Title (Units):	COMP 7740 Supplementary Programming (32 hours) (0,*,*)
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 familiar with programming skills for real-world problems, with an awareness of the importance of programming style, program testing and documentation, and program development methodologies.
Prerequisite:	Nil

## **Course Intended Learning Outcomes (CILOs):**

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

No.	Course Intended Learning Outcomes (CILOs)	
	Knowledge	
1	Explain programming methodologies and their importance	
2	Describe the elements in programming languages	
3	Describe programming styles, implementation and testing and their importance	
	Professional Skill	
4	Analyze computational problems and implement the solutions in a programming language	
	Transferable Skill	
5	Formulate problems 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 MSc in Information Technology Management students with inadequate programming background.

#### Teaching & Learning Activities (TLAs):

CILOs	TLAs will include the following:	
CILOs 1-6	Students will learn programming principles and problem solving methodologies via lectures and tutorials.	
CILOs 2-6	Tutorials and machine problems are devised so that students will implement their solution designs into their programs.	

#### Assessment:

No.	Assessment Methods	Weighting	CILOs to be addressed	Remarks	
1	Continuous Assessment	100%	1-6	Continuous assessments are designed to assess how well students have achieved the intended learning outcomes. They include the following:	

	programming assignments (30%), practical test(s) (30%), and written test(s) (40%). The overall assessment result is either "Satisfactory" or "Unsatisfactory".
--	--

## **Rubrics:**

Criteria	Excellent (A)	Good (B)	Satisfactory (C)	Fail (F)
Constructs of	The student acquires	The student	The student is able	The student is
programming	excellent knowledge	acquires sufficient	to use some	unable to describe
languages	in the language	knowledge in the	important language	any important
	structures, e.g.,	language structures,	structures, e.g.,	language structures,
	control structures, data	e.g., control	control structures,	e.g., control
	types, arrays,	structures, data	data types, arrays,	structures, data
	recursion, and so on.	types, arrays,	recursion, and so	types, arrays,
		recursion, and so	on.	recursion, and so
		on.		on.
Programming	By using flow charts	The student is <b>able</b>	The student can	The student cannot
methodologies	and pseudocodes, the	to design programs	design programs	design programs
	student is <b>highly</b>	that <b>sufficiently</b>	that <b>moderately</b>	that meet crucial
	capable of designing	meet the	meet some crucial	specifications.
	efficient programs	specifications.	specifications.	
	that meet all			
	specifications.			
Programming	The student	The student writes	The student writes	The student <b>does</b>
style	consistently writes	programs in styles	programs in styles	not demonstrate
	programs in styles that	that are <b>sufficiently</b>	that are <b>somewhat</b>	any style in the
	are highly structured	structured and can	structured and can	programs.
	and can be <b>easily</b>	be understood.	barely be	
	understood.		understood.	
Program	The programs work	The programs work	The programs work	The programs do
implementation	efficiently and meet	and meet <b>most</b> of	and meet some of	not work and do
and testing	all of the	the specifications	the specifications	not implement
and testing	specifications	Documentation can	Some	important
	Documentation is well	mostly be	documentation is	specifications No
	written	understood	included	documentation is
	wiitteil.	unacistooa.	menuucu.	written

# **Course Intended Learning Outcomes and Weighting:**

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

## **References:**

P. Deitel and H. Deitel, C: How to Program, 8<sup>th</sup> ed., Pearson, 2015.
S. G. Kochan, Programming in C, 4th ed., Addison-Wesley, 2014.

## **Course Content in Outline:**

### <u>Topic</u>

- I. Programming Methodologies
  - A. Design of algorithms
  - B. Flowcharts and pseudocodes
  - 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
  - E. Functions and program structures
  - F. Call by value and call by reference
  - G. Recursion
  - H. File processing
- III. Discipline of Programming
  - A. Programming style
  - B. Structured coding and program modularity
  - C. Program testing and debugging
  - D. Program documentation and maintenance