COURSE OUTLINE

Introduction to Programming
Course Title

IT 110
Dept. & Course No.

I. COURSE DESCRIPTION

This course introduces microcomputer programming, including problem solving procedures, flowcharts and program designs, debugging, and program documentation.

II. SEMESTER CREDITS: 3

III. CONTACT HOURS PER WEEK:

3	0	3
Lecture	**Lab**	**Total**

IV. PREREQUISITE : MA 105

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 computers.
2. Discuss the history of computer programming.
3. Plan, diagram, and chart computer programs.

VI. COURSE CONTENT:

A. Discussing the history of computers.
   1. Discussing the history of computers.
   2. Discussing the different types of computers.
   3. Discussing the components of a personal computer.
   4. Introducing commonly used computer terminology.

B. Discussing the history of computer programming and logic.
   1. Discussing the history of computer programming.
   2. Introducing commonly used programming terminology.

C. Planning, diagramming, and flowcharting programs.
   1. Planning a program.
   2. Diagramming the functions and controls.
   3. Flowcharting the data flow.
   4. Writing pseudocode for future implementation.

4. Document computer programs.

D. Documenting programs.
   1. Discussing the importance of documenting programs.
   2. Discussing different documenting methods.
   3. Adding documents / comments to existing pseudocode.

5. Discuss the program compilation process.

E. Compiling programs.
   1. Introducing a complier.
   2. Identifying the different components of the compiler.
   3. Recognizing the compiling process.

6. Debug computer programs.

F. Debugging programs.
   1. Analyzing compilation results.
   2. Correcting or debugging programming errors.

7. Create identifiers and assign data types.

G. Creating identifiers and assigning data types.
   1. Identifying possible use of variables.
   2. Producing acceptable variable names.
   3. Determining the data type for variables.
   4. Assigning values to variables.

8. Discuss decision structures.

H. Discussing decision structures.
   1. Identifying possible use of decision structures.
   2. Discussing Boolean expressions.
   3. Formulating syntactically correct decision structures.
   4. Analyzing results of decision structures.

9. Discuss looping structures.

I. Discussing looping structures.
   1. Identifying the possible use of looping structures.
   2. Discussing different types of structures statements.
   3. Formulating syntactically correct looping structures.
   4. Discussing infinite loops and how to prevent them.
   5. Discussing the process of terminating looping structures.

10. Introduce basic error handling.

J. Introducing Error Handling.
   1. Introducing and discussing error handling.
   2. Identifying possible use of error handling.
   3. Discussing the process of performing error handling.
   4. Discussing different types of error handling functions and components.

VII. EQUIPMENT AND MATERIALS:

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

VIII. TEXT:

A. Required Text:
   Farrell, Joyce. Programming Logic and Design, Introductory. Boston, MA: Course Technology.

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
Exercises / Assignments	40%
Quizzes	10%
Chapter Tests	15%
Midterm Exam / Project	15%
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

Palau Community College
IT 110-Introduction to Programming
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, diagram, and generate a flowchart of a possible solution to a given computer-programming assignment.
4	Perform all of the following tasks accurately and completely: • Consider and record all possible solutions and outcomes. • Compare all solutions and select the most effective one. • List all necessary components and identify the purpose of each of the components. • Sketch a flowchart of the selected solution indicating the flow of data beginning from input and ending from output.
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	Identify areas where decision structures are required and plan and develop a flowchart and pseudocode for the decision structures to produce the desired output.
4	Perform all of the following tasks accurately and completely: • Identify areas where decision structures are necessary. • Sketch possible structures for the decision structures and diagram possible results for each of the statements. • Identify Boolean expressions and logical statements to use with the structures. • Generate functionally and syntactically 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	Identify areas where LOOPING structures are required and plan and develop flowchart and pseudocode for the LOOPING structures to produce the desired output.
4	Perform all of the following tasks accurately and completely: • Identify areas where LOOPING structures are necessary and compare and select appropriate LOOPING structures to use. • Sketch possible statements for the LOOPING structures and diagram possible results for each of the statements. • Identify Boolean expressions, logical expressions, and termination statements to use with the LOOP. • Generate functionally and syntactically correct LOOPING 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 #4:

Numerical Value	**Plan and develop complete pseudocode based on the solution that consists of all the procedures and components necessary to make the program run and function correctly.**
4	Perform all of the following tasks accurately and completely: • Plan, design, and sketch a flowchart and generate the pseudocode representing data flow. • Identify and initialize all required variables and arrays. • Create syntactically and functionally correct DECISION and LOOPING structures where necessary. • Identify and apply additional features such as control breaks, error handling, and error trapping statements 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.