SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY

SAULT STE. MARIE, ONTARIO

Sault College

COURSE OUTLINE

COURSE TITLE:	Visual Basic - Prototyping		
CODE NO. :	CSD301	**SEMESTER:**	**5**
PROGRAM:	Computer Programmer/Analyst		
AUTHOR:	Willem de Bruyne		
DATE:	07/07/03	**PREVIOUS OUTLINE DATED:**	07/07/02
APPROVED:			
	DEAN		**DATE**
TOTAL CREDITS:	FIVE		
PREREQUISITE(S):	CSD300		
HOURS/WEEK:	FOUR		

For additional information, please contact Colin Kirkwood, Dean
School of Technology, Skill Trades and Natural Resources
(705) 759-2554, Ext. 688

I. COURSE DESCRIPTION:

This course is an extension of the CSD206 and CSD300 Visual Basic courses. Students were taught Microsoft's Visual Basic 6.0. The Computer Studies department has decided to upgrade to the Microsoft Visual Studio.Net software. The .NET platform is a complete rewrite of the development environment, in particular Visual Basic. Because of the completely new environment, the students will be brought back to an introductory level and moved swiftly along to more complex development. In doing so, the students will benefit two-fold, they will be have a strong knowledge base of the Visual Basic 6.0 environment, as well they will be learning Microsoft's cutting edge Visual Basic .NET. All learning styles will be addressed by having the students make use of manuals; lectures; small group work; online referencing, and step-by-step exercises.

The course focuses on hands-on, so there will be plenty of do-it-yourself features throughout the course.

II. LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:

1. Upgrading to .NET

   Potential Elements of the Performance:

   - Understand why Microsoft moved to the .NET platform
   - Differentiate between Visual Basic 6.0 and .NET
   - Understand Upgrading options
   - How to prepare your projects for the Upgrade
   - Understand common tasks in Visual Basic .NET
   - Resolve issues with Language

2. Introduction to Visual Basic .NET

   Potential Elements of the Performance:

   - Describe Visual Basic .NET
   - Describe the VB programming language
   - Identify each of the phases in the development cycle
   - Describe programs, programming, applications, and program development
   - Define an algorithm
   - Define objects, attributes, and methods
   - Explain object-oriented programming and design
   - Describe rapid application development
   - Identify the key components of .NET

3. The Visual Basic .NET Integrated Development Environment
   Potential Elements of the Performance:
   - Start VB .NET
   - Customize the VB .NET integrated development environment
   - Open a VB .NET project
   - Describe the basic components of the VB .NET integrated development environment
   - Run a VB .NET project
   - Set a property on a control
   - Navigate the code window
   - Modify code in an existing project
   - Save a VB .NET project
   - Print a VB .NET project's forms and code
   - Use VB .NET project help

4. Building an Application in the Visual Basic .NET Environment
   Potential Elements of the Performance:
   - Design a VB .NET application
   - Start a new VB .NET project
   - Change the size of a form
   - Change the property values of a form
   - Add controls to a form
   - Move and resize controls
   - Use the Label, TextBox, NumericUpDown, and Button controls
   - Change the property values of controls
   - Change the Name property to rename a control
   - Write the code for the Click event procedure
   - Display line numbers in the code window
   - Use control properties in a method
   - Document code with a comment header and comment statements

5. Working with Variables, Constants, Data Types and Expressions
   Potential Elements of the Performance:
   - Use the RadioButtons and GroupBox controls
   - Use the Layout toolbar to size and align controls
   - Set a default button on a form
   - Lock controls on a form
   - Declare variables and constants
   - Use variables and constants within code

- Describe VB .NET data types
- Convert between data types
- Code a form Load event procedure
- Use the Option Strict statement
- Use arithmetic expressions
- Describe the order of operation in code
- Use the Pmt function
- Use the Format$ function

6. Decision Making

<u>Potential Elements of the Performance</u>:

- Use the ComboBox control
- Code an If..Then…Else statement
- Code a nested If..Then…Else statement
- Code a Select Case statement
- Validate user input
- Use the MessageBox class
- Use string concatenation in code
- Use relational operators in code
- Use logical operators in code

7. Repetition and Multiple Forms

<u>Potential Elements of the Performance</u>:

- Add additional forms to a project
- Change the default icon on the title bar of a form
- Use the ListView controls to display a list of items on a form
- Use the CheckBox controls
- Use the Anhor property
- Work with Collections in code
- Code a variety of looping structures
- Code a concatenation operator
- Code a keyboard event
- Code a form Resize event procedure
- Work with multiple code windows
- Specify a Startup object for a project

8. Using Menus, Common Dialogs, Procedures, Functions, and Arrays
   Potential Elements of the Performance:
   - Add a menu bar and menus to an application using the MainMenu control
   - Add a shortcut menu to an application using the ContextMenu control
   - Use the StatusBar control in an application
   - Use the PictureBox control
   - Use common dialog boxes in an application to interact with the user
   - Use one-dimensional and multidimensional arrays in code
   - Write a function procedure to return a value
   - Write sub procedures
   - Write code to pass arguments to function and sub procedures
   - Describe the two methods for passing arguments
   - Create an enhanced message box
   - Use a collection of controls to access properties and methods in code

