***Course Syllabus***

***Spring 2014***

***CS 3260-001 C# .Net Software Development***

***WB112 – TR 04:00pm to 05:15pm***

***Instructor: Dennis A. Fairclough***

***Version 1.0***

***Course Description***

In CS 1400 and CS1410 you learned the “***Fundamentals of Programming***” within C#/C++ object-oriented (OOP) concepts, now CS 3260 continues to build and develop on the concepts of C# object-oriented programming, using the advanced concepts of the C# programming language. A solid understanding of CS 1400 and CS1410 concepts is assumed, in this course. In CS 3260 students will increase their ability to do object-oriented design (OOD) and programming, and implement object-oriented programs in the C# programming language. It presents the ideas, tools, structure, syntax, and object-oriented design and programming techniques for developing well-formed object-oriented programs. Students will study concepts such as problem solving, program structure, classes, methods (functions), data types, control constructs, exception handling and file and console I/O. We will also continue delve into the C# language in-depth and learn other important principles in designing object-oriented programs. Additional C# topics covered in this course include inheritance, polymorphism, generics (templates), recursion, exceptions, operator overloading, and the collection generic library and many advanced C# concepts. Students program a number of assignments to demonstrate their understanding of these concepts.

**Course Objectives**

This course builds on the object-oriented programming principles taught in CS 1400 and CS1410. A solid understanding of these concepts is assumed and required in this course. In CS 3260 students will increase their ability to do object-oriented design, implement object-oriented programs as well as improve their skills in Graphical User Interface (GUI) design and programming in the C# language.

At the completion of this course, students should be able to

* Analyze difficult computing problems and develop object-oriented solutions to those problems.
* Apply good object-oriented design and analysis methods to develop an object-oriented program.
* Analyze, design, and develop GUI solutions to solve complex problems.
* Analyze, design, develop, test and debug programs to assure their quality, reliability and usability.
* Use a sophisticated integrated development environment (IDE) and appropriate design procedures to construct reasonably complex C# programs that:
* Use references,
* Use inheritance and polymorphism,
* Use overloaded operators,
* Use programmer written methods and method templates,
* Use programmer written classes and class templates,
* Use standard exception handling techniques,
* Correctly manage memory (mainly handled by the Garbage Collector),
* Use recursion,
* Use classes and algorithms from the Collections Library,
* Design and use graphical user interfaces (GUI),
* Test and debug programs to assure their quality and usability,
* Document programs for understandability and maintainability by providing in-line comments, prewritten classes and elements of good programming style,
* Demonstrate their understanding of simple and complex data structures,
* Demonstrate their understanding of C# data types and data structures,
* Demonstrate their understanding of C# file I/O,
* Demonstrate their understanding of C# GUI interface analysis, design, development, testing and documentation,
* Demonstrate their understanding of generics,
* Database design and interfacing,
* Demonstrate their understanding of extension methods, LINQ, and
* Demonstrate their understanding of advanced C# concepts.

***Topical Sequence***

The following is a list of the topics to be covered in the course, in the approximate order of their presentation (This sequence is subject to change):

* Introduction to Object-Oriented Analysis, Design and Programming (OOA, OOD and OOP),
* Introduction to C# and the .Net Framework,
* Introduction to GUI Analysis, Design, Development and Programming,
* C# Language Basics,
* C# Data types,
* .Net Framework and CLR,
* Advanced C# & Framework Fundamentals,
* Regular Expressions,
* Exceptions,
* Generics and Collections,
* Advanced C#,
* LINQ Queries,
* LINQ Operators,
* Disposable, Garbage Collection & Memory Management,
* Streams, File I/O and Serialization,
* Assemblies,
* Reflection & Metadata,
* Threading, and
* Dynamic Programming.

