

**SAN DIEGO COMMUNITY COLLEGE DISTRICT  
CITY, MESA, AND MIRAMAR COLLEGES  
ASSOCIATE DEGREE COURSE OUTLINE**

**SECTION I****SUBJECT AREA AND COURSE NUMBER:** Computer and Information Sciences 210**COURSE TITLE:** System Analysis and Design**Units: 3**  
Grade Only**CATALOG COURSE DESCRIPTION:**

This course is an introductory, experiential study of the phases of the object-oriented software development life cycle (OOSDLC), including: stakeholder and requirements analysis; use cases development; software architecture; project management; user interface considerations; interactive and prototyping methodology; component construction; quality assurance; and configuration management. This course is intended for students seeking advanced knowledge and applications in Computer and Information Sciences.

**REQUISITES:****Advisory:**

ENGL 101 with a grade of "C" or better, or equivalent or Assessment Skill Level W6/R6

**FIELD TRIP REQUIREMENTS:** May be required**TRANSFER APPLICABILITY:** Associate Degree Credit & transfer to CSU and/or private colleges and universities UC Transfer Course List**TOTAL LECTURE HOURS:** 48 - 54**TOTAL LAB HOURS:****STUDENT LEARNING OBJECTIVES:**

Upon successful completion of the course the student will be able to:

1. Describe the phases and iterations of the object-oriented software development life cycle (OOSDLC).
2. Develop all the artifacts (documents etc.) associated with each phase of this OOSDLC.
3. Develop a stakeholder and requirements analysis, including use cases (detailed descriptions of usage scenarios).
4. Explain the transitions between phases, including the use of multiple iterations.
5. Develop a simple project plan.
6. Develop unit test, implementation, and quality assurance specifications.
7. Identify the key aspects of software quality assurance, including usability, reusability, reliability, flexibility, scalability, maintainability, security.
8. Describe the requirements for appropriate configuration and change management.
9. Draw a simple Unified Modeling Language (UML) diagram set.
10. Recognize software development best practices.

**SECTION II**

## 1. COURSE OUTLINE AND SCOPE:

### A. Outline Of Topics:

The following topics are included in the framework of the course but are not intended as limits on content. The order of presentation and relative emphasis will vary with each instructor.

I. The following topics will be included in the framework of the course but are not intended as limits on content. The order of presentation and relative emphasis will vary with each instructor.

- A. Introduction to Software Development Methodologies
  - 1. Software development life cycle
  - 2. Waterfall Method
  - 3. Enhanced Waterfall Method
  - 4. Iterative or Phased Development
  - 5. Object-Oriented Software Development Lifecycle (OOSDLC), based on the Rational Unified Process (RUP)
  - 6. Software Development Best Practices
- B. Introduction to Object-Orientation
  - 1. Components
  - 2. Attributes and Methods - integration of processing and data handling
  - 3. Benefits: inheritance, encapsulation, software reuse
  - 4. Object-Oriented Systems Analysis and Design
  - 5. Object-Oriented Tools and Languages
- C. Stakeholder and Requirements Analysis
  - 1. Understanding stakeholders and actors
  - 2. Determining requirements using use cases
  - 3. Documenting requirements in the requirements Specification
- D. Interactive Development (Iterations)
  - 1. Prototyping
  - 2. Phases development
  - 3. Documenting unit tests in the Unit Test Specification
- E. Project Planning and Management
  - 1. Translating use cases into deliverables
  - 2. Establishing and tracking milestones
  - 3. Resource planning
  - 4. Overview of Microsoft Project
- F. Implementation
  - 1. Translating use cases into functions and features
  - 2. Modular design and other quality considerations
  - 3. Design reviews-inspections
  - 4. Documenting implementation in the implementation specification
  - 5. Unit testing
  - 6. Integration
- G. Software Quality Assurance
  - 1. Software development best practices
  - 2. Involvement in all OOSDLC phases
  - 3. QA Standards and Certifications: ISO, CMM, IEEE,...
  - 4. Developing a quality assurance specification
- H. Change Management
  - 1. Incidence Management: reporting, tracking, handling, closing
  - 2. Software source management
- I. Configuration Management
  - 1. Software release management
  - 2. Alpha and Beta sites
  - 3. Data and test management
  - 4. Environment management
- J. Unified Modeling Language
  - 1. Software modeling
  - 2. Round-trip engineering
  - 3. Syntax and Artifacts
  - 4. Example products: Rational Rose, TogetherSoft Solo