9. Debugging, Creating Executable Files, and Distributing a Windows application
   Potential Elements of the Performance:
   - Describe the difference between syntax, logic, and run-time errors
   - Find syntax errors during design time
   - Find run-time errors and logic errors during run time
   - Set breakpoints
   - Use stepping
   - Evaluate variables using DataTips
   - Use Autos, Locals, and Me windows
   - Use the Command window
   - Create an executable file
   - Create a setup program to distribute a Windows application

10. Accessing Databases with ADO.NET, Handling Exceptions
    Potential Elements of the Performance:
    - Use the MonthCalendar control in an application
    - Use the PrintDialog control
    - Use the PrintDocument control
    - Write code in a class
    - Understand database connections

- Write code to connect to a database
- Understand how the CLR handles exceptions during run time
- Code a Try…Catch...Finally statement to handle exceptions
- Read records from a database
- Understand the role of SQL in interacting with a database
- Use simple SQL SELECT statements to query a database

11. Creating Web Applications and Writing Data to a Database
    Potential Elements of the Performance:
    - Declare a public procedure in a module
    - Use optional parameters in a procedure
    - Use SQL SELECT statements to read individual records from a database
    - Start a new ASP.NET Web application
    - Understanding Web applications and Web forms
    - Explain when to use HTML controls and Web controls in a Web application
    - Use table and Horizontal Rule HTML controls on a Web form
    - View HTML code for a Web form
    - Use Label, TextBox, and Button Web controls on a Web form
    - Test a Web application
    - Understand how ASP.NET, the CLR, and Microsoft Internet Information Services interact to run a Web application

III. TOPICS:

1. Upgrading to .NET
2. An Introduction to Visual Basic .NET and Program Design
3. Building an Application in the Visual Basic .NET Environment
4. Working with Variables, Constants, Data Types and Expressions
5. Decision Making
6. Repetition and Multiple Forms
7. Using Menus, Common Dialogs, Procedures, Functions, and Arrays
8. Debugging, Creating Executable Files, and Distributing a Windows application
9. The Visual Basic .NET Integrated Development Environment
10. Accessing Databases with ADO.NET, Handling Exceptions
11. Creating Web Applications and Writing Data to a Database

IV. REQUIRED RESOURCES/TEXTS/MATERIALS:

MICROSOFT Visual Basic .NET Comprehensive Concepts and Techniques *Authors: Shelly, Cashman, Quasney*

Instructor Handouts/e-mail attachments/web sites

V. EVALUATION PROCESS/GRADING SYSTEM:

The mark for this course will be arrived at as follows:

Quizzes 3 @ 20%	
Assign 4 @ 10%	
	100%

The following semester grades will be assigned to students in postsecondary courses:

	Grade	**Definition**	*Grade Point Equivalent*
	A+	90 – 100%	4.00
	A	80 – 89%	
	B	70 - 79%	3.00
	C	60 - 69%	2.00
	D	50 – 59%	1.00
	F (Fail)	49% and below	0.00
	CR (Credit)	Credit for diploma requirements has been awarded.	
	S	Satisfactory achievement in field /clinical placement or non-graded subject area.	
	U	Unsatisfactory achievement in field/clinical placement or non-graded subject area.	
	X	A temporary grade limited to situations with extenuating circumstances giving a student additional time to complete the requirements for a course.	
	NR	Grade not reported to Registrar's office.	
	W	Student has withdrawn from the course without academic penalty.	

VI. SPECIAL NOTES:

Special Needs:
If you are a student with special needs (e.g. physical limitations, visual impairments, hearing impairments, or learning disabilities), you are encouraged to discuss required accommodations with your instructor and/or the Special Needs office. Visit Room E1204 or call Extension 493 so that support services can be arranged for you.

Retention of course outlines:
It is the responsibility of the student to retain all course outlines for possible future use in acquiring advanced standing at other postsecondary institutions.

Plagiarism:
Students should refer to the definition of “academic dishonesty” in *Student Rights and Responsibilities*. Students who engage in “academic dishonesty” will receive an automatic failure for that submission and/or such other penalty, up to and including expulsion from the course/program, as may be decided by the professor/dean. In order to protect students from inadvertent plagiarism, to protect the copyright of the material referenced, and to credit the author of the material, it is the policy of the department to employ a documentation format for referencing source material.

Course outline amendments:
The Professor reserves the right to change the information contained in this course outline depending on the needs of the learner and the availability of resources.

Substitute course information is available in the Registrar's office.

Other Special Notes:

- Students will receive a grade of zero for late assignments unless prior permission is granted from the instructor.

- Students are expected to attend classes on a regular bases and treat their peers and instructors in a business like manner.

- Students are expected to inform the instructor via phone or e-mail if they are unable to attend class. Failure to comply will result in a 2% penalty for each occurrence.

- Students missing a test will receive a grade of zero unless prior permission is granted from the instructor

VII. PRIOR LEARNING ASSESSMENT:

Students who wish to apply for advanced credit in the course should consult the professor. Credit for prior learning will be given upon successful completion of a challenge exam or portfolio.

VIII. DIRECT CREDIT TRANSFERS:

Students who wish to apply for direct credit transfer (advanced standing) should obtain a direct credit transfer form from the Dean’s secretary. Students will be required to provide a transcript and course outline related to the course in question.