COURSE OUTLINE

IST 123 Programming in Visual Basic 2010

Course Number: 2
Course Title: Programming in Visual Basic 2010
Credits: 3

Class or Lecture Work Hours: 2
Laboratory Hours: 2
Clinical or Studio Hours: N/A
Co-op, Internship: N/A
Practicum, Course Length: 15 Weeks

Performance on an Examination/Demonstration: Not Applicable
Alternate Delivery Methods: Online, Telecourse
(Placement Score (if applicable); minimum CLEP score)

Required Materials:
Textbook: Programming in Visual Basic 2010
Julia Case Bradley, Anita C Millspaugh
ISBN: 9780073517254

Material: Flash drive

Catalog Description:
Programming in Visual Basic 2010 incorporates the basic concepts of programming, problem solving, programming logic, as well as the design techniques of an object-oriented, event-driven language. VB 2010 is a fully object-oriented language, which includes inheritance and polymorphism. This course allows the programmer to learn how to deal with a visual interface while acquiring important programming skills such as creating projects with objects, decisions, loops, and data management. A high priority is given to writing applications that are easy for the user to understand and to use. Students are presented with interface design guidelines throughout the course. Visual Basic 2010 is designed to allow the programmer to develop applications that run under Windows and/or in a Web browser without the complexity generally associated with programming.

Prerequisites: IST 109
Corequisites: None

Last Revised: 01/31/13

Course Coordinator (name, email, phone extension):
Assistant Professor Queen E. Okike
okikeq@mccc.edu
Extension 3464

Available Resources: (Identify library resources relevant to the course, including books, videos, journals, electronic databases, recommended websites.)

Online Learning Center:
www.mhhe.com/vb2010

Learning Center Resources: (Are there tutors for the discipline? Study groups?)
Course Goals. **The student will be able to:**

- Explain OOP concepts of objects, properties, methods, and events; the elements of debugging and using the Help system.
- Demonstrates techniques for good program design, including making the interface easy for users as well as guidelines for designing maintainable programs. Several controls are introduced, including text boxes, group boxes, check boxes, radio buttons, and picture boxes
- Apply standards to indicate the data type and scope of variables and constants and always to use Option Strict, which forces adherence to strong data typing. Error handling is accomplished using structured exception handling.
- Incorporate list boxes and combo boxes into projects; discuss looping procedures and printing lists of information. Apply single- and multidimension arrays, table lookups, and arrays of structures
- Design and develop simple Web applications that consist of Web pages that execute in a browser application.
- Create binding sources, table adapters, and datasets. Programs include accessing data from both Windows Forms and Web Forms. Also bind data tables to a data grid and bind individual data fields to controls such as labels and text boxes.
- Explain inheritance covered for visual objects (forms) and for extending existing classes. Apply Graphics, Animation, Sound, and Drag-and-Drop to projects.
- Create applications using multiple document interfaces (MDI), create toolbars, menu bars and status bars using the new ToolStrip and StatusStrip controls, and add Web content to a Windows form using the WebBrowser control. Also apply the new code snippet feature.

**COURSE CONTENT**

Unit I Learning Objectives: Introduction to Visual Basic 2010

**The student will be able to:**

- Describe the process of visual program design and development.
- Explain the term object-oriented programming.
- Explain the concepts of classes, objects, properties, methods, and events.
- List and describe the three steps for writing a Visual Basic project.
- Describe the various files that make up a Visual Basic project.
- Identify the elements in the Visual Studio environment.
- Define design time, run time, and break time.
- Write, run, save, print, and modify Visual Basic project.
- Identify syntax errors, run-time errors, and logic errors.
- Look up Visual Basic topics in Help.

Unit I Learning Objectives: More Controls

The student will be able to:
- Use text boxes, group boxes, check boxes, radio buttons, and picture boxes effectively.
- Set the BorderStyle property to make controls appear flat or three-dimensional.
- Select multiple controls and move them, align them, and set common properties.
- Make your projects easy for the user to understand and operate by defining access keys, setting a default and a cancel button, controlling the tab sequence, resetting the focus during program execution, and causing ToolTips to appear.
- Clear the contents of text boxes and labels.
- Change text color during program execution.
- Code multiple statements for one control using the With and End With statements.
- Concatenate (join) strings of text.
- Make a control visible or invisible at run time by setting its Visible property.

Unit III Learning Objectives: Variables, Constants, and Calculations

The student will be able to:
- Distinguish between variables, constants, and controls.
- Differentiate among the various data types.
- Apply naming conventions incorporating standards and indicating scope and data type.
- Declare variables using the Dim statement.
- Select the appropriate scope for a variable.
- Convert text input to numeric values.
- Perform calculations using variables and constants.
- Format values for output using the formatting functions.
- Use Try/Catch blocks for error handling.
- Display message boxes with error messages.
- Accumulate sums and generate counts.