### Course Materials

* **Required Textbook:**  
  **“C# 5.0 in a Nutshell”**, by Joseph Albahari; Ben Albahari  
  Oreilly Media, Inc., ISBN Number: 978-0-596-52757-0
* **Optional References - not required, but very helpful, for this course**

|  |
| --- |
| C# 3.0 Unleashed: With the .NET Framework 3.5, by Joseph Mayo, |
| Sams ISBN-13: 978-0-672-32981-4 |

|  |
| --- |
| Programming C# 3.0, 5th Edition |
| by Jesse Liberty; Donald Xie, O'Reilly, |
| |  | | --- | | Print ISBN-13: 978-0-596-52743-3 | |

* + *A good UML reference*  
    ***Fowler, M.* and Scott, K.**, "UML Distilled:,  
    Addison-Wesley, ISBN 0-201-32563-2
  + ***Textbook –*** CS1400Course Notes, available in the Bookstore.
  + ***Fowler, M.*** (2000).  *UML distilled:  A brief guide to the standard object modeling language*   
    (3rd Ed).  Reading, MA: Addison Wesley.

***Computer Labs***

* You have paid a small lab fee which covers a very small portion of the costs of running the computer lab. Within 48 hours of registering for the lab, your CS lab account should be set up. The account is based on information supplied by the UVU registration system. Your user name is your UVU ID. Your default password is your birth date in ***MMDDYY*** format.

***Example:***    
***Student Name:***  Ima Programmer    
***Birth date:***     2/28/1980

***Login Name:***    10001234   
***Password:***      022880

* The lab for this course is located on the 6th level of the Computer Sciences and Engineering Building. The website for the computer labs is located at [http://csevpn.tc.uvu.edu.](http://csevpn.tc.uvu.edu/%20) The lab is not responsible for network hard disk failures, so you will want to store backups of your assignments on removable media such as a USB drive. REMEMBER, “The computer mucked up my files” or the “dog ate my homework” is NOT an excuse☺
* To do the labs for this course, you will need to install Microsoft’s Visual Studio 2013 Professional. You can obtain this [www.msdn.com](http://www.msdn.com) and the license key can be obtained from the MSDNAA. The course Canvas website has a lot of information that will be helpful to for this course. The Canvas website may be accessed at <http://learn-uvu.uen.org> in the event that there is an ***emergency shutdown*** of the UVU campus you must continue your course instruction by using the information that I will post on this Canvas website in the ***“Emergency Instructions”*** folder. If you are ill please ***DO NOT*** come to class.

### *Course Procedures*

The format of this course will be, a graded quiz given on the assigned reading and videos, at the beginning of ***every*** class and will also alternate between presentations, in-class programming demonstrations and exercises, class discussion, and "think-aloud" design sessions. Class time will be devoted to addressing key issues, clearing up questions from previous lectures, providing a forum for discussion, and reviewing solutions to in-class exercises and programming assignments.  Your examinations will concentrate on the concepts and terms as covered in the lesson material and your ability to model solutions to problems using an object-oriented programming language. There will be in labs, exams, and programming projects to test your comprehension of the material.

In the event that there is an ***emergency shutdown*** of the UVU campus you must continue your course instruction by using the information that I will post on this Canvas website in the ***“Emergency Procedures”*** folder. If you are ill please ***DO NOT*** come to class.

**Course Philosophy**

Traditionally, the role of the student, in the classroom, has been relatively passive. The professor professes and the student quietly takes notes. Not so in my classroom. It's your education. I'm just here to help (***facilitate***). I believe in active participation by students in their educational process. This puts the primary responsibility for learning where it belongs -- on ***your*** shoulders. Come prepared by reading the assigned chapters and doing the programming assignments so you can get involved in classroom discussions. Together we can have a lot of fun reaching the course outcomes.

**Course Communication**

In the fast-paced world of technology, the only thing that is constant is change. In such a world, the ability to communicate quickly and efficiently is critical. To keep you abreast of changes in the world of computer science and changes in the course content and assignments or ***emergency shutdown*** of the UVU campus, we will use the course Canvas website, email for CS 3260 should be sent to [fairclde@uvu.edu](mailto:fairclde@uvu.edu) with the Subject as CS3260, (I ***DO NOT*** use Blackboard or UVLink email).

When you send me an email, send it to [fairclde@uvu.edu](mailto:fairclde@uvu.edu), please put the Subject CS 3260 Section 001. Emails without this phrase in the subject line may not get read. Send your zipped project folder to [Canvas](mailto:1410dennis@gmail.com) to be graded. DO NOT send your programs to my email address!

