Subject Description Form

<table>
<tr><td><b>Subject Code</b></td><td>COMP5211</td></tr>
<tr><td><b>Subject Title</b></td><td>Software Engineering Concepts</td></tr>
<tr><td><b>Credit Value</b></td><td>3</td></tr>
<tr><td><b>Level</b></td><td>5</td></tr>
<tr><td><b>Pre-requisite/ Exclusion</b></td><td>Nil</td></tr>
<tr><td><b>Objectives</b></td><td>The objectives of this subject are to:
1. provide a sufficient insight into the software development environment;
2. provide a detailed knowledge of the application of typical software engineering techniques;
3. provide appreciation of CASE tools.</td></tr>
<tr><td><b>Intended Learning Outcomes</b></td><td>Upon completion of the subject, students will be able to:
a) explain the software development life cycle;
b) know the basic techniques for requirement engineering, software design, and testing;
c) understand the CASE technology;
d) understand the maintenance issues and process; and
e) understand the need for coding standard, portability and reusability.</td></tr>
<tr><td><b>Subject Synopsis/ Indicative Syllabus</b></td><td>• <b>Programming myths and software crisis:</b> Principle of Software Engineering, The impact of CASE (Computer Aided Software Engineering) technology.
• <b>Software Requirement</b>: Software Requirement Engineering, Informal method and Formal method in software specification.
• <b>Software Design:</b> Software Structure, Software Development Tools: Notation & Techniques, Object-oriented Analysis and Design, Functional-oriented Analysis and Design, Critical comparison between OO Approach and Functional Approach.
• <b>Software Implementation</b>: Structured Coding Technique and Style Portability and Reusability.
• <b>Software Testing and Maintenance:</b> Software Quality Assurance, Software Verification and Validation Techniques, Re-engineering and reverse engineering concepts, Maintenance Issues.
• <b>Usability and usability engineering</b>: Software usability, Usability engineering.</td></tr>
<tr><td><b>Teaching/Learning Methodology</b></td><td>Class activities including - lecture, tutorial, lab, workshop seminar where applicable</td></tr>
<tr><td><b>Assessment Methods in Alignment with Intended Learning Outcomes</b></td><td>
<table>
<tr><td rowspan="2">Specific Assessment Methods/Tasks</td><td rowspan="2">% weighting</td><td colspan="5">Intended subject learning outcomes to be assessed</td></tr>
<tr><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td></tr>
<tr><td>Assignments, Tests & Projects</td><td>55</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td colspan="2"><table><tr><td>Final Examination</td><td>45</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr><tr><td>Total</td><td>100</td><td colspan="5"></td></tr></table></td></tr>
<tr><td rowspan="5"><b>Student study effort expected</b></td><td colspan="2"><b>Class Contact:</b></td></tr>
<tr><td>Class activities (lecture, tutorial, lab)</td><td>39 hours</td></tr>
<tr><td colspan="2"><b>Other student study effort:</b></td></tr>
<tr><td>Assignments, Quizzes, Projects, Exams</td><td>65 hours</td></tr>
<tr><td><b>Total student study effort</b></td><td><b>104 hours</b></td></tr>
<tr><td><b>Reading list and references</b></td><td colspan="2">(1) Sommerville, I., 2010, Software Engineering, 9th ed., Addison Wesley.
(2) Pressman, R., Maxim, B., 2014, Software Engineering, A Practitioner's Approach, 8th ed., McGraw Hill.
(3) Bennett, S., Ray, F., 2010, Object-Oriented Systems Analysis and Design Using UML, 4th Ed, McGraw-Hill.</td></tr>
</table>