Unit IV Learning Objectives: Decisions and Conditions

The student will be able to:
- Use block Ifs to control the flow of logic.
- Understand and use nested Ifs.
- Read and create flowcharts indicating the logic in a selection process.
- Evaluate conditions using the relational operators.
- Combine conditions using And and Or.
- Test the Checked property of radio buttons and check boxes.
- Perform validation on numeric fields.
- Call event procedures from other procedures.
- Create message boxes with multiple buttons and choose alternate actions based on the user response.
- Debug projects using breakpoints, stepping program execution, and displaying intermediate results.
Unit V Learning Objectives: Menus, Common Dialog boxes, Sub Procedures, and Functions

The student will be able to:

 Create menus and submenus for program control.
 Display and use the Windows common dialog boxes.
 Write reusable code in sub procedures and function procedures and call the procedures from other locations.

Unit VI Learning Objectives: Multiform Projects

The student will be able to:

 Add splash forms and about forms on a project.
 Present summary data on a separate form.
 Use Friend keyword.

Unit VII Learning Objectives: Lists, Loops, and Printing

The student will be able to:

 Create and use list boxes and combo boxes.
 Differentiate among the available types of combo boxes.
 Enter items into list boxes using the Items Collection in the Properties window.
 Add and remove items in a list at run time.
 Determine which item in a list is selected.
 Use the Items.Count property to determine the number of items in a list.
 Display a selected item from a list.
 Use Do/Loops and For/Next statements to iterate through a loop.
 Send information to the printer or the Print Preview window using the PrintDocument class.

Unit VIII Learning Objectives: Arrays

The student will be able to:

 Use the Select Case structure for multiple decisions.
 Make one event procedure handle multiple controls and access the name of the control that caused the event.
 Establish an array and refer to individual elements in the array with subscripts.
 Use the For Each/Next to traverse the array.
 Create a structure for multiple fields of related data.
 Accumulate totals using arrays.
 Distinguish between direct access and indirect access of a table.
 Write a table lookup for matching an array element.
 Combine the advantages of list box controls with arrays.
 Store and look up data in multidimensional arrays.

Unit IX Learning Objectives: Programming with Web Forms

The student will be able to:

 Explain the functions of the server and the client in Web programming.
 Create a Web form and run it in a browser.
 Describe the differences among the various types of Web controls and the relationship of Web controls to controls used on window interfaces.
 Understand the event structure required for Web programs.
 Design a Web Form using either a grid layout or a flow layout.
- Validate Web input using the validator controls.
- Define ASP, XML, WSDL, and SOAP

Unit X Learning Objectives: Accessing Database Files

The student will be able to:
- Use database terminology correctly.
- Create Windows and Web projects to display database data.
- Display data in a DataGrid control.
- Bind data to text boxes and labels.
- Navigate through the records in a dataset.
- Display the record number.
- Create a parameterized query.
- Allow the user to select from a combo box or list box and display the corresponding record in data-bound controls

Unit XI Learning Objectives: Saving Data and Objects in Files

The student will be able to:
- Store and retrieve data in files using streams.
- Save the values from a list box and reload for the next program run.
- Check for the end of file.
- Test whether a file exists.
- Display the standard Open File dialog box to allow the user to choose the file.
- Use serialization to store and retrieve objects.

Unit XII OOP – Creating Object-Oriented Programs

The student will be able to:
- Use object-oriented terminology correctly.
- Create a two-tier application that separates the user interface from the business logic.
- Differentiate between a class and an object.
- Create a class that has properties and methods.
- Use property procedures to set and retrieve private properties of a class.
- Declare object variables and assign values to the properties with a constructor or property procedures.
- Instantiate an object in a project using your class.
- Differentiate between shared members and instance members.
- Understand the purpose of the constructor and destructor methods.
- Inherit a new class from your own class.
- Apply visual inheritance by deriving a form from another form

Unit XIII Learning Objectives: Graphics in Windows and the Web

The student will be able to:
- Use graphics methods to draw shapes, lines, and filled shapes.
- Create a drawing surface with a Graphics object.
- Instantiate Pen and Brush objects as needed for drawing.
- Create animation by changing pictures at run time.
- Create simple animation by moving images.
- Use the Timer control to automate animation.
- Use scroll bars to move an image.
- Draw a pie chart using the methods of the Graphics object
Unit XIV Learning Objectives: Advanced Topics in Visual Basic

The student will be able to:

- Create a multiple document project with parent and child forms.
- Arrange the child forms vertically, horizontally, or cascaded.
- Store images in an image list.
- Add toolbars and status bars to your forms.
- Use calendar controls and date functions.
- Create data reports using Crystal Reports.

