

## **Subject Description Form**

<b>Subject Code</b>	COMP2021									
<b>Subject Title</b>	Object-oriented Programming									
<b>Credit Value</b>	3									
<b>Level</b>	2									
<b>Pre-requisite / Co-requisite / Exclusion</b>	<b>Pre-requisite:</b> COMP1011									
<b>Objectives</b>	<p>The objectives of this subject are to:</p> <ol style="list-style-type: none"> <li>1. introduce students the basic elements of object-oriented programming;</li> <li>2. teach students how to program computer systems using an object-oriented programming language; and</li> <li>3. familiarise students the tools that streamline object-oriented development.</li> </ol>									
<b>Intended Learning Outcomes</b>	<p>Upon completion of the subject, students will be able to:</p> <p><u>Professional/academic knowledge and skills</u></p> <ol style="list-style-type: none"> <li>(a) use an object-oriented programming language to solve computer problems;</li> <li>(b) use an object-oriented programming language to build computer systems;</li> </ol> <p><u>Attributes for all-roundedness</u></p> <ol style="list-style-type: none"> <li>(c) build computer systems in groups and develop group work; and</li> <li>(d) cooperate with team members in problem-solving.</li> </ol>									
<b>Subject Synopsis/ Indicative Syllabus</b>	<table border="1"> <thead> <tr> <th>Topic</th></tr> </thead> <tbody> <tr> <td>1. Object-based programming. Concept of objects and classes. Correspondence between software objects and real-world objects. Object life cycle.</td></tr> <tr> <td>2. “Has-a” relationships and encapsulation. Data hiding and protection.</td></tr> <tr> <td>3. Object-oriented programming. Concept of class hierarchies. “Is-a” relationships and inheritance. Overriding of methods. Polymorphism. Run-time binding. Abstract classes and methods.</td></tr> <tr> <td>4. Multiple inheritance/Interfaces</td></tr> <tr> <td>5. Exception handling.</td></tr> <tr> <td>6. Generic programming.</td></tr> <tr> <td>7. Concurrency.</td></tr> <tr> <td>8. Use of UML to model OO projects.</td></tr> </tbody> </table>	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.
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.										

<b>Teaching/ Learning Methodology</b>	This subject emphasises both the conceptual elements in computer programming and practical experiences. A high- level, object-oriented programming language, such as C++ or Java, will be used for illustration purposes.					
	The lectures will be used to deliver course material that will be practiced/reinforced during the tutorials/labs. Individual/Group projects will be given to give students hand-on development experience.					
<b>Assessment Methods in Alignment with Intended Learning Outcomes</b>	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed			
			a	b	c	d
	<b>Continuous Assessment</b>	<b>60%</b>				
	1. Assignments, Quizzes & Projects		✓	✓	✓	✓
	<b>Final Examination</b>	<b>40%</b>	✓	✓		
	Total	100%				
Notes:						
If a student fails either the continuous assessment component or fails the final exam component, then his/her overall grade shall not exceed C-.						
<b>Student Study Effort Expected</b>	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.	
<b>Reading List and References</b>	<b>Reference Books:</b>					
	1. Horstmann, Cay S., <i>Core Java Volume I – Fundamentals</i> , 10 <sup>th</sup> Edition, Prentice Hall, 2016.					
	2. Bates, Bert and Sierra, Kathy, <i>Head First Java</i> , 2 <sup>nd</sup> Edition, O'Reilly Media, 2005.					
	3. Bloch, Joshua, <i>Effective Java</i> , 2 <sup>nd</sup> Edition, Addison-Wesley, 2008.					
	4. Larman, Craig, <i>Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development</i> , 3 <sup>rd</sup> Edition, Prentice Hall, 2004.					