COURSE OUTLINE

Visual Basic Programming I
Course Title

IT 125
Dept. & Course No.

I. COURSE DESCRIPTION

This course utilizes Visual Basic to introduce program development for business applications. It will emphasize structured programming principles, including internal and external program documentation. Implementation of objects and event driven code will also be emphasized.

II. SEMESTER CREDITS: 3

III. CONTACT HOURS PER WEEK: 2 Lecture | 3 Lab | 5 Total

IV. PREREQUISITE: IT 105 and IT 110

V. STUDENT LEARNING OUTCOMES:

Upon completion of this course the student will be able, with 65% level of accuracy, to:

1. Discuss the history of Visual Basic.
2. Recognize and identify the different components of the Visual Basic environment.
3. Discuss and perform mathematical calculations is Visual Basic.

VI. COURSE CONTENT:

A. Discussing the history of Visual Basic.
   1. Discussing the history of Visual Basic programming language.
   2. Discussing the features of the language.

B. Recognizing and identifying the components of the Visual Basic programming environment.
   1. Opening the Visual Basic programming environment.
   2. Opening an existing program.
   3. Compiling the program.
   4. Analyzing the result / output.
   5. Running the executable.
   6. Closing / Exiting the application.

C. Discussing and performing mathematical calculations in Visual Basic.
   1. Using addition and assignment operators.
   2. Using text boxes and the Val function.

3. Using the subtraction operator.
4. Using unary minus.
5. Using the multiplication and division operators.
6. Performing integer division and using modulus.

4. Discuss the order of operation and error handling.

D. Discussing the order of operation and error handling in Visual Basic.
   1. Discussing the order of operation.
   2. Commenting and documenting programs for easy maintenance.
   3. Handling run-time errors.
   4. Creating warnings and other messages using the MsgBox feature.
   5. Controlling program flow around the error handler.

5. Define or declare and initialize variables.

E. Define or declare and initialize variables.
   1. Defining or declaring variables.
   2. Assigning appropriate names to variables.
   3. Assigning appropriate data type to variables.
   4. Initializing variables.

6. Discuss and apply different data types to variables.

F. Discussing and applying different data types.
   1. Exploring different types of data types.
   2. Identifying best data type to use with a variable.
   3. Declaring variables with appropriate data type.
   4. Assigning values to variables.

7. Discuss and integrate decision structures.

G. Discussing and integrating decisions structures to program code.
   1. Exploring different types of decision structures.
   2. Using decision structures.
   3. Using Nested decision structures.
   4. Using conditional and logical operators for comparison.

8. Discuss and integrate Looping structures.

H. Discussing and integrating Looping structures to program code.
   1. Discussing different types of loops.
   2. Using Do Loops.
   3. Using Nested Do Loops.
   4. Using For Next Loops.
   5. Using Nested For Next Loops.
   6. Breaking Loops.
   7. Incrementing or decrementing values within a loop.

9. Plan, design, diagram, and create a new Visual Basic program.

I. Planning, designing, diagramming, and creating a syntactically and functionally correct Visual Basic program.
   1. Listing what the program needs to produce the correct result.
   2. Composing a diagram of the program.
   3. Generating a flowchart of the program.
   4. Opening the Visual Basic environment.
   5. Starting a new program.
   6. Creating all the objects needed.
   7. Creating the needed functions and assigning them to the appropriate objects.
   8. Documenting the program for easy maintenance.
   9. Compiling the program.
   10. Analyzing the result or output of the compilation.
   11. Making any modifications to the program if needed.
   12. Recompiling the program if needed.
   13. Saving and running the executable.
   14. Closing / Exiting Visual Basic.

VII. Equipment and Materials

A. Student computer with Windows OS, Microsoft Word, and the Visual Basic Programming environment
B. Projector
C. Routine classroom materials
D. 1 USB storage device (at least 1GB)—student-furnished

VIII. Text

A. Required Text:
Knowlton, T. & Collings, S.. Microsoft Visual Basic BASICS. Cincinnati, OH: South-Western Educational Publishing, 2000.

