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

Course No.: <u>MSE545</u> **Title: <u>Specification & Design of Real-Time Software</u>**
Cr Hrs: <u>3</u>

Lecture Hours: <u>3</u> **Laboratory Hours: <u>0</u>**

COURSE DESCRIPTION:

This course addresses basic concepts and methods used in software specification and design of concurrent and real-time systems. We shall discuss the characteristics of such systems, the role of the software design in software development. We shall review and compare leading software design methods specifically suited for concurrent and real-time systems. Selected method shall be used in a case study to illustrate the design process and to produce related software engineering artifacts. The course material may require to do research real-time aspects of software design, perform laboratory experiments with the real-time operating systems, work as a team on a prototype implementation, and produce appropriate reports. The course will use extensively Internet tools. For the team project the elements of software process are required to be used.
Prerequisite: MSE500, MSE510

GOALS:

The purpose of the course is to have students understand the software life-cycle considerations for concurrent and real-time systems with particular stress on prototyping, incremental development and variants of spiral model. The students will survey various software design methods to become familiar with available approaches. The course will concentrate on design approaches applicable for real-time systems and will give opportunity to study in depth the notation, process, and apply the knowledge in a simple case study.

PERFORMANCE OBJECTIVES:

As the result of the course instruction the students will be able to:

1. describe specifics of concurrent and real-time applications
2. understand the basic real-time concepts
3. understand the software life-cycle consideration for real-time systems
4. understand specifics of real-time operating systems
5. describe and differentiate structured and object-oriented software design methods for real-time systems
6. understand the conceptual foundation for major design approaches for real-time systems
7. perform task structuring and develop the task architecture

8. use tools for real time software analysis and development

9. understand the schedulability concepts and rate monotonic analysis

10. work as a team developing real-time software artifacts

Department of Computing and Mathematics
COURSE OUTLINE FOR MSE545, Continued

TEXTBOOK

Doing Hard Time - Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns, Addison Wesley, 1999 (ISBN-0-201-49837-5)

SUGGESTED SUPPLEMENTAL MATERIALS:

a. *VxWorks Programmer's Guide - Wind River Systems Manual*, 1984-1995 (in RT Lab and on-line with Tornado)
b. *Meeting Deadlines in Hard Real Time Systems - the Rate Monotonic Approach, L.P.Briand, D.M. Roy; IEEE Computer Society Press, 1999*
c. *A Practitioner's Handbook for Real-Time Analysis, M.H.Klein, T.Ralya, B.Pollak, R.Obenza, M.G.Harbour; Kluwer Academic Publishers, 1993*
d. *Software Design Methods for Concurrent and Real-Time Systems*, H. Gomaa, The SEI Series in Software Engineering, 1993
e. *Software Specification and Design: A Disciplined Approach for Real-Time Systems*, K. Shumate, M. Keller; John Wiley, 1992
f. *Real Time UML - Developing Efficient Objects for Embedded Systems*, second edition, B.P. Douglass, Addison Wesley, 1999
g. *Real-Time Software Systems - An Introduction to Structured and Object-Oriented Design*, J.E. Cooling, PWS Publishing Company, ITP, 1997
h. *Strategies for Real-Time System Specification*, D.J.Hatley, I.A.Pirbhai; Dorset House Publ.,1987
i. *Real Time Structured Methods - System Analysis*, K. Edwards, John Wiley, 1993
j. *Real-Time Object-Oriented Modelling*, B. Selic, G. Gullekson; John Wiley, 1994
k. *Programming Embedded Systems in C and C++, M. Barr; O'Reilly & Associates, 1999*
l. *Real-Time Programming - a Guide to 32-bit Embedded Development, R.Grehen, R. Moote, I.Cyliax; Addison Wesley, 1998*
m. *Real-Time Systems, J.W.S.Liu; Prentice Hall, 2000*
n. *Developing Real-Time Embedded Software - In a Market-Driven Company, K.S.Ellison; John Wiley, 1994*
o. *Real-Time Systems*, C.M. Krishna, K.G. Shin, McGraw Hill, MIT Press, 1997
p. *Avionic Systems Design*, J.R. Newport, CRC Press, 1994
q. *Embedded Systems Building Blocks*, J.J. Labrosse; R&D Publications/Prentice Hall, 1995
r. *Real-Time System Design*, S.T. Levi, A.K. Agrawala; McGraw Hill, 1990,
s. *A Practical Guide to Real-Time Systems Development*, Goldsmith S. Prentice Hall, 1993
t. *Constructing Predictable Real-Time Systems,* Halang W., Stoyenko A. Kluwer Academic Publ, 1993

