<table>
<thead>
<tr>
<th>Module Code</th>
<th>CS7CS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Name</td>
<td>Advanced Software Engineering</td>
</tr>
<tr>
<td>ECTS weighting</td>
<td>10</td>
</tr>
<tr>
<td>Term</td>
<td>HT</td>
</tr>
<tr>
<td>Contact Hours</td>
<td>4 hours per teaching week plus laboratory hours</td>
</tr>
<tr>
<td>Module Personnel</td>
<td>Professor Siobhán Clarke</td>
</tr>
</tbody>
</table>

### Learning Outcomes

Students who complete this module should be able to:

- CS3LO2 assess the theory of classic architecture principles and apply an appropriate architectural model in a team-based application under development;
- CS3LO3 evaluate the quality of object-oriented code against standard OO principles;
- CS3LO4 refactor object-oriented code to conform to standard OO principles;
- CS3LO5 apply test-driven design principles;
- CS3LO6 assess the theory of team-based software development, and of agile processes, and apply in a group project;
- CS3LO7 analyse, specify, design, write and test a large-scale, team-based application solving a complex problem;
- CS3LO8 select and apply appropriate structuring mechanisms, data placement strategies and code quality techniques for large-scale applications;
- CS3LO9 document and demonstrate a large-scale, team-based application to assessors;
- CS3LO10 evaluate the current state of the art in any software engineering topic.

### Module Learning Aims

The purpose of this course is to expose students to the advanced theory and practice of contemporary software engineering techniques and team-based software development; and to the current state of the art in software engineering research. A core aim of this course is to provide students with experience of developing a large-scale application.

Upon completion of the course students will have gained experience of analysing, specifying, designing and implementing a large-scale application, as part of a team. The projects will bestrand specific. In addition to addressing debugging and testing challenges, they will also have been exposed to the challenges posed by communicating with team members and other stakeholders (e.g., potential funders), project management, code integration and demonstrating such an application.

### Module Content

Specific topics addressed in this module include:

- Software development lifecycles;
- Domain analysis;
- Requirements and specification;
- Software architecture;
- Object-oriented design principles;
- Agile process, in particular eXtreme Programming (XP);
- Test-driven development;
- Refactoring.

### Module Pre-Requisite

None

### Assessment Details

Coursework: 100%

Coursework evaluation is based on the end-of-year project documentation, a demonstration to the course lecturer and, where relevant, other stakeholders, an oral examination within teams, and peer assessment. 50% of the marks are group-wide, and 50% is individual. Criteria for evaluation are:
1. Application of agile process to group project
2. Application of appropriate systems’ algorithms in group project;
3. Code quality within group project code-base;
4. Individualised assessment based on team-members’ peer reviews, combined with lecturer/TA observation throughout the semester.
5. Q&A on software engineering theory.