<table>
<thead>
<tr>
<th><strong>Module Code</strong></th>
<th>CS3D5A, CS3D5B.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module Name</strong></td>
<td>Software Design and Implementation.</td>
</tr>
<tr>
<td><strong>Module Short Title</strong></td>
<td>N/a.</td>
</tr>
</tbody>
</table>
| **ECTS weightings** | CS3D5A: 5 (BAI CD,)  
CS3D5B: 10 (BAI D.) |
| **Semester/term taught** | CS3D5A: Semester 1 (BAI CD,)  
CS3D5B: Semesters 1 and 2 (BAI D.) |
| **Contact Hours** | Lectures/week: 0.  
Lab/week: 3.  
Tutorial/week: 1.  
CS3D5A: 44 hours total (BAI CD,)  
CS3D5B: 88 hours total (BAI D.) |
| **Module Personnel** | Lecturing staff: Anton Gerderlan  
gerdela@scss.tcd.ie |
| **Learning Outcomes** | 0. Undertake software design and construction as members of teams of various sizes.  
1. Learn how to choose, learn, and use new languages, tools, and techniques.  
2. Gather requirements and develop a problem specification.  
4. Plan implementation of the program taking into account time and team management.  
5. Implement a program of reasonable complexity in the Java language.  
6. Document the project using standard techniques.  
7. Test the solution using standard techniques.  
8. Present their work to their peers and their clients. |
| **Module Learning Aims** | To develop practical skills in software engineering through small and large group programming projects. To learn how to work effectively with others throughout project lifecycles. |
| **Module Content** | Introductions to:  
- Android platform for mobile devices.  
- Java programming language.  
- Android Studio integrated development environment.  
- Git source code management system.  
- Testing techniques for quality assurance.  
- Agile development methodology.  
For the larger assignments, the applications are proposed, discussed, and agreed by the groups themselves. |
| **Recommended Reading List** | Relevant material on the world-wide web will be introduced. |
| **Module Pre Requisite** | None. |
### Module Co-Requisite

None.

### Assessment Details

Assessment of this module is entirely by coursework with significant weighting being given to peer-review.

Most marks to be awarded for correctness and completeness of implementation with respect to the problem specification, with smaller components for quality of technical documentation and employment of suitable high-level design.

A breakdown of the marking scheme for a typical project is as follows. Some projects could have specific challenges which would warrant different schemes.

**Correctness and completeness of implementation:**

1. Satisfying the project specification, ≤ 30%. This covers the basic ability of the programme to perform as required by the project specification and (where appropriate) to correctly process sample provided.
2. Satisfying good programming practice (selection of suitable algorithms and data structures, resource management,) ≤ 20%.

**Quality of solution design:**

1. Application of object-oriented programming concepts (selection of suitable classes, designing of appropriate interfaces between those classes, projected ability to re-use the software components where applicable), ≤ 15%.
2. Construction of each class (selection of suitable methods and design patterns for each class, suitable parameterisation of the methods, appropriate visibility for class members,) ≤ 15%.

**Documentation:**

1. Project-level description and background, ≤ 10%.
2. Maintenance-oriented commentary in program code, ≤ 10%.

A total mark of at least 40% should be achieved to pass. Late work will not be accepted.

### Module approval date

*September 2011.*

### Approved By

### Academic Start Year

### Academic Year of Data