

Subject Description Form

Subject Code	COMP1011					
Subject Title	Programming Fundamentals					
Credit Value	3					
Level	1					
Pre-requisite / Co-requisite / Exclusion						
Objectives	<p>The objectives of this subject are to:</p> <ul style="list-style-type: none"> • provide students with knowledge on the fundamental elements in computer programming; and • introduce advanced computer programming techniques necessary for developing more sophisticated computer application programs. 					
Intended Learning Outcomes	<p>Upon completion of the subject, students will be able to:</p> <p><i>Professional/academic knowledge and skills</i></p> <p>(a) understand the programming elements for solving computing-related problems;</p> <p>(b) possess the ability to design and develop efficient computer programs for solving problems; and</p> <p>(c) possess the ability to learn other high level programming languages independently.</p> <p><i>Attributes for all-roundedness</i></p> <p>(d) develop skills in problem solving using systematic approaches;</p> <p>(e) identify and develop problem solutions in a logical manner; and</p> <p>(f) solve complex problems in groups and develop group work.</p>					
Subject Synopsis/ Indicative Syllabus	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">Topic</td> </tr> <tr> <td>1. Fundamentals of Computing. Basic concepts of computers and computing, compilation and interpretation, elementary programming constructs.</td> </tr> <tr> <td>2. Flow controls. Basic flow control: selection, repetition and functions.</td> </tr> <tr> <td>3. Data Collections. Structures, lists, sets and strings</td> </tr> <tr> <td>4. Program Design. Problem solving, problem correctness, testing and debugging</td> </tr> </table>	Topic	1. Fundamentals of Computing. Basic concepts of computers and computing, compilation and interpretation, elementary programming constructs.	2. Flow controls. Basic flow control: selection, repetition and functions.	3. Data Collections. Structures, lists, sets and strings	4. Program Design. Problem solving, problem correctness, testing and debugging
Topic						
1. Fundamentals of Computing. Basic concepts of computers and computing, compilation and interpretation, elementary programming constructs.						
2. Flow controls. Basic flow control: selection, repetition and functions.						
3. Data Collections. Structures, lists, sets and strings						
4. Program Design. Problem solving, problem correctness, testing and debugging						
Teaching/ Learning Methodology	<p>This subject emphasizes both the conceptual elements in computer programming and practical experiences. Teaching includes both lectures and hands-on Lab exercises reinforcing taught concepts. Students should attend both lectures and laboratory</p>					

	sessions. Continuous assessments help to reinforce the programming concepts and skills learned for applications.							
Assessment Methods in Alignment with Intended Learning Outcomes	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)					
			a	b	c	d	e	f
	Continuous Assessment	65%						
	1. Assignments		✓	✓			✓	✓
	2. Quizzes		✓	✓			✓	
	3. Project(s)		✓	✓	✓	✓	✓	✓
	Final Examination	35%	✓	✓	✓	✓	✓	
Total	100%							
Note: Students must pass both the continuous assessment and examination components to pass the course. The continuous assessment and the final examination will be designed to assess the specified learning outcomes. The formats may include written questions, programming exercises and quizzes.								
Student Study Effort Expected	Class contact:							
	▪ Lecture					39 Hrs.		
	▪ Lab					13 Hrs.		
	Other student study effort:							
	▪ Assignments, Quizzes, Projects, Exam					68 Hrs.		
Total student study effort					120 Hrs.			
Reading List and References	Reference Books:							
	1. Prata, Stephen, <i>C Primer Plus</i> , 6 th Edition, Addison-Wesley Professional, 2013.							
	2. King, K. N., <i>C Programming: A Modern Approach</i> , 2 nd Edition, W. W. Norton & Company, 2008.							
	3. Kernighan, B. W. and Ritchie, D. M., <i>C Programming Language</i> , 2 nd Edition, Prentice Hall, 1988.							
	4. Spraul, V. Anton, <i>Think Like a Programmer: An Introduction to Creative Problem Solving</i> , No Starch Press, 2012. (examples in this book are written in C++, but will improve your ability to think like a programmer in any language)							
5. Zelle, John, <i>Python Programming: An Introduction to Computer Science</i> , 2 nd Edition, Franklin, Beedle & Associates, 2010.								