My Canvas website is the ***only official*** site for course material. To keep you abreast of changes in the world of computer science and changes in the course content and assignments or ***emergency shutdown*** of the UVU campus, we will continue to use the course Canvas website.

**Canvas**

This course ***DOES NOT*** use Blackboard it uses [Canvas](mailto:1410dennis@gmail.com) .

### Labs

Programming labs are provided to integrate and reinforce concepts covered in this course. You should plan on spending anywhere from 3 to 8 hours per assignment. It's usually best to complete each lab shortly after finishing the lesson. Once completed, submit your ENTIRE Project file zipped (.zip file) to [Canvas](mailto:1410dennis@gmail.com) .

Assignments are due no later than 11:59am on the date specified in the course calendar. That’s the bad news; the good news is I will accept labs one (1) week after the due date. Projects that are submitted after this will receive 0 credit. Labs, even late ones ***MUST*** be turned in even if they are very late and may even receive sympathy points; however, this is ***very problematic***. Date and time of submission are based on the timestamp applied to your project when it is received by Canvas.

Each lab you submit will be unzipped, loaded into Visual Studio 2013, compiled and run to see if meets the lab specifications. A substantial part of your grade will depend on whether your code executes according to the lab specifications. A Checklist (Rubric) is provided for each assignment to help you make sure that your program meets the specification; nevertheless, DO NOT submit the Checklist, this is for you only and to verify you have completed everything for the project.

The other part of each lab's grade is based on coding style, simplicity and conformance to the Style Guidelines provided on Canvas.

Before submitting your lab, zip your ENTIRE Project folder, which will include all of the required files into a single zip archive. Name the file with your initials an underline assignment number an underline and the version number. For example, if I were submitting lab programming Lab 1, I would create a zip file named   Lab\_01\_DAF\_V1.0.zip. Once you have zipped together your Project file and named it, then submit it to [Canvas](mailto:1410dennis@gmail.com) .

**Checklist**

Every lab has an associated Checklist (Rubric). Before you turn in a lab, please look at its Checklist to make sure that your program meets all of the requirements for that lab. Please DO NOT include the Checklist, when you zip your project, it is there to help you make sure you have done everything.

**Grading Penalties**

When your project or lab is graded, points may be deducted when the assignment does not meet certain style and submission criteria. Point deductions for style and submission errors include the following:

|  |  |
| --- | --- |
| Submission | |
| No or invalid Project Prolog | 0 points† |
| No or invalid method Prolog’s | 0 points† |
| No "did not copy" declaration | 0 points† |
| Missing files | 0 points† |
| Project loads and compiles | 0 points† |
| Executable does not run | 0 points† |
| Files are not zipped together properly | 0 points† |
| Zip folder is not properly named | 0 points† |
| Zip folder contains unnecessary files | 0 points† |
| Style | |
| Code contains magic numbers, strings or characters. | 0 points† |
| Incomplete or missing function prolog’s | 0 points† |
| Invalid UML notation | 0 points† |
| Improper use of braces and indentation | 0 points† |
| Improper naming of variables and constants | 0 points† |

†When an assignment is not graded or does not receive full credit, you may correct it and resubmit it for ***FULL*** credit with one (1) week after you receive notification from the grader that it was ***NOT*** accepted. Make sure to change the version number to reflect this, i.e. if the original submission was V1.0 the resubmission should be version number V2.0. Labs will receive a grade of 10 points if correct or 0 points if it contains ***any*** problems.

### Labs

Each week of this course you will be expected to complete a Programming Lab. These labs are all provided on the course Canvas website ( <http://uvu.instructure.com> ) in the "***Labs***" folder. The programming labs provide additional material, beyond that provided in the textbook and give you an opportunity to explore important programming concepts.

You should plan on spending anywhere from 3 to 8 hours each week working on the programming labs. Each lab consists of additional study material, a programming project that reinforces the topic studied. Labs are always due on the dates specified. It is important to note that labs are allowed to be 1 week late and they will receive ***FULL*** credit and after that they MUST be turned in late and will **not** receive any credit. You should zip together the ENTIRE Project folder; however, DO NOT include the completed Checklist, and submit them to Canvas before 11:59am on the date that they are due.

