Subject Description Form

Subject Code	COMP2021
Subject Title	Object-oriented Programming
Credit Value	3
Level	2
Pre-requisite / Co-requisite / Exclusion	**Pre-requisite**: COMP1011/COMP1012/ENG2002
Objectives	The objectives of this subject are to: 1. introduce students the basic elements of object-oriented programming; 2. teach students how to program computer systems using an object-oriented programming language; 3. familiarise students the tools that streamline object-oriented development; and 4. introduce lifelong learning to students
Intended Learning Outcomes	Upon completion of the subject, students will be able to: *Professional/academic knowledge and skills* (a) use an object-oriented programming language to solve computer problems; (b) use an object-oriented programming language to build computer systems; *Attributes for all-roundedness* (c) build computer systems in groups and develop group work; and (d) cooperate with team members in problem-solving. *Learning to learn* (e) recognize the need for lifelong learning; (f) plan, conduct, evaluate, and adjust their self-learning activities in problem-solving and software development.

Subject Synopsis/ Indicative Syllabus

Topic
1. Object-based programming. Concept of objects and classes. Correspondence between software objects and real-world objects. Object life cycle.
2. “Has-a” relationships and encapsulation. Data hiding and protection.
3. Object-oriented programming. Concept of class hierarchies. “Is-a” relationships and inheritance. Overriding of methods. Polymorphism. Run-time binding. Abstract classes and methods.
4. Multiple inheritance/Interfaces
5. Exception handling.
6. Generic programming.
7. Concurrency.
8. Use of UML to model OO projects.

Teaching/ Learning Methodology

This subject emphasizes both the conceptual elements of computer programming and practical experiences. A high-level, object-oriented programming language, such as C++ or Java, will be used for illustration.

The lectures will be used to deliver course materials, and the knowledge learned will be practiced/reinforced during the tutorials/labs. Individual/Group projects will be given to help students obtain hands-on development experience.

Certain course project requirements concern aspects of object-oriented programming that are not fully covered in lectures. Students need to plan, conduct, evaluate, and adjust their self-learning activities to master the related knowledge to accomplish the corresponding tasks.

Peer review of the project design and implementation will be organized to highlight the need for lifelong learning and to inspire perfectionism in students.

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
Continuous Assessment	**60%**						
1. Assignments, Quizzes & Projects		✓	✓	✓	✓	✓	✓
Final Examination	**40%**	✓	✓				
Total	100%						

Notes:

Project software artifacts submitted at the end of the subject will be assessed with respect to the project requirements, and the peer review reports need to contain

	students' reflections on the processes and results of their self-learning activities as well as the identified paths to improve their self-learning approaches. If a student fails either the continuous assessment component or the final exam component, his/her overall grade shall not exceed C-.	
Student Study Effort Expected	Class contact:	
	▪ Lecture	39 Hrs.
	▪ Tutorial/Lab	13 Hrs.
	Other student study effort:	
	▪ Assignments, Quizzes, Projects, Exam	68 Hrs.
	Total student study effort	120 Hrs.
Reading List and References	**Reference Books:** 1. Horstmann, Cay S., *Core Java Volume I – Fundamentals*, 10th Edition, Prentice Hall, 2016. 2. Bates, Bert and Sierra, Kathy, *Head First Java*, 2nd Edition, O'Reilly Media, 2005. 3. Bloch, Joshua, *Effective Java*, 2nd Edition, Addison-Wesley, 2008. 4. Larman, Craig, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd Edition, Prentice Hall, 2004.	