

CIS159 – Visual Basic I

Lesson One

Instructor: Gary Marrer

Lesson One Objectives



- **Topics:**
- Class Intro: Introduction to Class (syllabus, class calendar, rules, format)
- Introduction to Computers
- History of Basic/Visual Basic Versions of Visual Basic
- Event driven programming
- First application
- Using Visual Studio IDE
- VB Errors
- Using Visual Studio Help

Welcome to Class!

Over the next sixteen weeks, we will become partners in learning Visual Basic.Net (VB). The course will be managed over the Internet with BlackBoard (Bb). Bb is used to present course materials along with being our official grade book. We will also have three optional (but strongly recommended) class meetings (dates and times published in the course schedule) where you will get the opportunity to address questions to me directly and interact with other students on the GCC Campus. I will also conduct study group sessions several times throughout the semester where your class and my other OOP Fundamentals class can come on campus and ask me questions about homework or test preparation. Again, these study sessions are not required.

I will provide the foundation (via lectures, assignments and tests) and each of you will read the lectures, textbook, and complete the exercises, assignments and tests. The content delivered to you on the Internet via BlackBoard is the same as the content is delivered in my traditional classrooms. However, instead of me being able to call on you with questions, we will be learning asynchronously where the timing of our communication will no longer be real-time. Our communication and discussion may take place the same day or maybe the next day or later in the week. We will need to keep this in mind as we communicate with others.

The sequence of topics covered in class is the same sequence in the textbook. For that matter, the textbook will drive a lot of what is discussed in class. I will not repeat the book in my lecture but may elaborate or expand on key topics. I will refer to the book in my examples and identify key sections in the book that cover class topics. The textbook is an integral part of hybrid learning.

An update: Please note that this class will be using Visual Basic Express 2008 and Visual Web Developer Express 2008. The computers on campus in the High Tech Center are equipped with Visual Studio 2008 Professional which includes both of these products (Express editions are not installed on campus). You can use XP, Vista or Windows 7 for this class.

[Click here to go to the Microsoft Express 2008 Download Page \(Free Download\)](#)



On-line classes (past experience)

I have taught on-line classes for several years using several on-line educational tools (WebCT, Web BlackBoard, home grown (with HTML/JavaScript/Java applets), Midas, etc.). I have also been an on-line student myself. On-line learning is the same and is different from traditional classes. They are the same in that both on-line and traditional class formats use a textbook and lectures along with tests and labs to teach you Visual Basic.Net. They are different in that on-line classes require the motivation to work at readings and assignments daily. Many believe that on-line classes are easier and require less effort. ***This is a fallacy.*** There is more flexibility with on-line classes but not less work. One of the primary success factors for any programming class is to stay up with the reading and lab assignments. This is critical. You will have weekly quizzes to test your comprehension of the chapter and give me feedback on your progress.

Where to start!

One requirement to participate on-line classes is that you need to know how to use the tools. You will need to explore BlackBoard (Bb) and the materials I have placed there. If you have not already done so, review the class tools video. If Bb is unavailable, you can always access lessons online from my faculty web page at <http://web.gccaz.edu/~gmarrer> and clicking the CIS159 link. We will also review Bb in our first GCC campus meeting.

Logging on with your Username and Password

We have a couple of systems that require logins. Our primary login is your Instructional account is your eGCC username and password. If you do not know what your username and password is, please let me know. We will be logging into all systems in our first class meeting. Even if you do not use GCC on-campus computers you will need to know your eGCC username and

password to submit your assignments via FTP to the courses/shared folder (more on this in a couple of weeks).

eGCC academic network (formerly called the Palette)

username - eGCC username (Instructor has this for new users)

password - eGCC password (see Student Help Desk Link in Bb for instructions)

BlackBoard

username - Enterprise User ID: Need an (MEID) Enterprise ID?

password - (MEID) Enterprise password

Questions.... Questions.... Questions

I love questions. If you have a question that you don't mind sharing with the class enter that question into the one of the question topics in BlackBoard Discussion Forums (VB Assignment Questions and VB Language Questions). If you have a question, which you would like to ask me and not share with the class, please send it as email. I will return responses to email sent to my personal account within 1 business day. I have placed on my lecture web pages several Ask Me



icons which when clicked will open up email with my address so that you can send me your question. Try it and send me an email now!

Format:

Each week the lecture document will provide the roadmap for the rest of the week. The topics found in this lecture document will remain more or less the same.

Objectives for week (Each week has a different set of objectives)

Lecture (lecture weekly) – main points reviewed

Assignments

Book highlights

My Turn: experiences in the field to support the class objectives

Chapter Review

Grading

There is nothing very scientific about the grading for this class. There are a total of 750 **points** (**Syllabus has details**) for which you receive more than 90% you will receive an A, 80% a B, 70%

a C and so on. You receive points for taking tests, participating in class (quizzes and discussion questions) and turning in lab assignments.

You must complete all assignments to receive a grade of A and get at least a score of 8 out of 10.

