Subject Description Form

Subject Code	COMP408
Subject Title	Parallel and Distributed Computing
Credit Value	3
Level	4
Pre-requisite / Co-requisite/ Exclusion	Pre-requisite: COMP304, COMP312 (COMP307 for 61025) Co-requisite/Exclusion: Nil
Objectives	• To provide students with contemporary knowledge in parallel and distributed computing; • To equip students with skills to design and analyze parallel and distributed applications.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: *Professional/academic knowledge and skills* (a) understand the evolution of high performance computing (HPC) with respect to laws and the contemporary notion that involves mobility for data, hardware devices and software agents; (b) understand, appreciate and apply parallel and distributed algorithms in problem solving; (c) evaluate the impact of network topology on parallel/distributed algorithm formulations and traffic their performance; (d) gain hand-on experience with the agent-based and Internet-based parallel and distributed programming techniques; (e) master skills to measure the performance of parallel and distributed programs; (f) learn advanced techniques such as Internet caching and its application in practical systems; *Attributes for all-roundedness* (g) evaluate whether a parallel and distributed application is efficient or not by using the right tools, especially those time-critical ones; (h) apply the different techniques, including internet-based ones, efficaciously in e-business perspectives. **Alignment of Programme Outcomes:** Programme Outcome 1: It makes the student learn to present results, which are produced from the assignment project(s) that verify what they have learned in class.

	The quality of the report(s) measures how the student(s) has mastered what they learned. Programme Outcome 2: It helps student(s) grasp what factors would affect system correctness and stability. Programme Outcome 3: The team assignment helps students learn how to collaborate ethically. Programme Outcome 4: It helps students polish their critical thinking through the process of analyzing the project/programming results. Programme Outcome 5: The laboratory exercises and project assignments improve the students' problem solving skills. Programme Outcome 6: The subject matter points out where the edge of distributed and parallel computing is, and meanwhile helps students develop methods, by example, for lifelong learning. Programme Outcome 7: The group project inculcates team spirit.
Subject Synopsis/ Indicative Syllabus	**Topic** **1. Overview** High performance computing (HPC) paradigms evolution with respect to different laws and learning curves; importance of Moore's Law; supercomputing and the grid; network of workstations; applications of parallel and distributed computing. **2. Parallel computing** Different HPC system architectures and models: tightly coupled versus loosely coupled architectures, SIMD versus MIMD architectures; shared memory MIMD; message passing; problem decomposition and parallelization; synchronization techniques; parallel languages. **3. Distributed computing** Fundamental issues and problem types; naming facility; Lamport's logical clock; message passing primitives; remote procedure call; synchronization mechanisms; resource allocation; client-server computing; agents. **4. Selected topics** In-depth studies on EITHER parallel computing OR distributed computing. Parallel computing topics may include design of parallel algorithms, common parallel operators and reduction, one-to-all versus all-to-all operators, grid computing, performance monitoring. Distributed computing topics may include load balancing, distributed deadlock, fault-tolerance, dependability of distributed systems, use of caching to reduce response time, Internet-based distributed computing, Internet congestion control, Internet end-to-end performance measurement, Internet traffic pattern analysis. **Laboratory Experiment:** **Topic**

	1. Installing mobile agent or relevant platform. 2. Learning the programming language for the platform. 3. Application of the programming language to solve problems, e.g., Internet congestion control, distributed resource management, traffic analysis. **Total**
Teaching/Learning Methodology	The methodology consists of three main parts other that lectures: i) understand and rehearse – understanding is deepened through repeated class, tutorial and take-home exercise; basically the students are drilled in important topics by resolving them alone and then in open discussions. ii) associate – at this level effective learning is easily achieved by associating with hand-on experience; for this reason the theories are practiced in laboratory exercises and group projects in which students can discuss and learn from one another with a team spirit. iii) test and examine – this reinforces the rehearsal in the learning process so that short-term items can become long-term memory.
Assessment Methods in Alignment with Intended Learning Outcomes	See table below.

Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)							
		a	b	c	d	e	f	g	h
1. Assignments	10%	✓	✓	✓	✓	✓	✓	✓	✓
2. Lab exercises				✓	✓	✓	✓	✓	
3. Project	30%	✓	✓	✓	✓	✓	✓	✓	✓
4. Mid-term	15%	✓	✓			✓	✓		
5. Examination	45%	✓	✓	✓	✓	✓	✓	✓	✓
Total	100 %								

The assessment methods are appropriate to produce the expected outcome because they together represent an effective rehearsal process, in light of cognitive science, that transfers knowledge in the short-term memory into the long-term memory.

Student Study Effort Expected	Class contact:	
	▪ Lecture	39 Hrs.
	▪ Tutorial/Lab	0 Hrs.
	Other student study effort:	
	▪ Assigned reading	15 Hrs.
	▪ Take-home exercise	15 Hrs.

	Total student study effort	69 Hrs.
Reading List and References	1. Selected current articles from ACM and IEEE journals and conference proceedings 2. A.K.Y. Wong, T.S. Dillon and W.W.K Lin, Harnessing the Service Roundtrip Time Over the Internet to Support Time-Critical Applications – Concepts, Techniques and Cases, Nova Science Publishers, Inc. New York, 2008 3. G. Coulouris, J. Dollimore and T. Kindberg, Distributed Systems: Concepts and Design, 4th Edition, Addison Wesley, 2005.	