## Module Descriptor 2017/18
School of Computer Science and Statistics.

<table>
<thead>
<tr>
<th>Module Code</th>
<th>CS7NS6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Name</td>
<td>Distributed Systems</td>
</tr>
<tr>
<td>Module Short Title</td>
<td>N/A</td>
</tr>
<tr>
<td>ECTS weighting</td>
<td>5</td>
</tr>
<tr>
<td>Semester/term taught</td>
<td>HT (2nd Semester)</td>
</tr>
<tr>
<td>Contact Hours</td>
<td>2 lecture hours and 1 tutorial hour per week</td>
</tr>
<tr>
<td>Module Personnel</td>
<td>Professor Vinny Cahill</td>
</tr>
</tbody>
</table>

### Learning Outcomes
When students have successfully completed this module they should be able to:
- NS7LO1 Describe the basic characteristics, structure and operation of a distributed system, and the issues that a distributed system poses for a system architect;
- NS7LO2 Identify and evaluate appropriate architectural models and paradigms for building large-scale distributed systems;
- NS7LO3 Design, construct, document and test distributed systems;
- NS7LO4 Reason about the trade-offs between scalability, performance, consistency, reliability and availability in distributed systems;
- NS7LO5 Make use of appropriate documentation and reference material.

### Module Learning Aims
Building distributed applications is still a difficult task due to the concurrency, communication latency, and possibility of partial failure that are inherent in computer networks. As in other areas of computer science, the trend in providing support for building distributed applications has been towards presenting application developers with high level abstractions on which to base the designs of their software.

This course tasks a critical look at some of the architectural issues involved in, and paradigms available for, the construction of large-scale distributed systems such as the infrastructures supporting Google’s search engine or Amazon’s online sales platform. In particular, the course considers how to develop systems that must make trade-offs between performance, consistency, reliability, and availability.

Students will be given opportunities to develop their problem solving, programming and communication skills by designing solutions to distributed systems programming problems as well as implementing such solutions as distributed systems.

### Module Content
Specific topics addressed in this module include:
- Distributed algorithm design
  - Mutual exclusion, leader election
- Characterisation of failure and failure models
- Atomic transactions
- Consistency models
  - Serializability
- Coordination, consensus, agreement
  - Distributed commit protocols
- Recovery
  - Logging, checkpointing
- Replication
  - Sequential consistency
- Process Groups
- Group communication
  - Reliable and ordered multicast

### Recommended Reading List
<table>
<thead>
<tr>
<th>Module Pre Requisite</th>
<th>Experience of building concurrent (multi-threaded) and networked applications.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Co Requisite</td>
<td>N/A</td>
</tr>
</tbody>
</table>
| Assessment Details   | Exam: 60%  
Coursework: 40%  
The 40% coursework mark is made up of continuous assessment (10%) and an individual project (30%).  
Student will work on the coursework outside of lecture and tutorial times. A mandatory development project of significant scope is undertaken over the semester. Continuous assessment is composed of in-class presentations on influential distributed systems. The final grade awarded will be a simple accumulation of grades achieved in each element. |
| Module approval date | 2017-2018                                                                       |
| Approved By          |                                                                                 |
| Academic Start Year  | 2017-2018                                                                       |
| Academic Year of Data| 2017-2018                                                                       |