B. Supplementary References: handouts

IX. Methods of Instruction

A. Lecture
B. Demonstration
C. Hands on Experience
D. Questions and Answers (Discussion)

X. Method of Evaluation

A. Description **Points**

Program Assignments	30%
Quizzes / Exercises	15%
Chapter Tests	15%
Midterm Exam / Project	20%
Final Exam / Project	20%

Total----------100%

B. Transmutation of percent to letter grade

90-100----------A
80-89 ---------- B
70-79---------- C
65-69----------D
0-64---------- F

TASK LISTING SHEET

IT 125 Visual Basic Programming I	Credits:	2	1	48
Course No. & Title		Lecture	Lab	Total Lab Hrs.

Task	Time

SLO #2.....5 hours

1. Open the Visual Basic environment.
2. Open an existing program.
3. Identify the different components of the program (e.g. IF).
4. Analyze the different components of the program and think of possible outcome of the statements (e.g. IF).
5. Compile the program.
6. Analyze the compilation result.
7. Test the program.
8. Analyze the test result.
9. Save the program.
10. Close the program and the Visual Basic Environment.

SLO #3 & 4.....5 hours

1. Plan a new Visual Basic program.
2. Diagram the program and all necessary components.
3. Write the pseudocode for the program.
4. Identify locations to use mathematical operations.
5. Analyze problem and choose the best operators to use.
6. Consider the order of operation.
7. Test the pseudocode and write down possible results for the operation.
8. Indicate where errors may occur.
9. Determine whether or not to send the user a message if an error occurs.
10. Determine in what form the messages are going to be sent.
11. Identify all objects needed to create the messages.
12. Write statements to handle the errors.
13. Save pseudocode for implementation later.

SLO #5 & 6.....5 hours

1. Identify all variables needed for the program.
2. Identify the data type for the variables.
3. Write down possible variable names based on good programming guidelines.
4. If numeric variables, identify whether or not the values are decimals.
5. Control the decimal places.
6. If string variables, identify whether or not to immediately assign value to the string.
7. Concatenate strings if necessary.
8. Include the variables in the pseudocode.
9. Save the pseudocode for implementation later.

SLO #7.....8 hours

1. Identify in the program where IF…ELSE statements are necessary.
2. Determine which Boolean expression (or logic test) to use with the IF statement.
3. Indicate whether or not a nested IF statements are necessary.
4. Based on the logic test, determine whether to increment or decrement values to terminate the IF statements.
5. Make sure that the IF statements are not infinite.
6. Write down the IF statements and test the pseudocode.
7. Write down the results of the IF statements and values for the decrementing or incrementing variables.
8. Make any necessary modifications to the statements.
9. Save the pseudocode for implementation later.

SLO #8.....8 hours

1. Identify in the program where LOOPING statements are necessary.
2. Determine the purpose of the loop.
3. Indicate whether or not nested loop statements are necessary.
4. Determine the logic test to use with the loop.
5. Indicate when and what will terminate the loop.
6. Make sure the loop in not infinite.
7. Write down the looping statements and test the pseudocode.
8. Write down the results of the looping statements.
9. Make any necessary modifications to the statements.
10. Save the pseudocode for implementation later.

SLO #9.....17 hours

1. Plan a program.
2. Indicate all the necessary components for the program.
3. Diagram the program.
4. Pseudocode the program based on the diagram.
5. Name all variables based on the good programming guidelines.
6. Identify the data type for all the variables.
7. Assign values to variables if necessary.
8. Concatenate string variables if necessary.
9. Determine what operators to use with any mathematical calculations.
10. Identify areas where errors may occur.
11. Write down possible psuedocode for error handling.
12. Indicate where IF…ELSE statements are necessary.
13. Identify the logic expression(s) to use with the statements.
14. Write the pseudocode for the IF…ELSE statements.
15. Indicate where LOOPING statements are necessary.
16. Identify the logic expression(s) to use with the statements.
17. Write the pseudocode for the LOOPING statements.
18. Save the pseudocode.
19. Open the Visual Basic environment.
20. Begin a new program.
21. Translate the pseudocode to Visual Basic following the correct syntax.

