

Title (Units): **COMP 1180 Structured Programming (3,3,2)**

Course Aims: This course provides students with basic knowledge of computer-oriented problem solving methodologies, algorithm development, structured programming concepts and design techniques, and implementation tools that facilitate debugging and testing. In particular, structured programming skills will be illustrated with a contemporary programming language.

Upon completion, students are expected to be proficient in developing software packages for solving real-world problems, with an awareness of the importance of programming style, program testing and documentation, and program development methodologies.

Prerequisite: Nil

Learning Outcomes (LOs):

Upon successful completion of this course, students should be able to:

No.	Learning Outcomes (Los)
	Knowledge
1	Explain the importance of programming methodologies
2	Describe the elements in programming languages
3	Describe the importance of programming styles, implementation and testing
	Professional Skill
4	Analyze computational problems and implement that design in a programming language
	Transferable Skill
5	Formulate problems as steps so as to be solved systematically
	Attitude
6	Build up analytical thinking and a habit of detailed documentation

Calendar Description: This course provides students with basic knowledge of computer-oriented problem solving methodologies, algorithm development, structured programming concepts and design techniques, and implementation tools that facilitate debugging and testing. In particular, structured programming skills will be illustrated with a contemporary programming language.

This course is open to Computer Science majors, Computing Studies majors, and Physics majors with Computer Science concentration only.

Assessment:

No.	Assessment Methods	Weighting	Remarks
1	Continuous Assessment	40%	Continuous assessments are designed to measure how well students have learned the basic conception of programming methodologies, structured languages, and discipline of programming.
2	Examination	60%	Final examination questions are designed to see how far students have achieved their intended learning outcomes. Questions will primarily be analysis and skills based to assess students' awareness of programming style, testing and documentation, development methodologies, and their abilities in developing software packages for solving real-world problems.

Rubrics:

Excellent (A)	<ul style="list-style-type: none">• Achieves all six CILOs, demonstrating a good mastery of computer program development in structured programming languages• With competence, able to formulate computational problems as steps so as to be solved systematically• A good mastery of analyzing, designing and implementing computational problems in structured programming languages• A solid understanding of the elements and the programming constructs in structured programming languages• A good mastery of computer program testing, debugging, and bug fixing• Demonstrate a good and consistent practice of programming style, and a habit of detailed documentation
Good (B)	<ul style="list-style-type: none">• Achieves all six CILOs, demonstrating a good understanding and capability of computer program development in structured programming languages• Able to formulate computational problems as steps so as to be solved systematically• Able to analyze, design and implement computational problems in structured programming languages• A good understanding of the elements and the programming constructs in structured programming languages• Able to perform computer program testing, debugging, and bug fixing• Demonstrate a good practice of programming style and detailed documentation
Satisfactory (C)	<ul style="list-style-type: none">• Achieves most of the six CILOs, demonstrating an adequate level of understanding and capability of computer program development in structured programming languages• Demonstrate an acceptable level of ability in formulating computational problems as steps so as to be solved systematically• Demonstrate an acceptable level of ability in analyzing, designing and implementing computational problems in structured programming languages• A basic understanding of the elements and the programming constructs in structured programming languages• Demonstrate an acceptable level of ability in performing computer program testing, debugging, and bug fixing• Understand the importance of programming style and detailed documentation
Marginal Pass (D)	<ul style="list-style-type: none">• Achieves most of the six CILOs, with minimal level of understanding and capability of computer program development in structured programming languages• Demonstrate a minimal level of ability in formulating computational problems as steps so as to be solved systematically• Demonstrate a minimal level of ability in analyzing, designing and implementing computational problems in structured programming languages• A minimal level of understanding of the elements and the programming constructs in structured programming languages• Demonstrate a minimal level of ability in performing computer program testing, debugging, and bug fixing• A minimal level of understanding of the importance of programming style and detailed documentation
Fail (F)	<ul style="list-style-type: none">• Achieves less than three of the six CILOs, with little understanding and capability of computer program development in structured programming languages• Unable to formulate computational problems as steps so as to be solved systematically• Unable to analyze, design or implement computational problems in structured programming languages• Understanding of the elements and the programming constructs in structured programming languages below the basic minimum level• Unable to perform computer program testing, debugging, and bug fixing• Poor programming style and develop computer programs with little or no documentation

Learning Outcomes and Weighting:

Content	LO No.
I. Programming Methodologies	1,4
II. Constructs of Structured Languages	2,4,5,6
III. Discipline of Programming	3,4,6

References: H. Deitel and P. Dietel, C: How to Program, Pearson Prentice-Hall, 5th edition, 2007.
A. Kelley and I. Pohl, A Book on C: Programming in C, 4th edition, Addison Wesley, 1998.
M. Weiss, Data Structures and Algorithm Analysis in C, Addison Wesley, 3rd edition, 2006.

Course Content in Outline:

Topic

- I. Programming Methodologies
 - A. Design of algorithms
 - B. Flowcharts and pseudo codes
 - C. Top-down program design - stepwise refinement
- II. Constructs of Structured Languages
 - A. Lexical elements, data types, operators and expressions
 - B. Control structures
 - C. Composite structures such as arrays and structures
 - D. Memory concepts and address management
 - E. Functions and program structures
 - F. Call by value and call by reference
 - G. Recursion
 - H. File processing
 - I. Dynamic memory allocation, linked lists, trees
- III. Discipline of Programming
 - A. Programming style
 - B. Structured coding and program modularity
 - C. Program documentation and maintenance