# Subject Description Form

| | |
|---|---|
| **Subject Code** | COMP3021 |
| **Subject Title** | Programming Language Paradigms |
| **Credit Value** | 3 |
| **Pre-requisite / Co-requisite/ Exclusion** | Pre-requisite: COMP2011 Data Structures or equivalent<br>Co-requisite/Exclusion: Nil |
| **Objectives** | • To provide students with an understanding of various principles and paradigms in programming languages;<br>• To develop skills in describing, analyzing, and using the features of programming languages. |
| **Intended Learning Outcomes** | Upon completion of the subject, students will be able to:<br>*Professional/academic knowledge and skills*<br>(a) understand the general language design principles;<br>(b) understand the control flow and execution of a programming language;<br>(c) be aware of different programming paradigms;<br>(d) evaluate the design of a given programming language for the application at hand;<br>*Attributes for all-roundedness*<br>(e) solve problems independently;<br>(f) think critically for a specific design and the rationale behind. |

<table>
<tr><td><b>Subject Synopsis/ Indicative Syllabus</b></td><td>
<table>
<tr><th>Topic</th><th>Duration of Lectures</th></tr>
<tr><td><b>1. Programming language paradigms</b><br>Overview of existing programming languages and programming paradigms; history of programming languages.</td><td>2</td></tr>
<tr><td><b>2. Scoping and sub-routine</b><br>Concept of blocks; environment; scope and visibility of variables; static and dynamic scoping; run-time stack; procedure call; parameter passing semantics; activation records and recursion.</td><td>4</td></tr>
<tr><td><b>3. Concurrent &amp; parallel languages</b><br>Multithreading, message passing, languages: Scala, Go, Erlang, Clojure.</td><td>6</td></tr>
<tr><td><b>4. Functional languages</b><br>Mathematical functions, lists, function composition, languages (e.g., Lisp, ML, Haskell).</td><td>6</td></tr>
<tr><td><b>5. Logic languages</b><br>Predicate calculus, theorem proving, logic programming, languages (e.g., Prolog and its variants).</td><td>6</td></tr>
<tr><td><b>6. Memory management in programming languages</b><br>Pointers, dangling references, garbage collection, common practices for programming.</td><td>2</td></tr>
<tr><td><b>Total</b></td><td><b>26</b></td></tr>
</table>
</td></tr>
<tr><td><b>Teaching/Learning Methodology</b></td><td>Lectures provide students the fundamental concepts of the topics, with corresponding illustrative examples.<br>Tutorials and lab sessions enable students to experience with the features of programming languages of different paradigms.<br>Assignments help students apply design and analysis techniques; whereas the project focuses on implementation skills.</td></tr>
</table>

<table>
<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="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>e</td><td>f</td></tr>
<tr><td>1. Assignments</td><td rowspan="4">55%</td><td>✓</td><td>✓</td><td></td><td></td><td>✓</td><td></td></tr>
<tr><td>2. Lab exercises</td><td>✓</td><td>✓</td><td>✓</td><td></td><td>✓</td><td></td></tr>
<tr><td>3. Project</td><td>✓</td><td>✓</td><td></td><td>✓</td><td></td><td></td></tr>
<tr><td>4. Mid-term / Tests</td><td>✓</td><td>✓</td><td></td><td></td><td>✓</td><td>✓</td></tr>
<tr><td>5. Examination</td><td>45%</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>
<p>Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:</p>
<p>All five tasks are relevant to the assessment of programming language design principles and paradigms (for items a, b).</p>
<p>Lab exercises are used to obtain hands-on experiences on programming languages of different paradigms (for item c); the project assesses the evaluation of a programming language for an application (for item d); the mid-term and examination are used to assess independent problem solving and critical thinking (for items e, f).</p>
</td>
</tr>
<tr>
<td><b>Student Study Effort Expected</b></td>
<td>
<table>
<tr><td>Class contact:</td><td></td></tr>
<tr><td>▪ Lecture</td><td>26 Hrs.</td></tr>
<tr><td>▪ Tutorial/Lab</td><td>13 Hrs.</td></tr>
<tr><td>Other student study effort:</td><td></td></tr>
<tr><td>▪ Assignments</td><td>13 Hrs.</td></tr>
<tr><td>▪ Project</td><td>52 Hrs.</td></tr>
<tr><td>Total student study effort</td><td>104 Hrs.</td></tr>
</table>
</td>
</tr>
<tr>
<td><b>Reading List and References</b></td>
<td>
<p><b>Textbooks:</b></p>
<ol>
<li>Robert W. Sebesta, Concepts of Programming Languages, Ninth Edition, Addison Wesley, 2009.</li>
<li>Allen B. Tucker and Robert E. Noonan, Programming Languages: Principles and Paradigms, Second Edition, McGraw-Hill, 2007.</li>
</ol>
<p><b>Reference Books:</b></p>
<ol>
<li>Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, Second Edition, Pearson/Addison-Wesley, 2007.</li>
<li>Franklyn A. Turbak and Mark A. Sheldon, Design Concepts in</li>
</ol>
</td>
</tr>
</table>

| | |
|---|---|
| | Programming Languages, MIT Press, 2008. |