CIS159 - Visual Basic I

The goal of this class is to give you an introductory understanding of the Visual Basic programming language. The topics covered come from a set of core competencies developed by the Maricopa Community College District to insure that all CIS159 students finish the course having covered the base topics (the competencies document can be accessed via the on-line class schedule on a link in the top right section of the course listing. **VB is a fun language and since it is a very mature language it has many tools and design aids to make programming in this environment easier than with other programming languages.** Keep in mind, as easy as it seems, VB can do all of the complex programming needed for today's Windows and Internet client/server component based development environments.



Lecture

Chapter 1

Introduction to Computers:

I am going to start with a review of the **pre-requisite knowledge**, or general computer knowledge and basic Microsoft Windows operating system skills that everyone should have coming into this class. **All of you should have this knowledge via CIS105 or a waiver.** Even though this section should be review, please don't be afraid to ask questions.



Some say to really understand how technology works you need to understand its history and how it was designed (this seems to make sense to me also). Microsoft's Visual Basic (VB) didn't

start off in 1991 without a little help. Visual Basic is an ancestor of the BASIC programming language. BASIC was created in the 1960's by two professors at Dartmouth College. John Kemeny and Thomas Kurtz developed BASIC for use by their students at a time when there were few programming options available. The original versions of BASIC used an interpreter that accepted BASIC source code and converted it at run-time into the appropriate machine code commands for the Operating System executing the program. This process is slower than traditional executable files (.EXE file extension) but since the interpreter for BASIC runs until completion or until it finds an error, it is a great programming language for students who get immediate feedback on their errors (VB programs take advantage of this and can be developed using the interpreter but can later be converted into .EXE files).

The BASIC programming language is a survivor. It has managed to evolve itself (just as the shark and alligator) and remain vital when many Information Systems (I/S) technologies have become obsolete after just a few years. What started as an unstructured interpreted language has with .NET moved to a fully fledged Object Orientated Programming environment and one of the foundations of the new Microsoft .NET technology. With the addition of VB.NET, Visual Basic is on par with Microsoft C++ in both power and performance. In fact, with .NET, C++, C# and VB all share the same class libraries so therefore have the same building blocks used in executing the program. In **CIS259 (Adv. Visual Basic)**, we will spend more time understanding the implications of .NET and .NET Framework class libraries.

A side note....

BASIC did have one factor in its favor. It was always rumored that Bill Gates preferred BASIC to other programming languages (the BASIC interpreter was Microsoft's first product - even before MS-DOS) and has made sure it was shipped with all of the Microsoft Operating Systems prior to Windows 98 (it has gone under the names BASICA, GWBASIC, Quick BASIC). With the introduction of the .NET environment, Microsoft hopes to re-revolutionize programming (especially on the Internet) and it is no secret that VB is there again to introduce another new Microsoft architecture. With the introduction of the personal computer, BASIC had widely been accepted as an easy to use programming language for the personal computing public. With the introduction of Microsoft Windows, Microsoft needed to bring the "peoples" programming language to the new Operating System. They did this with VB and got the unexpected consequences of it also becoming a favorite of corporate America. Today, Visual Basic continues to be one of the most popular programming languages (a lot of this is helped by the fact it runs on the world's most popular Operating System). VB is designed to provide the programmer an IDE (Integrated Development Environment) that provides a program source code editor, a toolbox with common windows controls (textboxes, command buttons, list boxes, etc.) a compiler to create a standalone executable .exe file, a debugger to help find logic errors and other assorted tools to make the programmer more productive. With .NET Microsoft took the best features of the VB, C++ IDE's and combined them into one IDE called Visual Studio IDE. We will be using this IDE to develop our programs. Familiarize yourself with the IDE and the toolbar and menus.

Versions of VB.NET

We will be using Visual Studio.NET Express 2008 Edition in class. This edition was especially designed to include Visual Basic.NET 2008 only and was designed specifically for new VB programmers. You can also do all of the assignments in one of the High Tech Centers on the GCC campus. Note: Visual Basic.Net only runs on the Windows Operating system. A program you create in class will not run on the Apple Mac OS or Linux operating systems.

GUI - Event programming

VB (as does most other programming languages that create applications that use the Windows GUI (Graphical User Interface)), rely on the process of events to control the flow of the program. An event may be the click of a button or icon or dragging or dropping a screen object from one area to another. As we move into weeks two and three, we will begin assigning VB source code (instructions for our computer) to various events performed by our graphical user interface. Event processing makes VB a very different programming language than its distant relatives (DOS BASIC) and when coupled with the screen graphical objects we will use in our VB programs (textboxes, command buttons, list boxes, etc.), you will create state of the art applications for the Windows Operating System. You can think of event driven programs as a program that is running waiting for the user to do something on the interface. Once a button is clicked or a menu option taken, code is executed behind the scenes to perform the requested option. Any questions?



Steps in development of a VB Project (also called application)

Like any quality process, programming should include a staged methodology, which will help insure that we create programs to the requirement and at high quality. One set of stages suggested by the textbook author included the following five steps. Many of you may recognize these steps from SDLC (System Development Life Cycle), which is the basis for many of the methodologies used to develop software and systems.

