

Subject Description Form

Subject Code	AMA2222
Subject Title	Principles of Programming
Credit Value	3
Level	2
Pre-requisite	Nil
Exclusion	Principles of Programming (AMA2222A)
Objectives	<ul style="list-style-type: none"> • To provide students with knowledge on the fundamental elements in computer programming. • To introduce advanced computer programming techniques necessary for developing more sophisticated computer application programs. • To enhance digital literacy and promote social responsibility and academic integrity related to the use of digital technologies.
Intended Learning Outcomes	<p>Upon completion of the subject, students will be able to:</p> <p><u>Professional/academic knowledge and skills</u></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;</p> <p>c) develop a computer program in a stepwise manner;</p> <p>d) use various software tools for developing computer programs;</p> <p>e) possess the ability to learn advanced programming techniques independently;</p> <p>f) possess the ability to learn other high level programming languages independently;</p> <p>g) <u>use generative AI to assist programming:</u></p> <p><u>Attributes for all-roundedness</u></p> <p>h) solve problems using systematic approaches;</p> <p>i) identify and develop problem solutions in a logical manner;</p> <p>j) learn independently any new technology;</p> <p>k) apply information technology with an entrepreneurial perspective;</p> <p>l) understand the social responsibility and academic integrity related to the use of digital technologies.</p>

Subject Synopsis/ Indicative Syllabus	<p>1. Fundamentals of Computing Basic concepts of computers and computing, compilation and interpretation, elementary programming constructs.</p> <p>2. Flow controls Basic flow control: selection, repetition and functions.</p> <p>3. Data Collections Structures, lists, sets and strings.</p> <p>4. Program Design Problem solving, problem correctness, algorithm design, testing and debugging.</p> <p>5. Introduction to Programming Paradigms and Applications Object-based Programming, Functional Programming, Logical Programming.</p> <p>6. AI and programming Application of generative AI in programming, social responsibility, academic integrity</p>																																		
Teaching/Learning Methodology	<p>This subject emphasizes both the conceptual elements in computer programming and practical experiences. The lectures will be taught in a workshop mode with hands-on exercises reinforcing taught concepts. Students are required to attend the laboratory sessions, which allows them to consolidate their concepts learnt in the lectures. Other practical work helps to reinforce the programming skills learned for applications.</p>																																		
Assessment Methods in Alignment with Intended Learning Outcomes	<table border="1" data-bbox="461 1256 1431 1671"> <thead> <tr> <th rowspan="2">Specific assessment methods/tasks</th> <th rowspan="2">% weighting</th> <th colspan="4">Intended subject learning outcomes to be assessed (Please tick as appropriate)</th> </tr> <tr> <th>(a) - (f)</th> <th>(g)</th> <th>(h), (i)</th> <th>(j) - (l)</th> </tr> </thead> <tbody> <tr> <td>1. Assignments, quizzes, lab projects</td> <td>36%</td> <td>✓</td> <td>✓</td> <td>✓</td> <td>✓</td> </tr> <tr> <td>2. Tests</td> <td>24%</td> <td>✓</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>3. Examination</td> <td>40%</td> <td>✓</td> <td></td> <td>✓</td> <td>✓</td> </tr> <tr> <td>Total</td> <td>100 %</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes: The assignments and lab projects require students to solve problems by writing a programing. This can assess technological proficiency and practical techniques in programming. To enhance digital literacy, students are allowed to use AI in some of the tasks. However, the quizzes, tests and the final exam will be given in written form where students are not allowed to use AI. Their understanding in the concepts, problem solving skills, computational thinking skills will be assessed.</p>	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)				(a) - (f)	(g)	(h), (i)	(j) - (l)	1. Assignments, quizzes, lab projects	36%	✓	✓	✓	✓	2. Tests	24%	✓		✓		3. Examination	40%	✓		✓	✓	Total	100 %				
Specific assessment methods/tasks	% weighting			Intended subject learning outcomes to be assessed (Please tick as appropriate)																															
		(a) - (f)	(g)	(h), (i)	(j) - (l)																														
1. Assignments, quizzes, lab projects	36%	✓	✓	✓	✓																														
2. Tests	24%	✓		✓																															
3. Examination	40%	✓		✓	✓																														
Total	100 %																																		

Student Study Effort Required	Class contact:	
	▪ Lecture	26 Hrs.
	▪ Lab	13 Hrs.
	Other student study effort:	
	▪ Assignments, quizzes, projects, exams	81 Hrs.
	Total student study effort	120 Hrs.
Reading List and References	<p>(1) John Zelle, Python Programming: An introduction to Computer Science, 3rd Edition, Franklin, Beedle & Associates, 2017</p> <p>(2) C. Thomas Wu, An Introduction to Object-Oriented Programming with Java, McGraw-Hill, 5th Edition Update, 2010.</p> <p>(3) Deitel & Deitel, Java: How to Program, Prentice-Hall, 11h Edition, 2017.</p> <p>(4) Deitel & Deitel, C++: How to Program, Prentice-Hall, 9th Edition, 2014.</p> <p>(5) Y. Daniel Liang, Introduction to Programming with C++, Pearson, 2014.</p> <p>(6) Tom Taulli, AI-Assisted Programming, O'Reilly Media, 2024</p>	