

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

Course No.: MSE650
Cr Hrs: 3

Title: Software Safety

Lecture Hours: 3

Laboratory Hours: 0

COURSE DESCRIPTION:

The objective of this course is to teach concepts, principles, methods and techniques as related to software safety as a part of overall system safety. Safety critical systems and high integrity systems require additional activities to assure that they are safe. As modern systems are software intensive there is a need to introduce safety analysis and design techniques into the development lifecycle. The field of safety critical systems covers multitude of diverse topics. The course discusses the safety requirements, hazard and risk analyses, fault tolerance, basics of software reliability, and issues of verification, validation, and certification. Various safety standards across application domain and selected tools supporting safety assurance of software products are introduced.

GOALS:

This course provides students with a practical knowledge and understanding of safety issues when specifying, designing, testing, and maintaining a software product as a component of a safety critical system. Students are exposed to the most important concepts of system and software safety, including hazard analysis, levels of integrity, failure modes and effect analysis, nature of faults and fault-tolerant techniques. Rudimentary concepts of reliability engineering are introduced to show the relation between software safety and reliability. Use of mathematical modeling is oriented toward showing contribution of formal methods and models to produce high integrity software systems. Issues of testing verification, validation and certification including applicable industry standards are discussed. Students learn how to use modern engineering tools supporting software safety and reliability assessment.

PERFORMANCE OBJECTIVES:

Upon completion of this course, students should be able to:

1. describe the essential concepts of system safety;
2. understand concepts of integrity, dependability and fail-safe operations;
3. understand basic techniques of hazard analysis
4. classify risks based on the severity and frequency
5. appreciate role of safety assuring activities in the development lifecycle
6. understand and apply the safety analysis as a part of high integrity software lifecycle
7. determine types of faults and understand concept of fault tolerance
8. understand the concept of system and software reliability
9. identify the methods of testing for safety and related verification and validation issues
10. be familiar with software quality safety standards and guidelines
11. be familiar with the causes of software failures leading to system safety violations

Department of Computing and Mathematics
COURSE OUTLINE FOR MSE650, Continued

TEXTBOOK:

Storey, Neil, *Safety Critical Computer Systems*, Addison Wesley Longman 1996.

SUGGESTED SUPPLEMENTAL MATERIALS:

- a. *Software Safety and Reliability*, by Debra Herrmann, IEEE Computer Society, 1999
- b. *Safeware - System Safety and Computers*, by Nancy Leveson, Addison Wesley, 1995
- c. *Software Reliability*, by Hoang Pham, Prentice Hall, Springer, 2000
- d. *Software Reliability - Measurement, Prediction, Application*, by John Musa, Anthony Iannino, Kazuhira Okumoto, McGraw Hill, 1987
- e. *High Integrity System Specification and Design*, by Jonathan Bowen and Michael Hinchey, Springer 1999
- f. *Fault Tolerant Computer System Design*, by Dhiraj Pradhan, Prentice Hall, 1996
- g. *Performance Modeling with Deterministic and Stochastic Petri Nets*, by Christoph Lindemann John Wiley and Sons, 1998

PREREQUISITE KNOWLEDGE AND TOPICS:

1. Familiarity with computer systems operations (hardware and software).
2. Proficiency in college mathematics including discrete math concepts.
3. Familiarity with the computer system concepts and the software engineering lifecycle.

TOPIC	CLASS HOURS	COURSE OBJECTIVES
1. System and software safety concepts	3	Describe the essential concepts of system safety;
2. Safety criteria	3	Understand concepts of integrity, dependability and fail-safe operations
3. Hazard analysis	3	Understand basic techniques of hazard analysis
4. Risk analysis	3	Classify risks based on the severity and frequency
5. Developing safety critical systems	3	Appreciate role of safety assuring activities in the development lifecycle
6. Fault tolerance	3	Determine types of faults and understand concept of fault tolerance
7. System reliability	6	Understand the concept of system and software reliability
8. Software safety and role of formal methods and modeling	6	Understand and apply the safety analysis as a part of high integrity software lifecycle

TOPIC (cont.)	CLASS HOURS	COURSE OBJECTIVES
9. Verification and Validation	3	Identify the methods of testing for safety and related verification and validation issues
10. Case studies on software safety	3	Be familiar with the causes of software failures leading to system safety violations
11. Quality and safety standards and certification	6	Be familiar with software quality and safety standards and guidelines

LABORATORY AND COMPUTER USAGE:

Access to Internet for class material and research. Access to tools supporting software safety and reliability analysis (hazard, fault tree, risk, reliability models, simulations).

GRADING SYSTEM:

The final evaluation is based on four components:

- PROJECT - system/software safety analysis project/case study/tool assessment (individual or team), assigned in 5-6th week of class, to be delivered in a form of formal safety analysis report (additionally an executive summary of project and main results in a form of an HTML file) and presented at the end of term (30%)
- TESTS - tests, including set of problems and questions from the material discussed in class and available in the textbook and additional readings (open books) administered about the last week of term, not interfering with the project presentations (30%)
- RESEARCH - individual research paper with class presentation (15-25 pages, about 2,000 words, in word processing or HTML format, both hard and soft copy required) discussing selected issues of system and software safety related to the class material; the paper shall be based on compilation of few literature positions, not limited to the reading list to be distributed in class - this may be only a starting point (20%)
- CLASS - class attendance, initiative, class discussions in the Q&A period, pop-up short quizzes on the software safety concepts and terms - open books (20%)

ESTIMATED CONTENT:

Skills: 25 %
Content: 75 %