u. *Object Oriented Software Engineering: A Use Case Driven Approach, Jacobson I, et al, Addison Wesley,1992*
v. *Object-Oriented Design with Ada*, K. Nielsen; Bantam Books, 1992,
w. *An Implementation Guide to Real-Time Programming*, L.Ripps; Prentice Hall, 1989,
x. *Structured Development for Real-Time Systems*, P.T.Ward, S.J.Mellor; Prentice Hall, 1985,
y. An access to the current articles in such magazines as: Embedded Computing, Real Times, Byte, Transactions of ACM, IEEE Computer, Dr. Dobbs Journal, Web resources (http://www.realtime-info.be/). etc.
z. An access to a computer laboratory with a fast, user friendly, high-level language environment and real-time operating systems. The language for the test programs and the project implementation is C/C++.

PREREQUISITE KNOWLEDGE AND TOPICS:

1. Knowledge of software engineering life-cycle concepts and methods, including notation for requirements specification of software systems..
2. Knowledge of computer operations including input/output and interrupt interface.
3. Proficiency in a high-level language programming, specifically C/C++ under UNIX environment (including compilation, linking, using libraries, basics of socket programming, timing, shared memory, message passing, signals, system calls),
4. Familiarity with the operating systems fundamentals (including interprocess communication, input/output, using memory resources, and operating system utilities and functions).

TOPIC	CLASS HOURS	COURSE OBJECTIVES
1. Real-Time Basic Concepts	3	Describe specifics of concurrent and real-time applications. Understand the basic real-time concepts. Understand specifics of real-time operating systems.
2. Software Life-cycle and Process for Real-Time Systems	6	Understand the software life-cycle consideration for real-time systems. Describe and differentiate structured and object-oriented software design methods for real-time systems. Understand the conceptual foundation for major design approaches for real-time systems. Work as a team developing real-time software artifacts.
3. Real-Time Operating System Concepts	3	Understand specifics of real-time operating systems.
4. Requirement Analysis: System and Software	6	Understand the software life-cycle consideration for real-time systems. Describe and differentiate structured and object-oriented software design methods for real-time systems.
5. Software Analysis: Object Structure and Behavior	6	Understand the software life-cycle consideration for real-time systems. Describe and differentiate structured and object-oriented software design methods for real-time systems.

6. Software Architecture and Task Structuring	6	Understand the conceptual foundation for major design approaches for real-time systems. Perform task structuring and develop the task architecture.
7. Software Mechanistic and Detailed Design	3	Understand the basic real-time concepts. Understand the conceptual foundation for major design approaches for real-time systems. Use tools for real time software analysis and development.
8. Rate Monotonic Analysis and Task Scheduling	6	Perform task structuring and develop the task architecture. Understand the schedulability concepts and rate monotonic analysis.
9. System Integration and Testing	3	Work as a team developing real-time software artifacts.
10. Safety Concerns Real Time Systems	3	Use tools for real-time software analysis and development. Understand the schedulability concepts and rate monotonic analysis.

LABORATORY AND COMPUTER USAGE:

Access to a computer laboratory with a fast, user friendly, high-level language host-target development environment and CASE tools. The language for the test programs and the project implementation is C/C++/Ada)

GRADING SYSTEM:

The final evaluation will be based on:

- P – real-time software team project with documentation on an HTML file; presented in the last week of classes (35%)
- T – tests, including set of problems and questions from the material discussed in class, experimented individually, and available in your text and additional readings (open books); administered in class (30%)
- R – individual research paper (10-20 pages, about 1,500 words, in word processing or html both hard and soft copy required) discussing a selected issue of real-time software analysis and design (including in-class presentation); based on compilation of selected few papers – other options are available as e.g. an individual task of installation and testing selected hardware/software component in the RT Lab (20%)
- C – class attendance, initiative with real-time laboratory hardware and software installations, active experimentation with real-time concepts, in-class presentation of accomplishments and lessons learned (15%)

ESTIMATED CONTENT:

Skills: 25 %
Content: 75 %