**Course Title:** Distributed Systems

<table>
<thead>
<tr>
<th>Course Type:</th>
<th>Main Course</th>
<th>Prerequisites:</th>
<th>Advanced Computer Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Type:</td>
<td>Theoretical</td>
<td>Project:</td>
<td>None</td>
</tr>
<tr>
<td>Units:</td>
<td>3</td>
<td>Teaching Hours:</td>
<td>48</td>
</tr>
</tbody>
</table>

**Objectives:**
Distributed systems are common. Computer scientists and engineers need to understand what are the principles and paradigms underlying distributed systems software and be familiar with several real world examples. This course systematically examines the underlying principles and how they are applied to a wide variety of distributed systems in depth.

**Syllabus:**

1. **Introduction**: Definition, Goals, Hardware and Software Concepts, Client-Server Model
2. **Communication**: Layered Protocols, Remote Procedure Call, Remote Object Invocation, Message-Oriented Communication, Stream-Oriented Communication
3. **Processes**: Threads, Clients, Servers, Code Migration, Software Agents
4. **Naming**: Naming Entities, Locating Mobile Entities, Removing Unreferenced Entities
5. **Synchronization**: Clock Synchronization, Logical Clocks, Global State, Election Algorithms, Mutual Exclusion, Distributed Transactions
6. **Consistency and Replication**: Data- And Client-Centric Consistency Models, Distribution Protocols, Consistency Protocols
7. **Fault Tolerance**: Process Resilience, Reliable Client-Server and Group Communication, Distributed Commit, Recovery
8. **Security**: Secure Channels, Access Control, Security Management, Kerberos
9. **Exemplar Distributed Systems**: Object-Based, Document-Based, Coordination-Based

**References:**