Subject Description Form

Subject Code	COMP1002
Subject Title	Computational Thinking and Problem Solving
Credit Value	4
Level	1
Pre-requisite / Co-requisite / Exclusion	Nil
Objectives	The objective of this subject is to: • equip students with no prior experience on computer programming with fundamental computational and skills. In particular, the students will learn how to abstract and solve problems, and how to implement them in a high-level programming language.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: *Professional/academic knowledge and skills:* (a) understand the basic concepts of computational thinking, including sequential logic, abstractions, conceptualization and problem-solving; (b) model real-life problems as computational problems; (c) develop computer solutions to problems of low-to-moderate complexity and implement them using a high-level programming language, e.g. Python; and (d) acquire the basic programming skills to implement solutions using suitable data types and constructs.
Subject Synopsis/ Indicative Syllabus	**Topic** **1. Introduction to Computational Thinking** Formulating problems for computers to solve them; logically organizing and analyzing data; representing data through abstractions; automating solutions through algorithmic thinking; implementing efficient solutions; generalizing the problem-solving process. **2. Problem Solving through a High-level Programming Language** Computing with numbers and strings; lists and files; functions; decision structures; loop structures and Booleans; sets and dictionaries. **3. Program Design** Problem analysis and design; function abstraction and modularization; bottom up and top down approaches

4. Application of Computational Techniques

Applications in different domains, for example, financial data computing, puzzle solving, development of games, web development, and scientific computation.

Teaching/ Learning Methodology

The 39-hour lecture will cover the main concepts and ideas in solving problems with computers and illustrate them using many examples. The students will also be given time to practice those concepts and ideas right away. The laboratory will be used to mainly cover program design.

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	
Continuous Assessment (such as assignments, quizzes and mini-projects)	**55%**	✓	✓	✓	✓	
Examination	**45%**	✓	✓	✓	✓	
Total	100%					

Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:

Assignments and quizzes are designed to help achieve learning outcome (a) and (d), whereas the two mini-projects are designed for achieving (b) and (c).

The examination will cover both (a), (c) and (d).

Student Study Effort Expected

Class contact:	
▪ Lecture/Lab/Tutorials	52 Hrs.
Other student study effort:	
▪ Assignments, Tests, Examination	53 Hrs.
▪ Self-Study	35 Hrs.
Total student study effort	140 Hrs.

Reading List and References

Reference Books:

1. Kowalski, Robert, *Computational Logic and Human Thinking: How to be Artificially Intelligent*, 1st Edition, Cambridge University Press, 2011.

2. Dromey, R. G., *How to Solve It by Computer. Prentice-Hall International*, Englewood Cliffs, NJ, USA, 1982. (There is a free copy online.)

	3. Zelle, John, *Python Programming: An Introduction to Computer Science* 3rd Edition. Franklin, Beedle & Associates Inc., 2017. 4. Downey, Allen B., *Think Python: How to Think Like a Computer Scientist*, Green Tea Press, 2015. 5. Punch, William F. and Enbody, Richard, *The Practice of Computing Using Python*, 3rd Edition, Addison Wesley, 2017. 6. Gries, Paul, Campbell, Jennifer and Montojo, Jason, *Practical Programming: An Introduction to Computer Science Using Python 3.6. Pragmatic Bookshelf*, 3rd Edition, 2017.