Evaluation of Student Learning

<table>
<thead>
<tr>
<th>Evaluation Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes</td>
<td>10%</td>
</tr>
<tr>
<td>Projects and laboratory assignments</td>
<td>50%</td>
</tr>
<tr>
<td>Units of Tests: Module One (Units I, II, III, IV)</td>
<td>10%</td>
</tr>
<tr>
<td>Module Two (Units V, VI, VII, VIII)</td>
<td>10%</td>
</tr>
<tr>
<td>Module Three (Units IX, X, XI)</td>
<td>10%</td>
</tr>
<tr>
<td>Module Four (Units XII, XIII, XIV)</td>
<td>10%</td>
</tr>
</tbody>
</table>

Total ........................................................................ 100%
## Grade Policy

<table>
<thead>
<tr>
<th>Grade</th>
<th>Definition</th>
<th>Nominal %</th>
<th>QPA quality point value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Superior Achievement</td>
<td>93-100</td>
<td>4</td>
</tr>
<tr>
<td>A-</td>
<td></td>
<td>90-92</td>
<td>3.7</td>
</tr>
<tr>
<td>B+</td>
<td></td>
<td>87-89</td>
<td>3.4</td>
</tr>
<tr>
<td>B</td>
<td>Above Average Achievement</td>
<td>83-86</td>
<td>3</td>
</tr>
<tr>
<td>B-</td>
<td></td>
<td>80-82</td>
<td>2.7</td>
</tr>
<tr>
<td>C+</td>
<td></td>
<td>77-79</td>
<td>2.4</td>
</tr>
<tr>
<td>C</td>
<td>Average Achievement</td>
<td>70-76</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>Minimally Passing</td>
<td>60-69</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>Academic Failure</td>
<td>0-59</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>Audit — no evaluation</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>W</td>
<td>Withdrawal (Student-initiated)</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>WI</td>
<td>Withdrawal (Instructor-initiated)</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>WA</td>
<td>Withdrawal (Administration-initiated)</td>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Audit**: If you audit the course, you will receive an “X” grade—this cannot be changed to a letter grade at a later date.

**Withdrawal Course Requirements**: To receive a W grade for any course, a student must consult with the course instructor or an appropriate division representative and then withdraw officially before two-thirds of the course has been completed by submitting a withdrawal form to the Office of Student Records. Withdrawal after this point results in a grade other than W (usually F). At any time before two-thirds of the course has been completed, the instructor may also withdraw with a W grade any student who has been absent excessively. A student thus withdrawn will not be entitled to any refund of tuition or fees. The student may appeal this action.

**Attendance Policy** Mercer County Community College does not have a “cut system.” Students are expected to attend all classes of every course on their schedules. Only illness or serious personal matters may be considered adequate reasons for absence. It is the prerogative of the instructor to excuse absences for valid reasons, provided the student will be able to fulfill all course requirements. Student performance in classes is formally verified at each class meeting. If a student’s attendance has been infrequent or performance unsatisfactory, he or she may receive notification in the mail. At any time, the instructor may withdraw the student from class for insufficient attendance.
**Classroom Conduct Statement** It is the student’s responsibility to attend all classes. If a student misses a class meeting for any reason, he/she is responsible for all content that is covered, for announcements made, and for acquiring any materials that may have been distributed in class. It is expected that students be on time for all classes. Students who walk into class after it has begun are expected to choose seats close to where they entered the room so that they do not disrupt the class meeting. Students are expected to follow ordinary rules of courtesy during the class sessions. Engaging in private, side conversations during class time is distracting to other students and to the instructor. Leaving class early without having informed the instructor prior to class is not appropriate. Unless there is an emergency, leaving class and returning while the class is in session is not acceptable behavior. Disruptive behavior of any type, including sharpening pencils during class while someone is speaking, is not appropriate. The college welcomes all students into an environment that creates a sense of community of pride and respect; we are here to work cooperatively and to learn together.
**Academic Integrity Statement** A student who knowingly represents work of others as his/her own, uses or obtains unauthorized assistance in the execution of any academic work, or gives fraudulent assistance to another student is guilty of cheating. The penalty for violating the honor code is severe. (See Student Handbook.) Any student violating the honor code is subject to receive a failing grade for the course and will be reported to the Office of Student Affairs. If a student is unclear about whether a particular situation may constitute an honor code violation, the student should meet with the instructor to discuss the situation. It is permissible to assist classmates in general discussions of computing techniques; general advice and interaction are encouraged. Each person, however, must develop his or her own solutions to the assigned homework and laboratory exercises. Students may not “work together” on graded assignments. Such collaboration constitutes cheating, unless it is a group assignment. A student may not use or copy (by any means) another's work (or portions of it) and represent it as his/her own.