Subject Description Form

Subject Code	EIE3320
Subject Title	Object-Oriented Design and Programming
Credit Value	3
Level	3
Pre-requisite	For 42470 and 42477: ENG2002 Computer Programming For 42375: EIE2264 Computer Programming
Co-requisite/ Exclusion	Nil
Objectives	This subject will provide students with the principles of object-oriented software design and programming from the perspective of Java implementation and UML. Students are expected to learn the concepts of and practical approaches to object-oriented analysis, design and programming using UML and Java.
Intended Subject Learning Outcomes	**Upon completion of the subject, students will be able to:** Category A: Professional/academic knowledge and skills 1. Understand the principles of object oriented design. 2. Apply Java in object oriented software development. 3. Apply UML in object oriented software modeling. 4. Apply object oriented approach to developing computer software. Category B: Attributes for all-roundedness 5. Learn independently and be able to search for the information required in solving problems. 6. Present ideas and findings effectively. 7. Think critically. 8. Work in a team and collaborate effectively with others.
Subject Synopsis/ Indicative Syllabus	**Syllabus:** 1. Introduction to Software Engineering Software products; software processes; software process models; 2. Java Programming Basic Java technologies; Java platform; Java language basic: variables, operators, expressions, statements, blocks, control flow, methods, arrays. 3. Object-Oriented Programming with Java Objects and classes; class definition; fields, constructors and methods; object interaction; grouping objects; array and collections; designing classes; inheritance and polymorphism; managing inheritance: creating subclasses and super-classes, hiding member variables, overriding methods. Interfaces and packages. 4. Web Programming with Java *JavaScript:* Client-side Web programming; JavaScript and HTML; Object, events, and event handlers in JavaScript. *Java Servlets*: architecture of servlets, client interaction, life cycle of servlets, saving client states; servlet communications, session tracking, and using server resources. 5. Unified Modelling Language (UML) Purposes of modelling. Structural Modelling: classes, relationships, class Diagrams, interfaces, packages, and object diagrams. Behavioural

modelling interactions, use cases, use case diagrams, interaction diagrams, activity diagrams. Architectural modelling: components, deployment, and collaborations. Mapping UML diagrams to Java Code.

Laboratory Experiment:

1. Laboratory Work
   Students will implement an on-line shopping system using Java Servlets and Tomcat Web server.

2. Practical Work
   Students will be requested to use integrated development environment (IDE) to write and debug Java programs during tutorial and lab sessions.

Teaching/ Learning Methodology

Teaching and Learning Method	**Intended Subject Learning Outcome**	**Remarks**
Lectures	1, 2, 3	fundamental principles and key concepts of the subject are delivered to students
MC Quizzes	1, 2, 3	students' knowledge on/understanding of certain topics can be easily estimated, and the corresponding teaching time will be adjusted accordingly
Tutorials	1,2,4,5,6	supplementary to lectures and are conducted with smaller class size; students will be able to clarify concepts through developing simple Java programs.
Laboratory sessions	4,5,7	Students will need to design, develop, test, and document Java programs.
Mini-project	3,4,5,7,8	Students in groups of 2-3 are required to build a 3-tier online shopping software. They will also need to use UML to document their software.

Assessment Methods in Alignment with Intended Subject Learning Outcomes

Specific Assessment Methods/ Task	% Weighting	Intended Subject Learning Outcomes to be Assessed (Please tick as appropriate)							
		1	2	3	4	5	6	7	8
1. Continuous Assessment (Total: 50%)									
• Assignments	8%		✓	✓	✓	✓		✓	
• Laboratory sessions	10%		✓		✓		✓	✓	✓
• Test	32%	✓							
2. Examination	50%	✓							
Total	100%								

The continuous assessment consists of a number of short quizzes, programming assignments, a mini-project, laboratory reports and a mid-term test.

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

Specific Assessment Methods/Tasks	Remark
Short quizzes	Multiple choices and true/false questions will be used to test and enhance students' understanding about the topics covered in lectures.
Assignments	Students will be asked to write Java programs and test the programs. They will also need to use UML diagram to illustrate the structure of their programs. Students will need to think critically and creatively in order to come up with a good solution for an existing problem.
Laboratory sessions and mini-project	Each group of students are required to produce a written report. Students will be accessed based on (1) the quality of their programs and (2) the clarity of their reports. Students will be asked to work as a team to develop a web application. Each of them will be responsible for part of the software.
Test and Exam	End-of-chapter problems will be used to evaluate students' ability in applying concepts and skills learnt in the classroom

Student Study Effort Expected	**Class contact (time-tabled):**	
	- Lecture	24 Hours
	- Tutorial/Laboratory/Practice Classes	15 hours
	Other student study effort:	
	- Lecture: preview/review of notes; homework/assignment; preparation for test/quizzes/examination	36 Hours
	- Tutorial/Laboratory/Practice Classes: preview of materials, revision and/or reports writing	30 Hours
	Total student study effort:	**105 Hours**
Reading List and References	**Textbooks:** 1. G. Booch, I. Jacobson and J. Rumbaugh, *The Unified Modeling Language User Guide*, 2nd ed., Addison-Wesley, 2005. 2. D.J. Barnes and M. Kolling, *Objects First with Java: A Practical Introduction using BlueJ*, 5th ed., Prentice-Hall, 2012. **Reference Books:** 1. H.M. Deitel and P.J. Deitel, *Java: How To Program (Early Objects)*, 10th ed., Prentice-Hall, 2014. 2. J. Lewis and W. Loftus, Java Software Solutions, 8th Edition, Pearson, 2015. 3. J. Rumbaugh, I. Jacobson and G. Booch, *The Unified Modeling Language Reference Manual*, 2nd ed., Addison-Wesley, 2004.	
Last Updated	February 2018	
Prepared by	Dr Lawrence Cheung	