Subject Description Form

Subject Code	COMP1012
Subject Title	Programming Fundamentals and Applications
Credit Value	3
Level	1
Pre-requisite/ Co-requisite/ Exclusion	Exclusion: COMP1011/ENG2002
Objectives	The objectives of this subject are to: 1. provide students with knowledge on the fundamental elements in computer programming; and 2. introduce students to the application of computer programming in solving practical problems in different application domains.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: *Professional/academic knowledge and skills* (a) understand the programming elements for solving computing-related problems; (b) possess the ability to design and develop computer programs for solving problems in different application domains; (c) possess the ability to learn other high-level programming languages independently; *Professional/academic knowledge and skills* (d) develop skills in general problem solving; (e) identify and develop problem solutions in a logical manner; and (f) solve problems in groups and develop group work.
Subject Synopsis/ Indicative Syllabus	**Topic** 1. **Fundamentals of computing** Basic concepts of computers and computing, elementary programming constructs, elementary data types. 2. **I/O and flow control** Sending output to screen, getting input from keyboard, basic flow control: selection, repetition and function. 3. **Data collection** Sequences, lists, tuples, sets, strings and dictionaries. 4. **File operation** Creating and opening files, reading from file, writing to file.

<table>
<tr><td></td><td>5. Program design
Modular program design using functions, testing and debugging.
6. Applications
Sorting and searching: programming vs built-in Python functions, elementary data manipulation, NumPy arrays and matrices, problems in different application domains.
7. Other programming languages
Elementary data manipulation in R, interfacing to Python.</td></tr>
<tr><td>Teaching/Learning Methodology</td><td>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 assessment helps to reinforce the programming concepts and skills learned in developing applications. Individual assignments provide additional practices to programming. Project(s) allow students to work in group to solve more practical problems. Quizzes mandate students to recap their knowledge and skill sets acquired through other assessment forms.

Final examination provides a summative assessment of overall student performance in applying programming skills in solving problems in various applications.</td></tr>
<tr><td>Assessment Methods in Alignment with Intended Learning Outcomes</td><td>
<table>
<tr><td rowspan="2">Specific assessment methods/tasks</td><td rowspan="2">% weighting</td><td colspan="6">Intended subject learning outcomes to be assessed</td></tr>
<tr><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td></tr>
<tr><td>Continuous Assessment</td><td>65%</td><td colspan="6"></td></tr>
<tr><td>Assignments</td><td></td><td>√</td><td>√</td><td></td><td></td><td>√</td><td>√</td></tr>
<tr><td>Quizzes</td><td></td><td>√</td><td>√</td><td></td><td></td><td>√</td><td></td></tr>
<tr><td>Project(s)</td><td></td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>Final Examination</td><td>35%</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td></td></tr>
<tr><td>Total</td><td>100%</td><td colspan="6"></td></tr>
</table>

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

The continuous assessment and the final examination are designed to assess the specified learning outcomes. The formats may include written questions, programming exercises, projects and quizzes.</td></tr>
</table>

Student Study Effort Expected	Class contact:	
	▪ Lecture	39 Hrs.
	▪ Lab	13 Hrs.
	Other student study effort:	
	▪ Assignments, Quizzes, Projects, Exam, Self-study	68 Hrs.
	Total student study effort	120 Hrs.
Reading List and References	1. David J. Pine. *Introduction to Python for Science and Engineering*, CRC Press, 2019. 2. Claus Führer, Jan Erik Solem and Olivier Verdier. *Computing with Python: An Introduction to Python for Science and Engineering*. Pearson, 2014. 3. William F. Punch and Richard Enbody. *The Practice of Computing Using Python*. 3rd Edition, Addison Wesley, 2017. 4. Jaynal Abedin and Kishor Kumar Das. *Data Manipulation with R*, 2nd Edition, Packt Publishing, 2015. 5. J.D. Long and Paul Teetor. *R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics*. 2nd Edition, O'Reilly, 2019.	