<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>ECTS weighting</td>
<td>5</td>
</tr>
<tr>
<td>Term</td>
<td>HT (2nd Semester)</td>
</tr>
<tr>
<td>Contact Hours</td>
<td>2 lecture hours 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 which a distributed system poses to a systems architect;

NS7LO2 Identify and evaluate appropriate architectural models for distributed problem scenarios;

NS7LO3 Design, construct, document and test distributed system solutions to realistic real-world problems;

NS7LO4 Reason about the performance trade-offs of decentralised architectures;

NS7LO5 Make use of appropriate documentation and reference material.

**Module Learning Aims**

Building distributed applications is a difficult task due to the concurrency, communication latency, and possibility of partial failure that is inherent in distributed systems. As in other areas of computer science, the trend in providing support for building distributed applications has been towards presenting the application developer with ever higher levels of abstraction and, in the particular case of distributed programming, of location transparency. This course takes a critical look at some of the paradigms and architectural issues involved in distributed programming and their likely evolution.

Students will be given opportunities to develop their problem solving, programming and written communication skills by designing solutions to distributed system programming problems, and implementing those solutions as fully networked distributed systems.

**Module Content**

Specific topics addressed in this module include:

- Network Programming
- Concurrency
- Representational State Transfer
- Remote Procedure Call, Remote Object Technology and Middleware
- Synchronization and coordination
- Distributed transactions
- Authentication
- Replication
- Peer-to-peer and other large scale architectures
- Enterprise Infrastructure and Paradigms – Web services
- Advanced network application frameworks

**Assessment Details**

Exam: 70%
Coursework: 30%

The 30% coursework mark is made up of continuous assessment (10%) and an individual project (20%).

A mandatory individual project of significant scope is undertaken over the term which students must work on outside lecture and tutorial times. Continuous assessment is composed of a number of marked tutorial exercises and programming assignments. The final grade awarded will be a simple accumulation of grades achieved in each element.