

Subject Description Form

Subject Code	COMP436
Subject Title	Middleware and Distributed Objects
Credit Value	3
Level	4
Pre-requisite / Co-requisite/ Exclusion	Pre-requisite: COMP201 (Nil for 61025) Co-requisite: Nil Exclusion: COMP403
Objectives	<ul style="list-style-type: none"> • To present an integrated view of the basic building blocks of a distributed system and how middleware can help developers to more easily satisfy the requirements of building distributed systems. • To provide the foundation knowledge of middleware, particularly object-oriented middleware. • To provide training in using CORBA as middleware to build practical distributed systems.
Intended Learning Outcomes	<p>Upon completion of the subject, students will be able to:</p> <p><i>Professional/academic knowledge and skills</i></p> <p>(a) understand the basic structure of distributed systems;</p> <p>(b) understand the motivation of using middleware;</p> <p>(c) understand the basic theories underlying the design of middleware;</p> <p>(d) learn to make judgment in choosing a suitable middleware for application problems;</p> <p>(e) understand the basic concepts of CORBA;</p> <p>(f) develop distributed object-based systems using CORBA.</p> <p><i>Attributes for all-roundedness</i></p> <p>(g) apply the technical knowledge learned to solve real-life practical problems;</p> <p>(h) appreciate and evaluate existing and new technologies.</p> <p>Alignment of Programme Outcomes:</p> <p>Programme Outcome 5: This subject contributes to the programme outcome through the teaching of the related concepts of distributed systems and middleware, practising programming techniques useful for developing middleware-based systems with laboratory exercises and assignments, as well</p>

	<p>as assessing their knowledge of the development and application of middleware in distributed systems with programming assignments and examination.</p> <p>Programme Outcome 6: This subject contributes to the programme outcome through the teaching of the trend of development of various technologies related to middleware.</p>							
<p>Subject Synopsis/ Indicative Syllabus</p>	<table border="1" data-bbox="448 383 1406 1473"> <thead> <tr> <th data-bbox="448 383 1406 427" style="text-align: center;">Topic</th> </tr> </thead> <tbody> <tr> <td data-bbox="448 427 1406 607"> <p>1. Principles of object-oriented middleware Role of middleware in distributed systems; types of middleware; object-oriented middleware; local versus distributed objects; developing systems with object-oriented middleware.</p> </td> </tr> <tr> <td data-bbox="448 607 1406 763"> <p>2. Fundamentals of CORBA Architecture; Interface definition language (IDL); system development using CORBA.</p> </td> </tr> <tr> <td data-bbox="448 763 1406 987"> <p>3. Communication paradigms of CORBA Synchronous requests; oneway requests; deferred synchronous requests; asynchronous requests; dynamic invocation; CORBA event service; pros and cons of different communication paradigms of CORBA.</p> </td> </tr> <tr> <td data-bbox="448 987 1406 1133"> <p>4. Portable Object Adaptor (POA) Objects vs. servants; lifecycle of objects; request invocation via POA; servant activator and servant locator.</p> </td> </tr> <tr> <td data-bbox="448 1133 1406 1279"> <p>5. Case study 1: load balancing Using POA to implement various load balancing solutions for distributed systems.</p> </td> </tr> <tr> <td data-bbox="448 1279 1406 1473"> <p>6. Case study 2: resource management Using CORBA to implement facilities for resource management in distributed systems, e.g. resource lookup, resource acquisition; CORBA naming service.</p> </td> </tr> </tbody> </table> <p>Laboratory Experiment:</p> <p>In the laboratory session, students will learn how to develop distributed systems using an implementation of CORBA and using Java as the programming language.</p> <p>Case Study:</p> <p>Case studies on load balancing and resource management with CORBA.</p>	Topic	<p>1. Principles of object-oriented middleware Role of middleware in distributed systems; types of middleware; object-oriented middleware; local versus distributed objects; developing systems with object-oriented middleware.</p>	<p>2. Fundamentals of CORBA Architecture; Interface definition language (IDL); system development using CORBA.</p>	<p>3. Communication paradigms of CORBA Synchronous requests; oneway requests; deferred synchronous requests; asynchronous requests; dynamic invocation; CORBA event service; pros and cons of different communication paradigms of CORBA.</p>	<p>4. Portable Object Adaptor (POA) Objects vs. servants; lifecycle of objects; request invocation via POA; servant activator and servant locator.</p>	<p>5. Case study 1: load balancing Using POA to implement various load balancing solutions for distributed systems.</p>	<p>6. Case study 2: resource management Using CORBA to implement facilities for resource management in distributed systems, e.g. resource lookup, resource acquisition; CORBA naming service.</p>
Topic								
<p>1. Principles of object-oriented middleware Role of middleware in distributed systems; types of middleware; object-oriented middleware; local versus distributed objects; developing systems with object-oriented middleware.</p>								
<p>2. Fundamentals of CORBA Architecture; Interface definition language (IDL); system development using CORBA.</p>								
<p>3. Communication paradigms of CORBA Synchronous requests; oneway requests; deferred synchronous requests; asynchronous requests; dynamic invocation; CORBA event service; pros and cons of different communication paradigms of CORBA.</p>								
<p>4. Portable Object Adaptor (POA) Objects vs. servants; lifecycle of objects; request invocation via POA; servant activator and servant locator.</p>								
<p>5. Case study 1: load balancing Using POA to implement various load balancing solutions for distributed systems.</p>								
<p>6. Case study 2: resource management Using CORBA to implement facilities for resource management in distributed systems, e.g. resource lookup, resource acquisition; CORBA naming service.</p>								
<p>Teaching/Learning Methodology</p>	<ol style="list-style-type: none"> 1. Lecture: students learn the technologies and concepts related to middleware. 2. Laboratory session: students implement short programs (with guidance of the tutor) related to the lecture to gain experience in using the technologies and concepts learned. The programming tools used are CORBA and Java. 							

Assessment Methods in Alignment with Intended Learning Outcomes	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)							
			a	b	c	d	e	f	g	h
	1. Assignments	40%	✓		✓		✓	✓		
	2. Lab exercises									
	3. Project									
	4. Mid-term	15%	✓	✓	✓	✓	✓		✓	✓
	5. Examination	45%	✓	✓	✓	✓	✓		✓	✓
	Total	100 %								
<p>Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:</p> <p>For the assignments, the students have to design and implement middleware-based systems using CORBA and Java to solve common problems in distributed systems study (e.g., load balancing). This requires good understanding/application of distributed systems concepts and programming skills/techniques in using CORBA/Java to solve real problems.</p> <p>The mid-term test and the examination aim at assessing the students' understanding of the concepts related to the theory and practice of middleware and distributed systems.</p>										
Student Study Effort Expected	Class contact:									
	▪ Lecture								39 Hrs.	
	▪ Lab (may be arranged)								0 Hrs.	
	Other student study effort:									
	▪								Hrs.	
	▪								Hrs.	
	Total student study effort								At least 39 Hrs.	
Reading List and References	Textbooks:									
	1. W. Emmerich, Engineering Distributed Objects, Wiley, 2000.									
Reference Books:										
1. F. Bolton, Pure CORBA, Sams, 2002.										
2. R. Orfali, D. Harkey and J. Edwards, Client/Server Survival Guide, 3rd ed.,										

Wiley, 1999.

3. IEEE Distributed Systems Online,.
4. Articles from journals, magazines, and conference proceedings, including ACM TOCS, IEEE TPDS, IEEE TSE, IEEE TOC, CACM, IEEE Computer, ICDE, DOA.