

3. Plan and design programs.
 - C. Planning, designing, and diagramming programs to solve real world situations.
 1. Selecting all essential functions that the program requires in order to perform correctly.
 2. Selecting all necessary identifiers or variables.
 3. Identifying possible variable names based on good programming practice and the data type.
 4. Sketching the program interface and all vital components.
 5. Diagramming the program's implementation process and all possible outcomes.
4. Declare and use variables and constants correctly.
 - D. Declaring and using variables and constants correctly in a program.
 1. Developing variables with appropriate names.
 2. Assigning values to variables.
 3. Selecting the scope of a variable.
5. Declare and use arrays.
 - E. Declaring and using Arrays correctly in a program.
 1. Developing arrays and assigning appropriate data type.
 2. Determining whether the array is one-dimensional or multi-dimensional.
 3. Identifying what information will be stored in the array.
 4. Identifying what additional statements or structures are needed to load and also to read information from the arrays.
6. Discuss, plan, and utilize selection structures.
 - F. Discussing, planning, and utilizing selection structures.
 1. Selecting the proper Selection structure to use.
 2. Identifying logical operators to use.
 3. Generating pseudocode for the Selection structure.
 4. Sketching a flowchart demonstrating the execution of the statements.

7. Discuss, design, and utilize repetition structures.
 - G. Discussing, designing, and utilizing Repetition structures.
 1. Selecting the proper repetition structure to use.
 2. Determining whether the repetition structure is used with an array, if so, which array and what is it used for.
 3. Generating pseudocode for the repetition structure.
 4. Sketching a flowchart or table demonstrating the execution and outputs of the statements.
8. Discuss, plan, and utilize sequential access data files.
 - H. Discussing, planning, and utilizing sequential access data files.
 1. Identifying scenarios in which sequential access files are most appropriate to use.
 2. Creating, opening, and closing sequential access files.
 3. Writing records to a sequential access file.
 4. Reading records from a sequential access file.
9. Discuss, develop, and utilize error handling and trapping statements.
 - I. Discussing, developing, and utilizing error handling and trapping statements.
 1. Identifying areas to insert error handling and trapping statements.
 2. Determining whether or not to use dialog boxes for error messages.
10. Discuss and utilize other programming languages.
 - J. Discussing and utilizing other computer programming languages.
 1. Discussing other commonly used programming languages.
 2. Comparing other commonly used programming languages to Visual Basic.
 3. Utilizing other programming languages to write programs.

VII. Equipment and Materials

- A. Student computer with Microsoft Word and software development applications.
- B. Projector
- C. Routine classroom materials
- D. 1 USB storage device (at least 1GB)—student-furnished

VIII. Text

- A. Required Text:
Loffelmann, Klaus & Purohit, Sarika Calla. Microsoft Visual Basic 2010 Developer's Handbook. O'Reilley Media, Inc., 2011.
- 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
Programming Assignments	40%
Quizzes / Exercises	15%
Chapter Tests	15%
Midterm Exam / Project	15%
Final Exam / Project	15%

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 205 Visual Basic Programming II
Course No. & Title

Credits: 2 1 48
Lecture Lab Total Lab Hrs.

Task

Time

SLO #3.....10 hours

1. Open Microsoft Word or other word processing software.
2. Analyze the given situation and list all necessary components of the program.
3. List all necessary identifiers and variables to use.
4. List possible identifier and variable names following the good programming practice guidelines.
5. Sketch a user interface to implement.
6. List all controls and their functions.
7. Draw a flowchart representing the data flow.
8. Pseudocode the program for future implementation.
9. Save the document.
10. Print the document.
11. Exit Microsoft Word or the word processing software used.
12. Review the plan, design, and diagram of the program.
13. Review the program's pseudocode.
14. Open the Visual Basic programming environment.
15. Begin a new Visual Basic program.
16. Create a new form and layout the programs interface.
17. Insert labels in the form.
18. Insert textboxes in the form.
19. Insert radio buttons and other controls in the form.
20. Save the form.
21. Close / Exit the Visual Basic programming environment.

SLO #44 hours

1. Open existing program.
2. View the program code.
3. Create variables to be used in the program.
4. Identify the appropriate scope for variables.
5. Declare variables within functions.
6. Declare global variables.
7. Assign appropriate names to variables.
8. Assign appropriate data types to variables.
9. Assign values to variables.
10. Save the program.
11. Close / Exit the Visual Basic programming environment.

SLO #54 hours

1. Open existing program.
2. View the program code.
3. Create arrays to be used in the program.
4. Select one-dimensional or multi-dimensional arrays.
5. Identify the appropriate scope for arrays.
6. Declare arrays within functions.
7. Declare global arrays.
8. Assign appropriate names to arrays.
9. Assign appropriate data types to arrays.
10. Assign values to arrays.
11. Save the program.
12. Close / Exit the Visual Basic programming environment.

