



# Course Specification

## (Bachelor)

Course Title: **Distributed and Concurrent Programming**

Course Code: **SE3609**

Program: **BSc in Software Engineering**

Department: **Software Engineering**

College: **College of Computing**

Institution: **Umm Al Qura University**

Version: **1.0**

Last Revision Date: **22/04/2025**



## Table of Contents

<b>A. General information about the course:</b> .....	3
<b>B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods</b> .....	4
<b>C. Course Content</b> .....	5
<b>D. Students Assessment Activities</b> .....	6
<b>E. Learning Resources and Facilities</b> .....	6
<b>F. Assessment of Course Quality</b> .....	7
<b>G. Specification Approval</b> .....	7



## A. General information about the course:

### 1. Course Identification

1. Credit hours: (3)

2. Course type

A.  University  College  Department  Track  Others

B.  Required  Elective

3. Level/year at which this course is offered: ( 3<sup>rd</sup> year/ 5<sup>th</sup> or 6<sup>th</sup> level) or ( 4<sup>th</sup> year/8<sup>th</sup> level)

4. Course General Description:

This course considers two closely related topics, Concurrent Systems and Distributed Systems, over 15 lectures. The aim of the first half of the course is to introduce concurrency control concepts and their implications for system design and implementation. The aims of the latter half of the course are to study the fundamental characteristics of distributed systems, including their models and architectures; the implications for software design; some of the techniques that have been used to build them; and the resulting details of good distributed algorithms and applications.

5. Pre-requirements for this course (if any):

CS2016 - Operating Systems

6. Co-requisites for this course (if any):

7. Course Main Objective(s):

By the end of this course, students will:

1. Understand the theoretical and practical aspects of concurrency and parallelism.
2. Gain proficiency in multi-threaded programming and inter-thread communication.
3. Be able to design and implement distributed systems using client-server architectures, sockets, and RMI.
4. Develop a strong foundation in networking protocols and event-based programming for distributed computing.
5. Apply communication patterns effectively in real-world distributed system scenarios.



## 2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	60	100%
2	E-learning	0	0
3	Hybrid <ul style="list-style-type: none"> <li>• Traditional classroom</li> <li>• E-learning</li> </ul>	0	0
4	Distance learning	0	0

## 3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	30
2.	Laboratory/Studio	30
3.	Field	0
4.	Tutorial	0
5.	Others (specify)	0
<b>Total</b>		<b>60</b>

## B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
<b>1.0</b>	<b>Knowledge and understanding</b>			
1.1	Understanding the characteristics and mechanisms of concurrent programs and managing threads	K1	Lectures, Labs	Assignment, Projects, Exams
1.2	Understand the ISO/OSI communication protocol stack.	K1	Lectures, Labs	Assignment, Projects, Exams



Code	Course Learning Outcomes	Code of PLOs aligned with the program	Teaching Strategies	Assessment Methods
1.3	Understand the characteristics of distributed programs;	K1	Lectures, Labs	Assignment, Projects, Exams
1.4	Understand and use RMI to build distributed object-oriented programs	K1	Lectures, Labs	Assignment, Projects, Exams
1.5	Understand the basics of event-based programming	K1	Lectures, Labs	Assignment, Projects, Exams
<b>2.0</b>	<b>Skills</b>			
2.1	Understand and apply primitive methods to create and manage threads	S1	Lectures, Labs	Assignment, Projects, Exams
2.2	Understand and apply techniques that prevent errors like Race Conditions, Deadlock and Starvation.	S1	Lectures, Labs	Assignment, Projects, Exams
2.3	Understand and implement the main inter-thread communication paradigms.	S2	Lectures, Labs	Assignment, Projects, Exams
<b>3.0</b>	<b>Values, autonomy, and responsibility</b>			
3.1	students will become autonomous in recognizing problems specifically connected with concurrency and distribution, and in choosing the proper techniques to obtain the desired behavior from concurrent and distributed software systems.	V1	Group Discussions, Guest Lectures	Reflection Essays on Ethical Scenarios.

### C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to Concurrency characteristic and mechanism	4
2.	Memory and file management	4





3.	Process management and multithreading	4
4.	Inter-thread communication paradigm	4
5.	Concurrency and parallsm	8
6.	Mechanism for multi-thread programming	8
7.	The ISO/OSI standard and communication protocols	4
8.	The Internet Protocols	4
9.	TCP and UDP	4
10.	Introduction to distributing system (client – server architecture)	4
11.	Implementing of distributed system through socket	4
12.	Implementing of distributing system through RMI	4
13.	Event-based program	4
14.	Communication patterns for distributed system	4
15.	Revision	4
<b>Total</b>		<b>60</b>

## D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Assignments and Quizzes	2-14	15
2.	Projects	2-14	15
3.	Practicals	2-14	10
4.	Mid Term	7	20
5.	Final Exam	16-17	40

\*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

## E. Learning Resources and Facilities

### 1. References and Learning Resources

#### Essential References

- Matloff, N. (n.d.). *Programming on parallel machines: GPU, multicore, clusters and more*. University of California, Davis. Retrieved from <http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>
- Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of massive datasets* (3rd ed.). Cambridge University Press. Retrieved from <http://mmds.org>



<b>Supportive References</b>	<ul style="list-style-type: none"> <li>Anenbaum, A. S., &amp; Bos, H. (2015). <i>Modern operating systems</i> (4th ed.). Pearson. ISBN 978-0133591620.</li> <li>Tanenbaum, A. S., &amp; van Steen, M. (2017). <i>Distributed systems: Principles and paradigms</i> (2nd ed.). CreateSpace Independent Publishing Platform. ISBN 978-1530281756.</li> <li>Cachin, C., Guerraoui, R., &amp; Rodrigues, L. (2011). <i>Introduction to reliable and secure distributed programming</i> (2nd ed.). Springer. ISBN 978-3642152597.</li> </ul>
<b>Electronic Materials</b>	
<b>Other Learning Materials</b>	

## 2. Required Facilities and equipment

Items	Resources
<b>facilities</b> (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Traditional Classroom
<b>Technology equipment</b> (projector, smart board, software)	Multimedia Projector
<b>Other equipment</b> (depending on the nature of the specialty)	N/A

## F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	Students	Direct, Indirect
Effectiveness of Students' assessment	Faculty, Peer reviewer	Direct, Indirect
Quality of learning resources	Faculty, Course coordinator	Direct, Indirect
The extent to which CLOs have been achieved	Course coordinator, Program management committee	Direct
Other		

**Assessors** (Students, Faculty, Program Leaders, Peer Reviewers, Others (specify))

**Assessment Methods** (Direct, Indirect)

## G. Specification Approval

<b>COUNCIL /COMMITTEE</b>	<b>SOFTWARE ENGINEERING DEPARTMENT COUNCIL</b>
<b>REFERENCE NO.</b>	<b>THE 17<sup>TH</sup> MEETING FOR THE ACADEMIC YEAR 1446H</b>
<b>DATE</b>	<b>22/04/2025</b>