Five Stages in Designing Software

(Manual operations)

Draw Interface - *what graphical controls (textboxes, command buttons, etc.) will be used*

Label Controls - *Identify and catalog all of the graphical components used and key properties*

Pseudo code - *document program logic and program control* **(Done with Visual Studio IDE)**

Code - create and edit source code (program instructions)

Test-Debug - test the program source to make sure it meets requirements

Building your first project

When you build (create and update) your project it is done in what is called **design time**. This is important because you will find that different features are available at design time and other features may be available at **run time** (when you are running your application) and break time (when you pause your program's execution). I do not want to repeat the book so **I want you to focus on pages 14-46** and see how projects are started and changed. See how you add graphical controls to a form and then program events trigger by those controls to give the program the ability to interact with the user and process program logic. When you start the Visual Studio.Net the IDE should bring up the **Start Page** first. **Click on New**. Make sure Visual Basic is selected on the left and on the right side of the dialog box, look for Windows Application. **If any of this part of the text is confusing, ask a question in the BlackBoard - VB Assignment Help Discussion or send me an email.**



Visual Basic Objects/Controls

When it comes to VB objects/controls we need to define a few terms and start to create a vocabulary for future class discussions. Objects (as you will learn in Chapter 6) are created by languages that support Object Orientated Programming techniques (OOP). We will not get into the details now, but VB does support OOP. We can also refer to (and I will) these objects as controls and components (all these words are interchangeable in this class). For whatever we are going to call them, these objects are graphical objects that will be placed on VB forms. We will be using these graphical controls to activate events that will cause sections of source code (source code is the actual VB statements) to execute and run our program (i.e. adding numbers, changing text, etc.). I would suggest that you read this chapter very slowly and go step by step through the walkthroughs for textboxes, command buttons etc. .

Object = Properties, Method and Events

Every object has properties, methods and events. You will be manipulating properties by changing the settings (background color is a property common to almost all controls) calling methods (a method is a built-in programs associated with that control) and supplying code for associated events (the click event is common to most controls). You can investigate properties by clicking on the control and then looking through the Properties Window and look at events by clicking on the Code Window icon (located in the Project Explorer window) and then clicking

on the combo box located in the top right corner of the Code Window dialog box). Methods are best reviewed with the on-line help and doing a search on the control name and the keyword method. We will cover methods and event extensively in the following chapters. For chapter one, we will just get into the basics.

Note: If you lose (no longer see on the screen) your **Solution Explorer**, **Controls Toolbox** or **Properties Window**, go to the VB toolbar and hold your cursor over each icon until you find the appropriate one. Click on the icon and the dialog window should reappear. You can also access these tools under the View option on the menu bar.

VB Errors....

Yes, you will have errors. You will have **syntax errors** (associated with creating invalid VB source code statements) , **run time** errors with the programs are written correctly but have server error that prevents them from running in Windows (maybe a file cannot be found) and logic errors which are the hardest to fine. **Logic errors** don't display errors but instead are errors in program logic that don't fail but also do not come up with the correct answer. This is caused from errors in your programming logic (more on all of these errors as we get into class and start writing our own programs).

VB Help...

There is lots of help in Visual Basic. This is largely due to the fact that the language has been around for so long and the help tools are very mature. I find the Search Help option to be the most helpful because once it is clicked it remains as a tab in the Solution Explorer and you can simply type the keyword which identifies what you are looking for and it will display all relevant topics in the output window. You can also highlight any programming keyword and hit F1 and it will take you to the help topic for that keyword. Beyond that you can also search the MSDN (Microsoft Developers Network) files that come with VB and find white papers and articles about various VB programming techniques.



Book Notes:

We have been using the **Bradley - Millspaugh** books for several years (we used them since the VB 5.0 version). We also use the Advanced VB Bradley Millspaugh text for the CIS259 class.

Before each chapter, read the summary at the end of the chapter and the table of topics in the beginning of the chapter. This will help you to focus on the important points. I will also let you know which parts of the chapter are the most important. For the first few chapters, the entire chapter is very important so read this chapter carefully.



MY Turn:

VB is RAD (Rapid Application Development)

No matter what programming language is your favorite, there is a very good chance that no matter the size of your organization, VB will have prominent role for prototyping applications that will later develop into other apps used in production. VB can quickly and efficiently create a system that can be evaluated and used to determine if a fully fledged system (maybe developed in C++) might be advantageous. Even if you never program a VB application, you may be the C++ programmer that needs to convert VB to C++ and the knowledge of VB will be a great help. Make sure that when you prepare your assignments you make them as professional looking as you can. They might come in handy later on if you have an interview and need to show a prospective employer a sample of VB (this includes screen drawings and pseudo code).

Video Lessons

Throughout the next sixteen Blackboard lessons you will be given the opportunity to review video lessons. These lessons are a combination of screen capture and audio where I walk you through a topic that has historically been difficult for students. Feedback on these video lessons has been very good and I would suggest you view each one (many a couple of times for complex topics).



Chapter Review

What are the main functions of the Visual Studio IDE?

What language did Visual Basic come from?

Will VB programs run on an Apple Macintosh?

Questions?