22. Document or comment areas of the program for easy maintenance.
23. Save the program.
24. Compile the program.
25. Run the program.
26. Analyze the result and make any necessary modifications.
27. Save the program.
28. Close the program.
29. Close the Visual Basic environment.

TOTAL.....**48 hours**

Palau Community College
IT 125-Visual Basic Programming I
Course Learning Outcomes

During the course experience, the ***Course Learning Outcomes*** (CLOs) will be assessed through the use of signature assignments. A rating scale will be used to determine the students' proficiency level of each CLO using specifically aligned assignments. The numerical ratings of 4, 3, 2 and 1 are not intended to represent the traditional school grading system of A, B, C, D and F. The descriptions associated with each of the numbers focus on the level of student performance for each of the course learning outcomes listed below.

Rating Scale: **4----------Exceeds Expectations**
3----------Meets Expectations
2----------Developing
1----------Below Expectations

CLO #1:

Numerical Value	**Design and develop a computer program by identifying and defining all needed variables.**
4	Perform all of the following tasks accurately and completely: • Identify and define all needed variables. • Name variables correctly based on the programming "rules of thumb" for naming variables. • Assigning appropriate data type to variables. • Assigning appropriate values to variables correctly.
3	Perform the tasks mentioned above with mixed quality, but most are adequate and complete.
2	Perform the tasks mentioned above with mixed quality, but most are inadequate or incomplete.
1	Perform the tasks mentioned above inaccurately or incompletely.

CLO #2:

Numerical Value	**Design and develop a computer program by identifying areas where decision structures are necessary and developing the structures**
4	Perform all of the following tasks accurately and completely: • Identify areas where decision structures are necessary. • Select the best type of decision structure to use. • Identify the logical and comparison operators needed by the selected decision structure. • Generate logically correct decision structures.
3	Perform the tasks mentioned above with mixed quality, but most are adequate and complete.
2	Perform the tasks mentioned above with mixed quality, but most are inadequate or incomplete.
1	Perform the tasks mentioned above inaccurately or incompletely.

CLO #3:

Numerical Value	**Design and develop a computer program by identifying areas where LOOPING structures are necessary and developing the structures.**
4	Perform all of the following tasks accurately and completely: • Identify areas where LOOPING structures are necessary. • Select the best LOOPING structure to use (single DO loop, nested DO loop, single FOR NEXT loop, nested FOR NEXT loop, single WHILE loop, or nested WHILE loop). • Identify the logical and comparison operators needed by the selected LOOPING structure. • Generate logically correct LOOPING structure.
3	Perform the tasks mentioned above with mixed quality, but most are adequate and complete.
2	Perform the tasks mentioned above with mixed quality, but most are inadequate or incomplete.
1	Perform the tasks mentioned above inaccurately or incompletely.

CLO #4:

Numerical Value	**Plan and design a computer program by identifying areas where other programming statements are necessary and developing the statements.**
4	Perform all of the following tasks accurately and completely: • Identify areas where data storage structures are necessary and generate structures correctly. • Identify areas where other calculation statements are necessary and generate the statements correctly. • Identify areas where error handling statements are necessary and generate the statements correctly. • Identify areas where terminating statements are necessary and generate the statements correctly.
3	Perform the tasks mentioned above with mixed quality, but most are adequate and complete.
2	Perform the tasks mentioned above with mixed quality, but most are inadequate or incomplete.
1	Perform the tasks mentioned above inaccurately or incompletely.

CLO #5:

Numerical Value	**Develop a syntactically and functionally correct computer program by implementing the selected solution that consists of all the procedures and components necessary to make the program run and function correctly and accurately.**
4	Perform all of the following tasks accurately and completely: • Develop syntactically correct statements for naming, defining, and assigning values to variables. • Develop syntactically correct decision structures where necessary. • Develop syntactically correct LOOPING structures where necessary. • Develop syntactically correct statements for other features such as error handling and error trapping structures to increase program efficiency.
3	Perform the tasks mentioned above with mixed quality, but most are adequate and complete.
2	Perform the tasks mentioned above with mixed quality, but most are inadequate or incomplete.
1	Perform the tasks mentioned above inaccurately or incompletely.