Subject Description Form

Subject Code	EIE4108
Subject Title	Distributed Systems and Cloud Computing
Credit Value	3
Level	4
Pre-requisite	EIE3320 Object Oriented Design and Programming
Co-requisite/ Exclusion	Nil
Objectives	This subject will provide students with the principles of distributed systems and cloud computing. It enables students to master the development skills for providing and constructing distributed services on the Web and cloud. Through a series of lab exercises, students will be able to develop interoperable and distributed Web and cloud applications.
Intended Subject Learning Outcomes	**Upon completion of the subject, students will be able to:** Category A: Professional/academic knowledge and skills 1. Understand the concepts of distributed systems, cloud computing, and big data 2. Identify the key components in distributed systems, cloud services, and big data analytics 3. Build distributed systems. 4. Understand the advantages and limitations of different distributed system and cloud architectures. 5. Understand the enabling technologies for building distributed systems. 6. Understand the different components of distributed systems. 7. Set up and configure a distributed application. Category B: Attributes for all-roundedness 8. Think critically. 9. Learn independently. 10. Work in a team and collaborate effectively with others. 11. Present ideas and findings effectively.
Subject Synopsis/ Indicative Syllabus	**Syllabus:** 1. Introduction to Distributed Systems and Cloud Computing 1.1. Definition and examples of distributed systems; 1.2. Technologies for Network-Based Systems: multi-core and multi-threading; virtual machines; 1.3. Distributed and Cloud Computing Models: client-server; clusters; grids; peer-to-peer 2. Enabling Technologies for Building Distributed Systems 2.1. Socket Programming: datagram sockets; stream-mode sockets 2.2. Remote Method Invocation 2.3. Extensible Markup Language (XML): XML markup; XML namespaces; XML schema 3. Service-Oriented Architecture for Distributed Computing 3.1. Service and Service-Oriented Architectures 3.2. Web Services: simple object access protocol (SOAP); building web services with SOAP; web services description language (WSDL); role of WSDL in Web services; remote web-services invocation using WSDL; Web service implementation 3.3. RESTful Web Services: architectural principles of REST; REST vs. SOAP; AJAX; RESTful implementation; JAX-RS

4. Cloud Platform Architecture and Programming Environments
   4.1. Service Models: public clouds; private clouds; hybrid clouds
   4.2. Data Centres
   4.3. Virtualization: level of virtualization; hardware virtualization; server and storage consolidation;
   4.4. Layer and Types of Clouds: IaaS; PaaS; SaaS
   4.5. Cloud Programming Environments: Google App Engine;

5. Big Data Analytics
   5.1. Introduction to Big Data: big data use cases; source of big data
   5.2. Storing Big Data: unstructured databases; NoSQL; CAP theorem; key-value stores; document stores
   5.3. Distributed Computing with MapReduce: map and reduce tasks
   5.4. Hadoop: Hadoop clusters; Hadoop distributed file systems; implementation examples

Programming Exercises and Laboratory Experiments:

1. Multi-Threading
2. Socket Programming
3. Remote Method Invocation (RMI)
4. Web Services

Teaching/ Learning Methodology

Teaching and Learning Method	**Intended Subject Learning Outcome**	**Remarks**
Lectures	1,2,4,5,6	Fundamental principles and key concepts of the subject are delivered to students.
Tutorials	1,3,4,5,6,8,9	Supplementary to lectures and are conducted with smaller class size; Students will be able to clarify concepts and to have a deeper understanding of the lecture material; Programming exercises will be provided to strengthen students' hands-on experiences.
Laboratory sessions	3,6,7,8,10, 11	Students will go through the development process of various distributed systems and evaluate their performance.

Assessment Methods in Alignment with Intended Subject Learning Outcomes

Specific Assessment Methods/ Tasks	**% Weighting**	**Intended Subject Learning Outcomes to be Assessed (Please tick as appropriate)**										
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1. Continuous Assessment	40%											
• Short quizzes	3%	✓	✓		✓	✓	✓					
• Assignments	10%	✓	✓		✓	✓	✓		✓	✓		
• Tests	17%	✓	✓		✓	✓	✓		✓	✓		
• Laboratory sessions, mini-project	10%			✓			✓	✓	✓		✓	✓
2. Examination	60%	✓	✓		✓	✓	✓		✓	✓		
Total	100 %											

The continuous assessment consists of assignments, laboratory reports and tests.

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

Specific Assessment Methods/Tasks	Remark
Short quizzes	Short multiple choice quizzes are conducted to measure the students' understanding of the theories and concepts as well as their comprehension of subject materials.
Assignments, tests and examination	Assignments are of two types: (1) short essays on different types of distributed systems and (2) programming exercises demonstrating the operating principles of different distributed systems. The purposes are to strengthen students' understanding on the topics they learnt in classes. Students will be accessed based on their ability in applying concepts and skills learnt in the classroom. Students need to think critically and creatively in order to come with an alternate solution for an existing problem. Test and examinations are given to students to assess their competence level of knowledge and comprehension and their ability to apply knowledge and skills in new situations. The criteria (i.e. what to be demonstrated) and level (i.e. the extent) of achievement will be graded according to six levels: Excellent (A+ and A), Good (B+ and B), Satisfactory (C+ and C), Marginal (D) and Failure (F). These will be made known to the students before an assignment/homework is given. Feedback about their performance will be given promptly to students to help them improvement their learning.
Laboratory sessions and lab reports	Students are required to build two to three distributed systems and web services during the lab sessions. They are also required to write reports to explain the architecture and operating principle of their systems. Students will be accessed based on (1) their ability to apply knowledge that they learn in classes to build distributed systems and (2) their ability to write a clear report that explains the principle of operation and architecture of the systems that they have created.

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	**Reference Books:** 1. M.P. Papazoglou, *Web Services and SOA: Principles and Technology*, 2nd Edition, Prentice-Hall, 2013. 2. G. Coulouris, *Distributed Systems: Concepts and Design*, 5th ed., Addison-Wesley, 2011. 3. A.S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*, Prentice-Hall, 2007. 4. T. Erl, *Cloud Computing: Concepts, Technology and Architecture*, Prentice-Hall, 2013. 5. V. Mayer-Schönberger and K. Cukier, Big Data: A Revolution That Will Transform How We Live, Work, and Think, John Murray Pub., 2013. 6. T. White, "Hadoop: The Definitive Guide", O'Reilly, 3rd Ed. 2012
Last Updated	February 2018
Prepared by	Dr M.W. Mak