Subject Description Form

Subject Code	COMP 6702
Subject Title	Advanced Topics in Computation Theory
Credit Value	3
Level	6
Pre-requisite / Co-requisite/ Exclusion	Nil.
Objectives	• To provide students with in-depth knowledge on the key concepts in computation theory. • To introduce and practice advanced algorithmic techniques necessary for solving sophisticated computer science problems.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: (a) Understand the literature connected to computation theory; (b) Master important research methodology to tackle their research domain; (c) Present clearly their research works in written and verbal manner; (d) Conduct scholarly research in their problem domain independently.
Subject Synopsis/ Indicative Syllabus	**1. The computation model** Turing machine concepts; efficiency; the halting problem; undecidability. **2. An overview of complexity classes** **3. NP-completeness** the NP class; reductions; NP-completeness; other classes (coNP, EXP, NEXP). **4. Space complexity** the PSPACE and NL classes. **5. Randomized computation** probabilistic Turing machines; the classes RP, coRP, ZPP, BPP. **6. Complexity of counting** the class #P; #P completeness. **7. Interactive proof** interactive proof systems; the class IP; probabilistically checkable proof.
Teaching/Learning Methodology	Lectures provide students the main concepts of the topic, together with comprehensive examples for easy understanding. Tutorials offer an opportunity to students for practicing their analysis and problem solving skills. Written assignments will be utilized to help students develop analysis and

<table>
<tr><td></td><td>problem solving skills for computer science problems.</td></tr>
<tr><td>Assessment Methods in Alignment with Intended Learning Outcomes</td><td>
<table>
<tr><td rowspan="2">Specific assessment methods/tasks</td><td rowspan="2">% weighting</td><td colspan="6">Intended subject learning outcomes to be assessed (Please tick as appropriate)</td></tr>
<tr><td>a</td><td>b</td><td>c</td><td>d</td><td></td><td></td></tr>
<tr><td>1. Assignments</td><td>60%</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td></td><td></td></tr>
<tr><td>2. Examination</td><td>40%</td><td></td><td>✓</td><td>✓</td><td></td><td></td><td></td></tr>
<tr><td>Total</td><td>100%</td><td colspan="6"></td></tr>
</table>
Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:
Assignments help students understand computation theory concepts, apply them to solve problems, and conduct scholarly research in their problem domain (items a, b, d). In addition, assignments serve as practices for students to present their works clearly (item c).
The exam is used to assess the students in terms of problem solving skills and writing skills (items b, c).
</td></tr>
<tr><td>Student Study Effort Expected</td><td>
<table>
<tr><td>Class contact:</td><td></td></tr>
<tr><td>▪ Lecture/Tutorial</td><td>39 Hrs.</td></tr>
<tr><td>Other student study effort:</td><td></td></tr>
<tr><td>▪ Self-study</td><td>83 Hrs.</td></tr>
<tr><td>Total student study effort</td><td>122 Hrs.</td></tr>
</table>
</td></tr>
<tr><td>Reading List and References</td><td>
1. C.H. Papadimitriou, Computational Complexity, First Edition, Addison-Wesley, 1994.
2. O. Goldreich, Computational Complexity: A Conceptual Perspective, First Edition, Cambridge University Press, 2008.
3. S. Arora, B. Barak, Computational Complexity: A Modern Approach, First Edition, Cambridge University Press, 2009.
</td></tr>
</table>