

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

Course No.: SE310
Cr Hrs: 3

Title: Analysis and Design of Software Systems

Lecture Hours: 3

Laboratory Hours: 0

COURSE DESCRIPTION:

This course focuses on the fundamental methods employed in the analysis and design of software systems. Analysis is the process of determining a complete and consistent set of system requirements. Design is the process of producing a system architecture, both logical and physical, and determining an appropriate way to construct the software. The result of these processes is a documented model of the desired system. The student will learn and practice methods appropriate for both object-oriented and procedural systems. Prerequisite: CS 315.

GOALS:

The purpose of this course is to introduce students to specific techniques which are widely used in the analysis and design of software systems. Traditional techniques will be reviewed and current methodologies will be taught and practiced. Emphasis will be placed on the standard notation used to document software requirements and designs. The translation of a documented design to implemented software will be discussed, however, software construction (programming) will not be emphasized in the course.

PERFORMANCE OBJECTIVES:

1. To be able to describe object-oriented and structured (functional) analysis and design processes
2. To be able to describe purpose of models and artifacts
3. To be able to define analysis phases
4. To understand the analysis of requirements
5. To be able to develop and specify conceptual models
6. To be able to develop and document use cases
7. To be able to understand standard UML notation
8. To be able to design and document classes

Department of Computing and Mathematics
COURSE OUTLINE FOR SE310, Continued

9. To be able to develop and document collaborations
10. To be able to develop and document system states
11. To be able to understand the use of design patterns
12. To be able to define and document characteristics of system behavior (contracts)
13. To understand issues with mapping designs to code
14. To understand analysis and design process issues

TEXTBOOK:

Larman, C. (1998). *Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design*. Prentice Hall PTR, Upper Saddle River, NJ. ISBN 0-13-748880-7.

Fowler, M. (2000). *UML Distilled*. (Second Edition). Addison Wesley Longman, Inc. Reading, MA. ISBN 0-201-657838-X.

SUGGESTED SUPPLEMENTAL MATERIALS:

Cooper, J.W. (2000) *Java Design Patterns, A Tutorial*. Addison Wesley Publishing Co., New York. ISBN 0-201-48539-7

PREREQUISITE KNOWLEDGE BY TOPIC:

1. Basic object oriented programming.
2. Basic data structures and algorithms.

TOPIC	CLASS HOURS	COURSE OBJECTIVES
1. Overview of the Analysis and Design Process	2	Describe purpose of analysis and design phases. Describe analysis and design process and artifacts.
2. Requirements Analysis	4	Describe the purpose of requirements analysis. Understand basics of widely used methodologies for analysis of requirements (procedural and OO)
3. Use Cases	3	Understand and demonstrate use case requirements analysis methodology.

4.	Modeling with UML	4	Understand how to read and construct basic UML diagrams.
5.	Class Design	5	Describe basic elements of OO class design. Understand and demonstrate use of inheritance, information hiding, polymorphism and appropriate class cohesion/coupling.
6.	Collaborations, Event Sequencing and States	5	Understand use of multiple static and dynamic views of a design.
7.	Design Patterns	5	Describe usefulness of design patterns and understand the basic software patterns.
8.	Design Documentation	3	Describe and understand design documentation for OO and procedural designs.
9.	Post-Design Issues	2	Understand tools and processes for translating design to code and hardware/software implementation.

LABORATORY:

No closed laboratory sessions. Open laboratories used for homework assignments.

COMPUTER USAGE:

The chief resource is a laboratory of Windows NT Workstations and various design tools, compilers, word processors and spreadsheet applications.

GRADING SYSTEM:

Varies by instructor

ESTIMATED CONTENT:

Skills: 20%
Content: 80%