## **B. Reading Assignments:**

Reading assignments are required and may include but, are not limited to, the following:

- I. Schneider, Geri and Jason P. Winters. Applying Use Cases - A Practical Guide. Massachusetts, Addison-Wesley Longman, Inc. 1998
- II. Royce, Walker. Software Project Management. Massachusetts, Addison-Wesley Longman, Inc., 1998
- III. Web sites: <http://www.rational.com>; <http://www.togethersoft.com>

## **C. Appropriate Assignments that Demonstrate Critical Thinking:**

Critical thinking assignments are required and may include, but are not limited to, the following:

- I. All assignments and in-class exercises are oriented towards active, critical participation by students, either individually or as teams. These activities include:
- II. Research papers that compare and contrast key aspects of software quality assurance for two or more systems.
- III. Written analysis of system requirements for appropriate configuration and change management.
- IV. In-class (oral) presentations comparing and contrasting best practices in software development.

## **D. Appropriate Outside Assignments:**

Outside assignments may include, but are not limited to, the following:

- I. Short compare and contrast paper on two cases from online research about the use of a formal software development process, such as the Rational Process.
- II. Oral presentation on the benefits of object orientation.
- III. Short paper on the various industry standards for software quality assurance.

## **E. Writing Assignments:**

Writing assignments are required and may include, but are not limited to, the following:

- I. Semester project paper reflecting a "real-life" case study application of the principles learned in the course (Example: Develop the specifications for the implementation of an information technology solution for a chain of dental offices)
- II. Requirements, Implementation, Quality Assurance, and Unit Test Specifications applied to the semester project.
- III. Simple project plan (resources, milestones, risk analysis, critical path)
- IV. Examples of use cases (Example: Setting up an appointment), optionally part of the Requirements Specification for the semester project.

## **2. METHODS OF EVALUATION:**

A student's grade will be based on multiple measures of performance unless the course requires no grade. Multiple measures may include, but are not limited to, the following:

- I. Qualitative evaluation is based on: In-class and homework activities, including a major semester project. Class attendance and participation
- II. Quantitative evaluation is based on: Midterm exam (multiple choice, fill-in-the-blank, true-false, short answer) Final exam (multiple choice, fill-in-the-blank, true-false, short answer)

## **3. METHODS OF INSTRUCTION:**

Methods of instruction may include, but are not limited to, the following:

- \* Distance Education
- \* Other (Specify)
- \* Methods of instruction are to be determined by each instructor and may include, but are not limited to the following:
- \* 1. In-class lecture

- \* 2. Modern case studies
- \* 3. Group exercises (see "semester project" above - opportunities for students to work together in groups of two or more and learn to collaborate on the completion of various assignments).
- \* 4. The development of artifacts approximating real-life use.
- \* 5. Guest speakers from the software computer industry community.

#### **4. REQUIRED TEXTS AND SUPPLIES:**

Textbooks may include, but are not limited to:

##### **TEXTBOOKS:**

1. Dennis, Alan and Barbara Haley Wixom. Systems Analysis and Design, Current ed. New York: John Wiley and Sons, 1999,
2. Kruchten, Philippe. The Rational Unified Process, An Introduction-Second Edition, 2nd ed. Massachusetts, Addison-Wesley Longman, Inc., 2000,

##### **MANUALS:**

##### **PERIODICALS:**

##### **SOFTWARE:**

##### **SUPPLIES:**

**ORIGINATOR:** Rose LaMuraglia

**CO-CONTRIBUTOR(S)**

**DATE:** 10/13/2005