

Subject Description Form

Subject Code	COMP1011					
Subject Title	Programming Fundamentals					
Credit Value	3					
Level	1					
Pre-requisite / Co-requisite / Exclusion	Exclusion: COMP1012, ENG2002					
Objectives	<p>The objectives of this subject are to:</p> <ol style="list-style-type: none"> 1. provide students with knowledge on the fundamental elements in computer programming; and 2. introduce basic 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> <ol style="list-style-type: none"> (a) understand the basic components of high-level programming language (e.g., C++); (b) understand the basic routine of writing programs for solving standard computational and logical problems; (c) possess the ability to learn other high-level programming languages independently; <p><i>Attributes for all-roundedness</i></p> <ol style="list-style-type: none"> (d) learn and acquire basic skills in problem-solving; and (e) identify and develop problem solutions in a logical manner. 					
Subject Synopsis/ Indicative Syllabus	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">Topic</th> </tr> </thead> <tbody> <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> </tbody> </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 emphasises 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 sessions. Continuous assessments help to reinforce the programming concepts and skills learned for applications.</p>					

Assessment Methods in Alignment with Intended Learning Outcomes	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed				
			a	b	c	d	e
	Continuous Assessment	30%					
	1. Assignments		✓	✓			✓
	2. Quizzes		✓	✓			✓
	3. Individual project(s)		✓	✓	✓	✓	✓
	Final Examination	70%	✓	✓	✓	✓	✓
	Total	100%					
	<p>Notes:</p> <p>If a student fails either the continuous assessment component or fails the final exam component, then his/her overall grade shall not exceed C-.</p> <p>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.</p>						
Student Study Effort Expected	Class contact:						
	▪ Lecture					39 Hrs.	
	▪ Lab					13 Hrs.	
	Other student study effort:						
	▪ Assignments, Quizzes, Individual projects, Exam					68 Hrs.	
	Total student study effort					120 Hrs.	
Reading List and References	Reference Books:						
	1. Deitel, H. and Deitel, P., C++ How to Program, Prentice Hall, 2024.						
	2. Stephen Prata, C Primer Plus, 6 th Edition, Addison-Wesley Professional, 2013.						
	3. V. Anton Spraul, Think Like a Programmer: An Introduction to Creative Problem Solving, No Starch Press, 2012.						
	4. K. N. King, C Programming: A Modern Approach, 2 nd Edition, W. W. Norton & Company, 2008.						