Subject Description Form

Subject Code	COMP2022
Subject Title	Programming for FinTech Applications
Credit Value	3
Level	2
Pre-requisite / Co-requisite / Exclusion	Pre-requisite: Any programming related subject, e.g., COMP1001 or relevant IT background
Objectives	The objectives of this subject are to: • introduce to students about common programming techniques; • equip students with skills to quickly develop small programs to process and analyze data; and • allow students to practice with data processing and analysis for FinTech applications.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: *Professional/academic knowledge and skills* (a) understand the concept behind programming, independent of the programming language; (b) be able to develop small programs to process and analyze data; and (c) be able to apply programming skills to address more realistic problems arising from FinTech applications. *Attributes for all-roundedness* (d) work in a team to solve problems in FinTech to a larger scale. **Alignment of Programme Outcomes:** Programme Outcome 1: This subject contributes to having students practice their writing skills with report writing for the project. Programme Outcome 4: This subject contributes to developing student critical thinking through lab exercises, quizzes and the project. Programme Outcome 5: This subject contributes to enhancing the technical knowledge needed by students to solve problems through lab exercises, quizzes and the project. Programme Outcome 7: This subject contributes to team work for students to work in a team for the project. Programme Outcome 8: This subject contributes to the understanding of FinTech and to

	develop solutions through lab exercises and the project.
Subject Synopsis/ Indicative Syllabus	**Topic**
	1. Problem and Application Formulation Problem analysis; solution exploration; use of tools (e.g. Excel, Matlab) for fast solutions; handling large data sets; time series; common algorithms (e.g., concepts of sequence matching, alignment, similarity).
	2. Programming in Excel Use of formula and graph; table lookup; computing statistics; regression; programming macro with Visual Basic.
	3. Programming in Matlab Data representation in Matlab; vectors, matrices and vectorization; visualizing data; data manipulation via scripts and functions; file I/O; use of Matlab libraries.
	4. Programming in R R as a statistical computing language; vectors and matrices; functions; data import and export; built-in R packages; performing data analysis; interfacing to Python, C/C++ and Java.
	5. Case Studies and FinTech Application Development Concept of rapid programming; realizing common algorithms; modeling and Monte Carlo simulation; solving practical problems with multiple tools (e.g., exporting R results in .csv format for graphing); reacting to ad hoc changes in query and data.
Teaching/ Learning Methodology	A mixture of lecture-like class activity and project activity. Class activities include lectures, tutorials and laboratories.

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
1. Lab Exercises		✓	✓	✓	
2. Project		✓		✓	✓
3. Quizzes		✓	✓	✓	
4. Examination	45%	✓	✓	✓	
Total	100 %				

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

	Continuous assessments consist of lab exercises, a project and quizzes, which are designed to facilitate students to achieve the intended learning outcomes. Lab exercises are designed to encourage students to acquire good understanding of the relevant knowledge and to apply it to solve programming problems (i.e., enrich their hands-on programming experience). The project is designed to enhance students' ability to a deeper understanding of a problem of a larger-scope and solving it systematically. Quizzes are given to ensure the students' understanding of the concepts and capability of programming skills.
	Examination will provide a summative evaluation of the overall ability and understanding of the students in applying programming in FinTech.

Class contact:	
▪ Class activities (lecture, tutorial, laboratory, etc.)	39 Hrs.
Other student study effort:	
▪ Assignments, Quizzes, Projects, Examination	66 Hrs.
Total student study effort	105 Hrs.

Reading List and References	**Reference Books:**
	1. Joel Grus, *Data Science from Scratch*, O'Reilly Media, 2015.
	2. Lillian Pierson, *Data Science for Dummies*, John Wiley and Sons, 2015.
	3. Julitta Korol, *Microsoft Excel Programming Pocket Primer*, Mercury Learning and Information, 2015.
	4. Amos Gilat, *MATLAB: An Introduction with Applications*, 5th Edition, John Wiley & Sons, 2015.
	5. Stephen J. Chapman, *Essentials of MATLAB Programming*, Cengage Learning, 2009.
	6. Jaynal Abedin and Kishor Kumar Das, *Data Manipulation with R*, 2nd Edition, Packt Publishing, 2015.
	7. Paul Teetor, *The R Cookbook*, O'Reilly, 2011.
	8. Roger D. Peng, *Exploratory Data Analysis with R*, LeanPub, 2015.