COURSE OUTLINE

Course Code: SCSJ4423
Course Name: Real-Time Software Engineering
Total Contact Hours: 42 hours
Course Pre-requisite: Software Engineering (SCSJ1203)

SYNOPSIS

This course equips the students with knowledge in embedded real-time systems and real-time software development particularly on how software engineering approaches assist real-time software development. The knowledge unit for this course area emphasize the following topics; real-time concepts, embedded real-time development methodologies, real-time operating systems, embedded real-time hardware fundamental and real-time analysis. The objective of this course is to introduce students with key software engineering practices in real-time software development and give practical experience to the students in developing embedded real-time software using appropriate software methods and tools.

LEARNING OUTCOMES

By the end of the course, students should be able to:

No.	Course Learning Outcome	Programme Learning Outcome(s) Addressed	Assessment Methods
1.	Explain the issues and basic concepts of real-time and real-time software development.	PO1 (C2, P2, A2) PO3 (C3, P3, A2)	Q, A, T, F
2.	Apply appropriate software engineering methods and tools in developing embedded real-time software.	PO3 (C4, A3, P2)	Pb, A, T, PR, F
3.	Analyze the timing performance of a real-time software design using real-time schedulabity analysis tools.	PO5 (CTPS1 – CTPS3) PO6 (TS1 – TS3)	Pb, Pr, F
4.	Work in a team to apply real-time software engineering knowledge in developing a medium program as a group mini project.		Pb, PR, A, Pr, Peer
	(T – Test ; Q – Quiz; Pb - Problem-Based; A – Assignment; Peer – Peer assessment; PR – Project ; Pr – Presentation, F – Final Exam)		

STUDENT LEARNING TIME

Teaching and Learning Activities			Student Learning Time (hours)	
Face to face Learning	• Lecturer Centered	Lecture	14	
	• Student Centered	- Practical/Lab/Tutorial	4	
		- Student Centered Activity	24	
	• Others		0	
	Sub Total			42
Self Learning	• Non Face to face or Student Centered Learning (SCL)		34	

	• Revision	14	
	• Assessment Preparation	17	
	• Others	0	
	Sub Total		55
Formal Assessment	• Continuous Assessment	10	
	• Final Examination	3	
	• Others	0	
	Sub Total		13
TOTAL SLT			**120**

TEACHING METHODOLOGY

Lecture and Discussion, Problem-Based Learning, Mini Project, Presentation, Independent Study

WEEKLY SCHEDULE

Week	Topics	Activities/hours
Week 1	**1.0 Introduction to Real-Time Systems** 1.1 Categories of computer systems 1.2 Embedded real-time systems 1.3 Characteristics and challenges	Lecture : 1 Student Centred Learning: 2
Week 2	**2.0 Real-Time Software Overview** 2.1 Challenges in software engineering of ERT software 2.2 Software Life Cycle 2.3 Software engineering essential for ERT	Lecture : 1 Student Centred Learning: 2 Assessment: Problem solving 1
Weeks 3 & 4	**3.0 Requirement Specification and Design** 3.1 Requirements stage 3.2 Prototyping 3.3 Design fundamentals 3.4 Elements of modular design 3.5 Design Models – functional, OOD & data flow	Lecture : 2 Student Centred Learning: 4 Assessment: Assignment 1 Assessment: Problem solving 2
Weeks 5 & 6	**4.0 Concurrency** 4.1 What & why concurrency 4.2 Concepts of concurrency 4.3 Sequential and Concurrency programming 4.4 Real-time & other application areas	Lecture : 3 Student Centred Learning: 6 Assessment: Problem solving 3
Weeks 7	**Test 1**	Assessment: Test 1

Weeks 8	**5.0 Concurrent Programming** **5.1** Construct **5.2** Concurrent C Language **5.3** BACI C-- **5.4** Simulator Environment **6.0 Real-Time Operating Systems** 6.1 Basic features 6.2 Scheduling concept	
Weeks 9	**6.0 Real-Time Operating Systems** 6.3 Control of share resources 6.4 Inter-task communication 6.5 Memory management 6.6 MicroC/os RTOS	Lecture : 2 Student Centred Learning : 4 Assessment: Problem solving 4
Week 10-11	**7.0 Real-Time Analysis** 8.1 Schedulability analysis 8.2 Scheduling policies 8.3 Designing with rate-monotonic analysis	Lecture : 2 Student Centred Learning : 4 Assessment: Problem solving 5
Weeks 12-13	**6.0 Hardware Fundamental for Software Engineering** 6.1 Fundamental of microprocessor based systems 6.2 Input-output interfacing technique 6.3 Embedded real-time robotics case study	Lecture : 1 Student Centred Learning : 2 Assessment: Mini Project
Week 14	**Mini Project Presentation** **Revision**	Student Centred Learning : 3
Week 15 **Week 16**	**STUDY WEEK** **EXAMINATION WEEK**	

REFERENCES :

1. Cooling J. (2003). "Software Engineeing for Real-Time Systems", Addison-Wesley.
2. Robert Oshana and Mark Kraeling (2013) Software Engineering for Embedded Systems Methods, Practical Techniques, and Applications, Newnes.
3. Burns A., Wellings A. J. (2001), "Real-time Systems and Programming Languages", Second Edition, Addison Wesley, UK.
4. Ben-Ari M., (2006). Principles of Concurrent and Distributed Programming, Addison-Wesley, England:.
5. Douglass, Bruce Powel (2007), "Real time UML workshop for embedded systems", Newness, Burlington.
6. Hermann Kopetz, B. P. (2011). "Real-Time Systems". Second edition, London: Springer.

GRADING

No.	Assessment	Number	% each	% total
1	Assignments	1	10%	10
2	Quiz	2	2.5%	5
3	Project	1	13%	13
4	In-Class Problem Solving	5	2%	10
5	Peer Assessment	1	2%	2
6	Test 1	1	20%	20
7	Final Exam	1	40%	40
	Overall Total			**100**