SLO #66 hours

1. Open existing program.
2. View the program code.
3. Identify areas in the program that require selection structures.
4. Determine what selection structure to use.
5. Select IF... THEN statements.
6. Select IF... THEN... ELSE statements.
7. Choose SELECT CASE statements.
8. Diagram the output for the selection statements.
9. Save the program.
10. Close / Exit the Visual Basic programming environment.
11. Open the program's flowchart.
12. Include the selection statements in the program's flowchart.
13. Save the flowchart.
14. Close / Exit the flowchart.

SLO #76 hours

1. Open existing program.
2. View the program code.
3. Identify areas in the program that require repetition structures.
4. Determine what repetition structure to use.
5. Select FOR... NEXT statements.
6. Select WHILE statements.
7. Select DO WHILE statements.
8. Select DO UNTIL statements.
9. Insert loop-terminating statements.
10. Diagram the output for the repetition statements.
11. Save the program.
12. Close / Exit the Visual Basic programming environment.
13. Open the program's flowchart.
14. Include the repetition statements in the program's flowchart.
15. Save the flowchart.
16. Close / Exit the flowchart.

SLO #8.....4 hours

1. Open existing program.
2. View the program code.
3. Identify areas in the program to use sequential access files.
4. Create sequential access files.
5. Write data to sequential access files.
6. Read data from sequential access files.
7. Save and close sequential access files.
8. Save the program.
9. Close / Exit the Visual Basic programming environment.
10. Open the program's flowchart.
11. Include the function of the sequential access files in the program's flowchart.
12. Save the flowchart.
13. Close / Exit the flowchart.

SLO #9.....4 hours

1. Open existing program.
2. View the program code.
3. Identify areas in the program to include error handling and trapping statements.
4. Apply the CancelError property.
5. Include On Error Go To statements.
6. Test the error handling and trapping statements.
7. Make changes if necessary.
8. Save the program.
9. Close / Exit the Visual Basic programming environment.
10. Open the program's flowchart.
11. Include the error handling and trapping statements in the program's flowchart.
12. Save the flowchart.
13. Close / Exit the flowchart.

SLO #10.....10 hours

1. Install and configure computer to support programming in other high level languages.
2. Write computer programs in other high level programming languages.
3. Compile/Interpret the completed program.
4. Run/Execute the program.
5. Debug the program.
6. Test the program.

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

Palau Community College
IT 205-Visual Basic Programming II
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	Plan, design, and develop a syntactically and functionally correct programs that utilizes the decision or selection structure (e.g. IF...ELSE statements).
4	Perform all of the following tasks accurately and completely: <ul style="list-style-type: none"> • Identify the programming problem or the given task, consider all possible solutions, and choose the best approach to the problem. • Pseudocode the solution to the problem. Identify areas where decision statements are necessary, select the best decision statements to use, identify the logical or Boolean expression necessary to make the decision statements work, and generate logically correct decision statements. • Determine which programming language to use and translate the pseudocode into that language. • Test the program, analyze the results, and make corrections if necessary
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	Plan, design, and develop a syntactically and functionally correct programs that utilizes the looping, repetition, or iteration structure (e.g. DO WHILE statements).
4	Perform all of the following tasks accurately and completely: <ul style="list-style-type: none"> • Identify the programming problem or the given task, consider all possible solutions, and choose the best approach to the problem. • Pseudocode the solution to the problem. Identify areas where looping statements are necessary, select the best looping statements to use, identify the variables that will control the loop and the logical or Boolean expression necessary to make the looping statements work, and generate logically correct looping statements. • Determine which programming language to use and translate the pseudocode into that language. • Test the program, analyze the results, and make corrections if necessary
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	Plan, design, and develop a syntactically and functionally correct programs that utilizes arrays.
4	Perform all of the following tasks accurately and completely: <ul style="list-style-type: none"> • Identify the programming problem or the given task, consider all possible solutions, and choose the best approach to the problem. • Pseudocode the solution to the problem. Define the array and identify structures necessary to fill the array with values, perform array calculation, and/or print array values. Identify the variables necessary to make the array statements work, and generate logically correct statements. • Determine which programming language to use and translate the pseudocode into that language. • Test the program, analyze the results, and make corrections if necessary
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, design, and develop a syntactically and functionally correct programs that utilizes sort methods.
4	Perform all of the following tasks accurately and completely: <ul style="list-style-type: none"> • Identify the programming problem or the given task, consider all possible solutions, and choose the best approach to the problem. • Pseudocode the solution to the problem. Identify structures necessary to make the program work. Determine which sort method to use and pseudocode the algorithm for the selected method. • Determine which programming language to use and translate the pseudocode into that language. • Test the program, analyze the results, and make corrections if necessary
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	Plan, design, and develop a syntactically and functionally correct programs that connects to, retrieves values from, updates values in, and writes values to an underlying database.
4	Perform all of the following tasks accurately and completely: <ul style="list-style-type: none"> • Identify the programming problem or the given task, consider all possible solutions, and choose the best approach to the problem. • Plan, design, and develop the underlying database. • Pseudocode the solution to the problem. Identify structures necessary to make the program work. Determine which database connection interface to use and pseudocode the algorithm for the selected connection. • Determine which programming language to use and translate the pseudocode into that language. Test the program, analyze the results, and make corrections if necessary
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.