Once you have zipped your ENTIRE Project folder, submit it to [Canvas](mailto:1410dennis@gmail.com) . Remember to name your file as Lab\_01\_DAF\_V1.0.zip, the name is Lab\_01, the \_DAF is your initials the \_V1.0 is the version number and .zip is the file extension. Make sure you submit **all** the files asked for in the lab.

### Exams

Two midterm exams will be given as take home exams, on the days indicated in the course schedule. Because the exams will be available for several days, ***NO MAKE-UP EXAMS WILL BE GIVEN***. The ***Final Exam*** will be administered in class on the day indicated in the schedule; however, you MUST pass the Final Exam with a score of ***60%*** or better to be able to pass the class.

***Grading***

Evaluation of your achievement of the course objectives will be based on the following and is *subject to change during the semester*:    
Programming Labs 95% to 100% +1 full grade

>85% and <95% + ½ grade

<85% to >65% - ½ grade

65% or less - 1 full grade

Quizzes 80% to 100% + ½ grade

>70% to <80% + ¼ grade

<70% to 0% - ½ grade  
Exams (2 Midterms & 1 Final)           450 points (Approximately)  
Total  Points    450 points +/- Lab and Quizzes

***Grades will be assigned based on the following scale-***  
**Note:** You cannot pass this course without a passing grade (60% or better) on the final examination!   
  93% and above  A   
  90%-92%         A-   
  87%-89%         B+   
  83%-86%         B   
  80%-82%         B-   
  77%-79%         C+   
  73%-76%         C   
  70%-72%         C-   
  67%-69%         D+   
  63%-66%         D   
  60%-62%         D-   
  59% and below   E

### Students with Disabilities

If you have any disability that may impair your ability to successfully complete this course, please contact the Accessibility Services Department (room LC-312, ext 8747).  Academic accommodations are granted for all students who have qualified documented disabilities.  Services are coordinated with the student and instructor by the Accessibility Services Department.

### Withdrawing From Class

Refer to the University web site for the deadlines for tuition payment for full semester classes.

Refer to the University web site for the last day to drop this course without it showing on your transcript, and to receive a 100% refund.

Refer to the University web site for the last day to withdraw from this course or change to an audit. Withdrawal from the course after this date requires the department chair's approval. Approval is not normally given, except under extenuating circumstances. **Note:** If you stop coming to class without officially withdrawing from the course, you will receive an **'E'** even if you are passing the course at the time you stopped attending.

**Note:** If a student stops attending (but does not officially withdraw) before the last day to drop, he/she will receive an “***E*** ”. If a student stops attending (but does not officially withdraw) beyond the last day to drop, he/she will receive the grade earned up to that point, which will be an “***E*** ”.

### Honesty

If attempt to cheat your way through this course, you are only cheating yourself. It will catch up with you in this and subsequent courses, or when you start your first job in the computer industry. I assume you are here to learn. You only learn to program by writing programs ***yourself!***

I strongly encourage you to find a friend, or better yet, a group of friends to study with for this course. It often helps to have someone else explain an idea, or explore a different solution to a problem. However, I expect that all work that you hand in will be your own. If I find that you have copied all or part of someone else’s code, I will give you **and** the person you copied from a zero on that particular assignment.

All programs must contain the following declaration as comments in the file prologue. The declaration must be signed on the hard copy of the code that you turn in to your instructor:

***"I declare that the following source code was written solely by me. I understand that copying any source code, in whole or in part, constitutes cheating, and that I will receive a zero on this project if I am found in violation of this policy.***”   
   
If you have questions, don't understand a concept, or are having problems understanding why a program, you are working on does not work, come by and talk to me. I strive to keep long office hours, so that I can help students. My office is CS520h.

There are no tutors that are available to help you, sorry. However; I try to be in my office CS520H as much as possible and will be happy to assist you in anyway I can. See “Contact Info” on Canvas for the times I will be in my office.

***That’s all folks!***