

Subject Description Form

Subject Code	COMP3021
Subject Title	Programming Language Paradigms
Credit Value	3
Pre-requisite / Co-requisite/ Exclusion	Pre-requisite: COMP2011 Data Structures or equivalent Co-requisite/Exclusion: Nil
Objectives	The objectives of this subject are to: <ol style="list-style-type: none">1. provide students with an understanding of various principles and paradigms in programming languages; and2. develop skills in describing, analysing, and using the features of programming languages.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: <u>Professional/academic knowledge and skills</u> <ol style="list-style-type: none">(a) understand the general language design principles;(b) understand the control flow and execution of a programming language;(c) be aware of different programming paradigms;(d) evaluate the design of a given programming language for the application at hand; <u>Attributes for all-roundedness</u> <ol style="list-style-type: none">(e) solve problems independently; and(f) think critically for a specific design and the rationale behind.

Subject Synopsis/ Indicative Syllabus	Topic	Duration of Lectures
	1. Programming language paradigms Overview of existing programming languages and programming paradigms; history of programming languages.	2
	2. Scoping and sub-routine Concept of blocks; environment; scope and visibility of variables; static and dynamic scoping; run-time stack; procedure call; parameter passing semantics; activation records and recursion.	4
	3. Concurrent & parallel languages Multithreading, message passing, languages: Scala, Go, Erlang, Clojure.	6
	4. Functional languages Mathematical functions, lists, function composition, languages (e.g., Lisp, ML, Haskell).	6
	5. Logic languages Predicate calculus, theorem proving, logic programming, languages (e.g., Prolog and its variants).	6
	6. Memory management in programming languages Pointers, dangling references, garbage collection, common practices for programming.	2
	Total	26
Teaching/Learning Methodology	<p>Lectures provide students the fundamental concepts of the topics, with corresponding illustrative examples.</p> <p>Tutorials and lab sessions enable students to experience with the features of programming languages of different paradigms.</p> <p>Assignments help students apply design and analysis techniques; whereas the project focuses on implementation skills.</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	f
	1. Assignments	55%	✓	✓			✓	
	2. Lab exercises		✓	✓	✓		✓	
	3. Project		✓	✓		✓		
	4. Mid-term / Tests		✓	✓			✓	✓
	5. Examination	45%	✓	✓			✓	✓
	Total	100%						
<p>Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:</p> <p>All five tasks are relevant to the assessment of programming language design principles and paradigms (for items a, b).</p> <p>Lab exercises are used to obtain hands-on experiences on programming languages of different paradigms (for item c); the project assesses the evaluation of a programming language for an application (for item d); the mid-term and examination are used to assess independent problem solving and critical thinking (for items e, f).</p>								
Student Study Effort Expected	Class contact:							
	▪ Lecture		26 Hrs.					
	▪ Tutorial/Lab		13 Hrs.					
	Other student study effort:							
	▪ Assignments		13 Hrs.					
	▪ Project		52 Hrs.					
	Total student study effort		104 Hrs.					
Reading List and References	<p>Textbooks:</p> <ol style="list-style-type: none"> Robert W. Sebesta, Concepts of Programming Languages, 12th Edition, Pearson, 2019. Allen B. Tucker and Robert E. Noonan, Programming Languages: Principles and Paradigms, Second Edition, McGraw-Hill, 2007. <p>Reference Books:</p> <ol style="list-style-type: none"> Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, Second Edition, Pearson/Addison-Wesley, 2007. Franklyn A. Turbak and Mark A. Sheldon, Design Concepts in Programming Languages, MIT Press, 2008. 							