

EMBRY-RIDDLE AERONAUTICAL UNIVERSITY
Department of Computing and Mathematics
COURSE OUTLINE FOR

Course No.: MSE640

Title: Concurrent and Distributed Systems

Cr Hrs: 3

Lecture Hours: 3

Laboratory Hours: 0

COURSE DESCRIPTION:

The objective of this course is to teach principles of software development for concurrent and distributed systems. Specification, design and implementation techniques will be described and illustrated by examples and practical exercises. Principles and practices of concurrent programming, including synchronization and communication issues, and a survey of languages suitable for implementing concurrent solutions will be covered.

Prerequisites: MSE 530

GOALS:

Provide students with practical knowledge and understanding of concurrent and distributed software development. Students will be exposed to the most crucial phases of the development process, including specification, design, implementation, and verification, with emphasis on implementation techniques. In addition, students will learn how to use industrial quality software development tools.

PERFORMANCE OBJECTIVES:

1. Describe the essential elements of concurrent/distributed systems
2. Discuss the major problems in development of concurrent/distributed software
3. Propose and select a method, techniques, and tools to solve a particular application problem
4. Use a selected method to develop and evaluate the specification of a concurrent/distributed application
5. Develop, analyze and evaluate a design for a concurrent/distributed application
6. Use a variety of synchronization and communication concepts to implement concurrent/distributed software
7. Determine the major criteria (in terms of safety and liveness properties) and assess the quality of the implementation
8. Use modern software tools and environments for developing concurrent/distributed systems

Department of Computing and Mathematics
COURSE OUTLINE FOR MSE640, Continued

TEXTBOOK:

M. Ben-Ari, *Principles of Concurrent and Distributed Programming*, Prentice-Hall, 1990.

SUGGESTED SUPPLEMENTAL MATERIALS:

- a. Andrews G., R.A. Olsson, *Concurrent Programming Principles and Practices*, Benjamin/Cummings, 1991.
- b. Andrews G., R.A. Olsson, *The SR Programming Language: Concurrency in Practice*, Benjamin/Cummings, 1993.
- c. Axford T., *Concurrent Programming - Fundamental Techniques for Real-Time and Parallel Software Design*, John Wiley and Sons, 1989

PREREQUISITE KNOWLEDGE BY TOPIC:

1. MSE 530, Specification and Design of Software Systems

	TOPIC	CLASS HOURS	COURSE OBJECTIVES
1.	General Overview	3	Describe the essential elements of concurrent/distributed systems
2.	Software Engineering Principles Applied to Concurrent/Distributed Systems	3	Discuss the major problems in development of concurrent/distributed software.
3.	Formal Approaches to Program Specification and Verification: Temporal Logic, Petri Nets, etc.	6	Propose and select a method, techniques, and tools to solve a particular application problem
4.	Properties of Concurrent Programs: Liveness and Safety	3	Determine the major criteria (in terms of safety and liveness properties) and assess the quality of the implementation

5.	Concurrent and Distributed Program Design: DARTS, HOOD, etc	3	Use modern software tools and environments for developing concurrent/distributed systems
	TOPIC (cont.)	CLASS HOURS	COURSE OBJECTIVES
6.	Concurrency Mechanisms: Multitasking and Multithreading, Tasks' Interaction and Mutual Exclusion	3	Use a variety of synchronization and communication concepts to implement concurrent/distributed software
7.	Task Allocation and Scheduling	3	Determine the major criteria (in terms of safety and liveness properties) and assess the quality of the implementation
8.	Concurrent and Distributed Languages: Ada, Concurrent C++, etc.	3	Use modern software tools and environments for developing concurrent/distributed systems
9.	Shared Variables (Critical Regions, Semaphores) vs. Message Passing (Asynchronous, Synchronous, Rendezvous)	3	Use a variety of synchronization and communication concepts to implement concurrent/distributed software
10.	Selection of Typical Problems: Producer-Consumer, Bounded Buffer, Readers and Writers, Dining Philosophers	3	Use a variety of synchronization and communication concepts to implement concurrent/distributed software
11.	Fault Tolerance and Real-Time Communication	3	Use a variety of synchronization and communication concepts to implement concurrent/distributed software
12.	Review of Applications in Aviation/Aerospace	6	Use modern software tools and environments for developing concurrent/distributed systems. Propose and select a method, techniques, and tools to solve a particular application problem

LABORATORY:

None

COMPUTER USAGE:

Occasional use as a learning tool in the classroom.

GRADING SYSTEM:

Assignments include the following: development of a documentation for an individual project, oral and written reports on selected aspects of concurrent/distributed software development, exercises based on practical examples and concerned with using a variety of implementation techniques. The final evaluation will be based on an individual project, submission of a report/documentation, an in-class presentation, a two-hour comprehensive test, and the final exam.

ESTIMATED CONTENT:

Skills:	25%
Content:	75%