

INFO 3231 – 001
Business Applications Development

Spring 2008
Wednesday, Friday 11:00 am – 12:15 pm
Friday, Room 04

INSTRUCTOR: Bong-Keun Jeong
OFFICE: 204 Barnard
PHONE: 687-6716
OFFICE HOURS: Wednesday, Friday 9:30am - 10:50am or by appointment
EMAIL: bjjeong@uncc.edu

COURSE DESCRIPTION

A study in the development of business applications software. Course emphasizes graphical user interface development using object-oriented, event-driven programming methods and techniques with a high level development tool (Microsoft Visual Studio).

Prerequisites: Undergraduate level INFO 2231 Minimum Grade of C or Undergraduate level ITCS 2231 Minimum Grade of C or Undergraduate level ITCS 1214 Minimum Grade of C or Undergraduate level ITCS 1215 Minimum Grade of C

COURSE OBJECTIVES

Upon completion of this course, each student should be able to effectively design, develop, and test business applications written in Visual Basic (VB) using Integrated Development Environment (IDE). The learning objectives include:

- To utilize predefined classes in .NET Framework Class Library and on-line documentation.
- To define, describe, and explain general coding and formatting rules with appropriate data types for specific applications in Visual Basic
- To understand and use decision and repetition statements in a VB program.
- To define, describe, and use methods and objects in VB.
- To use arrays and strings.
- To create appropriate graphical user interface (GUI) for windows applications.
- To create and use user-defined classes and class libraries.
- To develop, test and document a professional looking software package.

TEXTBOOKS

Required:

- *Microsoft Visual Basic 2005: RELOADED, 2nd Ed.* by Diane Zak

Recommended:

- *Visual Basic 2005: How to Program, 3rd Ed.* by Deitel and Deitel.

TENTATIVE COURSE SCHEDULE

The Instructor reserves the right to change the course contents and/or schedule. Consult this page and course materials page often so you can have up-to-date course schedule.

Week	Date	Topic	Reading Assignment (See course materials page in Blackboard for more reading assignments)
1	1/9	Course Introduction	
	1/11	Introduction to Visual Studio 2005 IDE	Ch. 1
2	1/16	Creating a User Interface	Ch. 2
	1/18	Variables, Data Types and Operators	Ch. 3 and App C
3	1/23 **	Variables, Data Types and Operators	Ch. 3 and App C
	1/25	Selection Structures	Ch. 4
4	1/30 **	Selection Structures	Ch. 4
	2/1 **	Exam 1 (Ch.1 - Ch.4)	
5	2/6	Repetition Structures	Ch. 5
	2/8 **	Repetition Structures	Ch. 5
6	2/13	Defining and Calling Procedures	Ch. 7
	2/15 **	Defining and Calling Procedures	Ch. 7
7	2/20 **	Defining and Calling Procedures	Ch. 7
	2/22 **	Exam2 (Ch. 4, Ch. 5, Ch. 7)	
8	2/27	Array	Ch. 8
	2/29 **	Array	Ch. 8
9	3/5	Spring Break	
	3/7	Spring Break	
10	3/12	Classes and Objects in Visual Basic .NET	Ch. 10
	3/14 **	Classes and Objects in Visual Basic .NET	Ch. 10
11	3/19 **	Classes and Objects in Visual Basic .NET	Ch. 10
	3/21	Spring Recess	
12	3/26 **	Classes and Objects in Visual Basic .NET	Ch. 10
	3/28 **	Exam3 (Ch. 8 and Ch. 10)	
13	4/2 **	Multiple Document Interface (MDI)	Class Examples
	4/4 **	Multiple Document Interface (MDI)	Class Examples
14	4/9 **	Multiple Document Interface (MDI)	Class Examples
	4/11	ADO.NET 2.0	Ch. 11
15	4/16 **	ADO.NET 2.0	Ch. 11
	4/18 **	ADO.NET 2.0	Ch. 11

16	4/23 **	ADO.NET 2.0	Ch. 11
	4/25	Group Presentation	
17	5/7 **	Final Exam (MDI and Ch. 11)	
** indicates a lab session (either lab 339 or 280)			

GRADING

The following ten-point grading system (based on the student's overall average) will be used:

90-100	A
80-89	B
70-79	C
60-69	D
Below 60	F

The average is composed of the following items and weights:

Exams (4) 20% each	80%
Programming Assignments	10%
Group Project	10%

WORKLOAD

I expect that you will spend at least 6 hours per week outside of class time for assignments, projects, and exams. If you feel you forgot much about programming from prerequisite courses, spend more time trying out examples so you cannot be left behind. Due to limited lab availability, you should improve your programming skill by yourself with the help of the textbooks. The textbook (Zak) serves you well for this purpose. Writing your own program takes time, but I believe you will enjoy it eventually.

COMMUNICATIONS

The best way to contact me is via email. If you have to send a big attachment, use Blackboard email. I found the discussion board in Blackboard very useful for communications with students outside class. You may get additional information about assignments, projects, and exams. You are responsible for any announcements in Blackboard. Thus, check it out as often as you can.

ASSIGNMENTS AND PROJECT

All homework assignments must be completed individually. Homework deadline is always at the beginning of the class when it is due unless specified by the instructions. In case of a late submission on the same day, 20% of the points earned from the submission will be deducted. Late homework will not be accepted after the due date, and you will receive a 0 for that assignment. Each assignment will be posted at least one week ahead of its due date. Assignments may be replaced with quizzes.

Students are encouraged to discuss homework assignments with other students (but not to copy from others). Refer to student conduct section to avoid any misunderstanding about cheating.

Each student will form a group (maximum size of four) to complete a group project, the requirements of which are to be announced on or before 2/29.

EXAMS

You must take all four exams including the final exam.

There will be two parts of the exam. Part A will test your general knowledge gained from textbook readings and classroom discussions. Part B will test whether you can apply the knowledge into programming and hence contains hands-on programming.

In order to be ready for the hands-on part (Part B) of the exams, do not just read the book. Instead, *try out as many examples as possible*. Learning how to program does not mean you understand the logic by just looking at an example code. You should at least be able to reproduce your own version of working code for each example code. Again, remember that you learn a lot more from such programming practices than from a reading of hundreds of pages in the textbooks.

ATTENDANCE POLICIES

Students are expected to attend all classes although attendance itself is not going to be counted towards the final grade.

Students will be kept responsible for any materials covered, announcements made, assignments passed out, and any other type of work they may miss during any absence from class.

INCOMPLETE GRADE POLICY

The "incomplete" is not based solely on a student's failure to complete work or as a means of raising his/her grade by doing additional work after the grade report time. An incomplete grade can be given when a student has a serious medical problem or other extenuating circumstance that legitimately prevents completion of required work by the due date. In any cases, the student's work to date should be passing, and the student should provide proper written proof (e.g., a doctor's note), in order to get an 'I' grade.

STUDENT CONDUCT

- All students are expected to comply with the UNCC Code of Student Academic Integrity, Code of Student Responsibility, and all University Policies governing student conduct. Cheating on exams, plagiarism of papers and software projects, and any other dishonesty of any type in this course will result in a final grade of F and will be reported to the appropriate University authorities for further discipline.
- There are many forms of cheating, and it is very difficult to list all the forms. In general, you must type every word of your program. Although you can discuss and exchange ideas with others, you are not allowed to exchange files. Do not take someone else's code and copy and paste it into your program. If you have to use other's code (e.g., an algorithm found on the internet), always reveal the source in programmer's comment.

Computers are very stupid but honest. Most of the programming errors are coming from the misunderstanding that your computer can understand what you order.