3rd 2011	• Application framework, Form Validation and data exchange. Standard, user and custom controls, Forms, load and close events. • Keyboard and mouse handling. • More Visual Studio IDE, Debugging techniques. • Slides: Applications
Feb 10th 2011	• More C#, CLR, .NET framework • More User Interface Components and Application Concepts. • Project #1 Due, Start working on Project #2 • Slides: Basic.NET, Applications
Feb 17th 2011	• Memory management and finalization. • File I/O Slides: FileIO
Feb 24th 2011	• Registry I/O, Windows Clipboard and application data exchange. • Binary and Soap Formatters. • Serialization. • Project #2 Due, Start working on Project #3 Slides: FileIO
Mar 3rd 2011	• Introduction to Database concepts using SQL Server 2008. • Data Access: ADO.NET. • Designing databases using SQL Server's Enterprise Manager. Slides: Database
Mar 10th 2011	• Data Readers, Data Adapters, Typed and un-typed Datasets. • Storing and retrieving textual and binary data to SQL Server. • Object-relational database layer. • Project #3 Due, Start working on Project #4 Slides: Database
Mar 17th 2011	Spring Break (March 12-20)
Mar 24th 2011	• Graphics Device Interface of .NET known as GDI+. • 2-D Vector Graphics.

	• Printing and print previewing.
March 31st 2011	• Graphics continued: Typography, Anti-aliasing, Double Buffering and GDI+ Transforms. • Zooming, scrolling • Project #4 Due • Project Proposals Due • Slides: Graphics
Apr 7th 2011	• Drag and Drop • Client application models and frameworks • SDI, MDI, MTI • Slides : Applications
Apr 14th 2011	• Multithreading • .Net Networking • TCP, UDP and HTTP • Sockets Slides: Multithreading, Networking
Apr 21st 2011	• Virtual Monday for BU. No CS503 class.
Apr 28th 2011	• NET Framework 4.0, C# 4.0 • Windows Presentation Foundation. WPF
May 5th 2011	• WPF • Windows Communication Foundation (WCF) • Misc. Topics Last Day of Classes Final Projects Due by 6 PM. No exceptions.

Important Dates:

http://www.bu.edu/reg/dates/odates-1011.html

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

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
>
> 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**

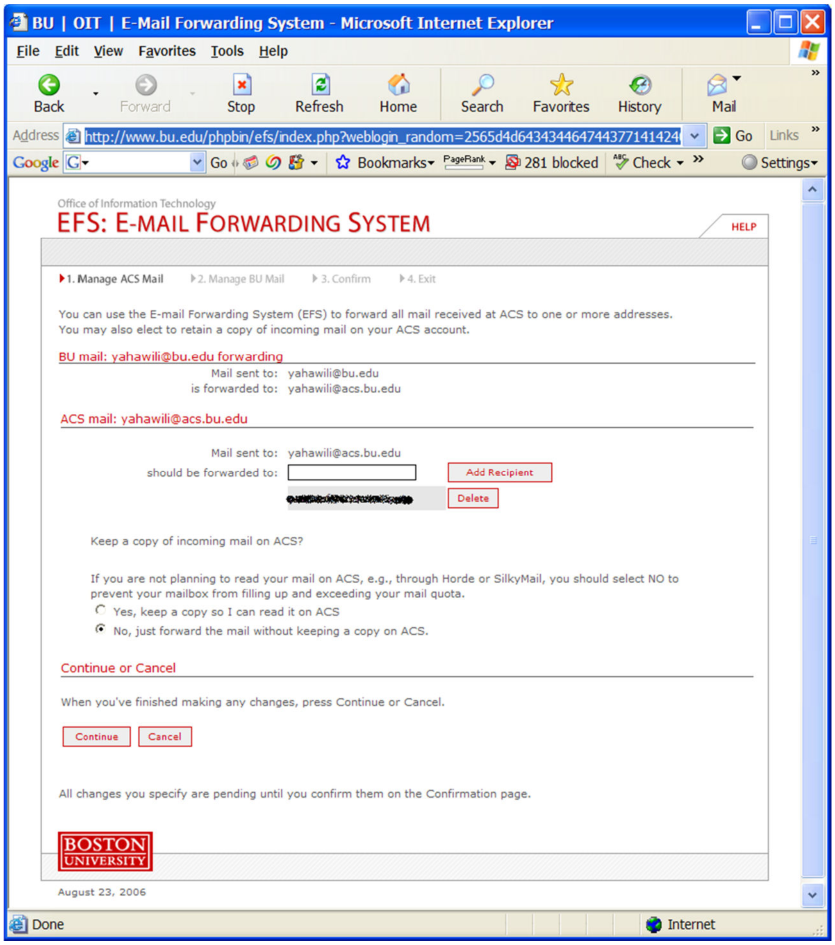
For setting up mail accounts, using mail programs, acs web-based mail, etc, refer to the following URLs:

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/



Address http://www.bu.edu/phpbin/efs/index.php?weblogin_random=2565d4d6434344647443771414241

Office of Information Technology

EFS: E-Mail Forwarding System

1. Manage ACS Mail 2. Manage BU Mail 3. Confirm 4. Exit

You can use the E-mail Forwarding System (EFS) to forward all mail received at ACS to one or more addresses. You may also elect to retain a copy of incoming mail on your ACS account.

BU mail: yahawili@bu.edu forwarding

Mail sent to: yahawili@bu.edu
is forwarded to: yahawili@acs.bu.edu

ACS mail: yahawili@acs.bu.edu

Mail sent to: yahawili@acs.bu.edu
should be forwarded to: ______ Add Recipient
[redacted] Delete

Keep a copy of incoming mail on ACS?

If you are not planning to read your mail on ACS, e.g., through Horde or SilkyMail, you should select NO to prevent your mailbox from filling up and exceeding your mail quota.

○ Yes, keep a copy so I can read it on ACS
● No, just forward the mail without keeping a copy on ACS.

Continue or Cancel

When you've finished making any changes, press Continue or Cancel.

Continue Cancel

All changes you specify are pending until you confirm them on the Confirmation page.

BOSTON UNIVERSITY

August 23, 2006