# Course descriptor F21AS

<table>
<thead>
<tr>
<th>Course code</th>
<th>F21AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course title</td>
<td>Advanced Software Engineering</td>
</tr>
<tr>
<td>Credits</td>
<td>15</td>
</tr>
<tr>
<td>School</td>
<td>Mathematics and Computer Sciences</td>
</tr>
<tr>
<td>SCQF Level</td>
<td>11</td>
</tr>
<tr>
<td>Semester</td>
<td>2</td>
</tr>
</tbody>
</table>

## Aims

This course aims to:
- To consolidate proficiency in imperative programming and software development
- To further develop object oriented programming and object oriented design methods
- To provide knowledge of simple data structures and algorithms
- To introduce concurrent programming techniques
- To instil understanding of the concepts and benefits of advanced software engineering methods
- To give further practical experience of the use of UML in software engineering
- To give practical experience of developing a substantial software engineering team project
- To enable the deployment of patterns in software engineering

## Syllabus

- Data structures: stacks, queues, lists, priority queues, binary trees
- Algorithms: searching (linear and binary) and sorting
- Advanced object oriented design techniques
- Thread based programming: thread creation and interaction, shared variables and synchronisation
- Methodologies in software engineering practice; Unified Modelling Language; design patterns
- Project planning and management in software engineering
- Comparison of agile and plan driven approaches

## Learning Outcomes

### Subject Mastery

- Skill in the use of UML notation and translation of UML designs to working programs
- Understanding of basic data structures and algorithms and ability to critically evaluate their appropriateness and limitations for a range of moderately complex problems.
- Demonstration of skill in design and implementation of practical GUI based and theaded applications
- To demonstrate a critical understanding of modern software engineering practice and be able to evaluate the strengths
and weaknesses of current software engineering methods and techniques

- To be able to choose appropriate metrics to measure software quality and quantity in a modern software engineering environment

- To be able to choose a suitable software development environment and development methodology for specific software development tasks and justify the choice

| Personal Abilities | appreciation of use of methodology to ground system analysis, design and development
|                    | understanding of different programming paradigms and their inter-relation
|                    | practice in working in a group, choosing a methodology, reaching a consensus, and working with others to a deadline
|                    | taking responsibility for own work, taking responsibility in the development of resources, critical reflection on development process and work undertaken by self.
|                    | effective appreciation of professional standards in modern software engineering practice.
|                    | showing initiative, creativity and team working skills in collaborative software development

| Assessment method  | 50% written examination, 50% coursework |