F28LL Programming Languages

### COURSE DETAILS
- **Course Code:** F28LL
- **Full Course Title:** Programming Languages
- **SCQF Level:** 8
- **SCAF Credits:** 15
- **Available as Elective:** No

### DELIVERY LEVEL
- **Undergraduate:** Yes
- **Postgraduate Taught:** No
- **Postgraduate Research:** No

### COURSE AIMS
- To gain understanding of different language paradigms
- To gain understanding of defining concepts of programming languages
- To develop skills in programming in languages from key paradigms

### LEARNING OUTCOMES – SUBJECT MASTERY
- Understanding of distinguishing characteristics of language paradigms
- Understanding of relationships between languages
- Understanding of generic language concepts
- Ability to program in languages from key paradigms
- Ability to use tool sets for these languages
- Ability to relate learned knowledge to work based computing projects

### LEARNING OUTCOMES – PERSONAL ABILITIES
- Understanding of how to choose an appropriate language for different problem domains
- Can relate and/or apply learned knowledge at work place
- Take significant responsibility for their work and for a range of resources
- To be aware of distinctive features of programming languages and of the impact of their choice on industrial projects
- Can communicate effectively with work colleagues on learned issues

### SYLLABUS
- Overviews of language history, definition (lexicon, syntax, semantics), implementation (compiler, interpreter, virtual machine)
- Overviews of language paradigms: e.g. imperative (high-level, system, low-level), declarative (functional, logic), concurrency/parallelism
F28LL Programming Languages

- Overviews of programming language concepts: variable, lvalue & rvalue, assignment (sharing/copying), data abstraction (sequential, structured, recursive, shared/distributed), type mechanisms (weak/strong, static/dynamic, ad-hoc/parametric polymorphism), declaration (scope, extent), control abstraction (sequence, choice, repetition, block, procedure, labels/jumps, exceptions, processes), expression abstraction (functions), parameter mechanisms (value, reference), evaluation mechanisms (strict/lazy, ordered/unordered, concurrent)
- An introduction to programming in languages from key paradigms e.g.
  - scripting: e.g. Python
  - declarative/functional: e.g. SML
  - declarative/logic: e.g. Prolog

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Level</th>
<th>Title</th>
<th>School</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>F27SB</td>
<td>7</td>
<td>Software Development 2</td>
<td>School of Math and Comp Sci.</td>
<td>Pre-Requisite</td>
</tr>
<tr>
<td>F27CX</td>
<td>7</td>
<td>Introduction to Computer Systems</td>
<td>School of Math and Comp Sci.</td>
<td>Pre-Requisite</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location and Assessment Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edi</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>Coursework</td>
</tr>
</tbody>
</table>