Aims

The aim of the module is to expose students to a comparative study of abstract concepts that are relevant to the design of programming languages, with particular focus on the constructs and implementation of various programming languages. The following concepts are introduced as examples: abstraction, syntax, semantics, binding times, data and sequence control, run-time resources, translators.

Intended Learning Outcomes

On completion of this module students should be able to:

Knowledge and understanding

1. Explaining different type of programming languages including their constructs and syntax. [1]
2. Specification of ‘better’ programme design and development using these programming languages; [2]
3. Analyse the strengths and weaknesses of various programming language paradigms; [3]

Subject-specific cognitive skills

4. Explain what a program (written in different programming language) does; [12]
5. Critically evaluate the capabilities and limitations of different programming language in order to inform selection of particular language for the implementation of given problems; [14]

Key/transferable skills

6. Plan, develop, evaluate and report on individual pieces of work. [21]

Content

- History of programming languages
- Language design issues
- Syntax and semantics of Programming Languages
- Data types
- Abstraction and encapsulation
- Sequence control
- Subprogram control
- Exceptions and exception handling
- Concurrent programming constructs
- Programming Languages Paradigms
- develop and debug small programmes in various programming languages

Methods of Learning, Teaching and Assessment

Total student effort for the module is 100 hours on average.

Learning and Teaching

<table>
<thead>
<tr>
<th>Type of session</th>
<th>ILOs Assessed</th>
<th>Student Effort</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Number in the Semester/s</td>
<td>Hours per week normally</td>
</tr>
<tr>
<td>Lecture</td>
<td>1-5</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>Tutorial</td>
<td>2-6</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>Laboratory</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Private study</td>
<td>1-5</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Assessment

<table>
<thead>
<tr>
<th>Assessment Focus</th>
<th>Weight %</th>
<th>Assessment Type</th>
<th>ILOs Assessed</th>
<th>Exam Semester</th>
<th>Exam/ Written Coursework Length (words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>individual</td>
<td>40%</td>
<td>Two In-Lab tests for two distinct programming languages</td>
<td>2-6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>One 120 minute unseen written exam.</td>
<td>1-5</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Feedback given to students in response to assessed work

Feedback will be provided for each assessed component in written form as appropriate. Any coursework will be returned to students with feedback on the accompanying coursework turn in sheet. If students require additional feedback, they are welcome to speak with the TA, and the module-leader.
Developmental feedback generated through teaching activities:

Dialogue between students and TAs and module-leader in the Labs and during office hours.

Reading List: