Las Positas College
3000 Campus Hill Drive
Livermore, CA 94551-7650
(925) 424-1000
(925) 443-0742 (Fax)

Course Outline for CS 47

CAPSTONE PROJECT

Effective: Fall

I. CATALOG DESCRIPTION:
CS 47 — CAPSTONE PROJECT — 3.00 units

This is the last course in the Computer Programming (CP) degree or certificate sequence. You will work in teams and write a client-driven work-like project requiring planning, writing, execution and presentation. The project will require the development of a large application selected by each team. According to client specifications, each team will define their project and break it down into parts. Each member of the team will write one of the parts. The team will prepare a written and oral report to present their project. This course will require the use of all of the programming and systems analysis skills developed in previous courses and will serve as a means of demonstrations of mastery of program competencies.

1.00 Units Lecture 2.00 Units Lab

Prerequisite
CIS 60 - Systems Analysis and Design

CS 43 - Professional Communications
and

CS 20 - Advanced Programming with Data Structures/C++
or

CS 33 - Advanced C++ Programming
or

CS 34 - Advanced Java Programming
or

CS 35 - Advanced Visual Basic Programming
or

CS 36 - Windows and MFC Programming
or

CS 44 - Advanced Web Programming
with a minimum grade of C

Strongly Recommended
CS 45 - Database Programming

Grading Methods:
Letter or P/NP

Discipline:

	MIN
Lecture Hours:	18.00
Lab Hours:	108.00
Total Hours:	126.00

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 1

III. PREREQUISITE AND/OR ADVISORY SKILLS:
Before entering the course a student should be able to:

A. CIS60
B. CS43
C. CS20

D. CS33
E. CS34
F. CS35
G. CS36
H. CS44
    1. Present the elements and features of the website development environment
    2. Explain and use the design process for eCommerce
    3. Write, compile, test and debug XML/XSL programs
    4. Define and use user interfaces
    5. Define and develop class modules
    6. Develop and use event-driven programs (HTML, form tags for data entry
    7. Database Design
    8. Explain database design concepts and the role of database components
    9. Model data and design XML database structure
    10. Explain the use of XML databases and information in the business environment
    11. Systems Design
    12. Specify major subsystems and interfaces
    13. Perform feasibility studies of design alternatives (e.g. Java, JavaScript, ActiveX, etc.)
    14. Prepare and conduct design reviews
    15. Technical Documentation
    16. Write in a concise and precise form appropriate for technical documentation
    17. Explain and use the processes and techniques of XML/XSL technical documentation
    18. Prepare materials written to convey specific technical XML/XSL problems, their related issues, and their solutions
    19. Adhere to XML/XSL documentation industry and organization guidelines and standards
    20. Testing and Debugging
    21. Select debugging and testing methodology, and develop comprehensive and systematic test plan
    22. Test programs, and document errors and solutions
    23. User Interface Design
    24. Recognize a wide range of problems, and assess their impact on the system
    25. Use a wide range of troubleshooting methods and tools to isolate problems

Before entering this course, it is strongly recommended that the student should be able to:

A. CS45

IV. MEASURABLE OBJECTIVES:

Upon completion of this course, the student should be able to:

A. Technical Competencies-Database Design
    1. Model data and design database structure
    2. Design, create and utilize relational databases
    3. Create and edit tables, develop complex queries and create reports
    4. Create and customize forms and reports
    5. Explain the use of databases and information in the business environment
    6. Develop database business applications
B. Systems Analysis
    1. Gather data to identify client requirements
    2. Interpret and evaluate requirements for completeness, relevance, accuracy, and consistency
    3. Identify time, technology and resource constraints
    4. Resolve conflicts between requirements and constraints, and negotiate resolution with client
    5. Develop high-level systems and functional specifications
    6. Perform data flow, event analysis and object modeling
    7. Develop concepts including alternatives and prepare a cost/benefit estimate for each option
    8. Identify risks and their impact on the overall project
    9. Define general scope of work to meet requirements and constraints
    10. Establish measurable performance requirements
    11. Develop business process and logical data model
C. Systems Design
    1. Specify major subsystems and interfaces
    2. Develop detail design specifications
    3. Select design methodology and tools
    4. Identify physical requirements for systems implementation
    5. Prepare and conduct design reviews
    6. Develop test plan
D. Technical Documentation
    1. Write in a concise and precise form appropriate for technical documentation
    2. Explain and use the processes and techniques of technical documentation
    3. Record system specifications accurately and completely
    4. Organize the data in the systems repository
E. Testing and Debugging
    1. Select debugging and testing methodology, and develop comprehensive and systematic test plan
    2. Select program debugging tools and techniques
    3. Develop testing procedures
    4. Conduct tests in the most efficient way
    5. Test programs, and document errors and solutions
    6. Select testing tools and develop test system
    7. Perform system integration testing
    8. Evaluate the effectiveness of the testing plan and procedures on a continuous basis
    9. Assess overall product effectiveness and performances, and perform summative evaluation
F. User Interface Design
    1. Define the requirements for the user interface
    2. Detail the development process and methods best suited for the project
    3. Develop user interface schema to meet user requirements
    4. Participate and conduct design and development reviews
    5. Construct user interfaces for flexibility and adaptability
    6. Perform user interface tests, and troubleshoot problems
    7. Follow organization and industry standards for development
G. Problem Solving
    1. Recognize a wide range of problems, and assess their impact on the system
    2. Use a wide range of troubleshooting methods and tools to isolate problems

3. Select the appropriate approach to identify causes of the problem based on the given situation

H. Project Management
1. Describe the main steps and organizational issues in project management
2. Evaluate project requirements, and clearly define and articulate project scope and goals
3. Develop detailed task list and analyze relationships between separate tasks and the overall project
4. Develop general project flow chart, identifying critical tasks and task interdependencies
5. Identify project time, personnel, budget and equipment requirements
6. Identify, evaluate and monitor risks throughout the project, and prepare contingency plans
7. Identify critical milestones and project performance, budgets and the use of resources
8. Participate in and actively contribute to project reviews
9. Document and report project status in a timely manner using appropriate channels
10. Work effectively within the system and with members of the team and organization

I. Task Management
1. Break down projects and activities into a series of tasks
2. Identify task priorities and interdependencies, and organize in a logical sequence
3. Estimate time and resources necessary to complete specific tasks
4. Develop realistic schedule and work processes to accomplish assigned tasks
5. Recognize and resolve conflicts in the use of resources or in goals between separate tasks
6. Secure or request resources in a timely manner to accomplish tasks on schedule
7. Monitor task performance and completion against project plan and standards
8. Make process improvements and adjustments as tasks progress
9. Evaluate the impact of one's work on the rest of the project and team

J. Business Skills, Professionalism, & Non-technical Competencies- Business Communications
1. Describe and apply the purpose and different uses of communication in business and industry
2. Identify the appropriate communication format for a specific purpose and situation
3. Communicate effectively using a wide range of communication formats
4. Use clear, focused, specific, and grammatically correct language and terminology
5. Balance visual and verbal elements, and text in written communication and presentations

K. Professional Development/ Self Learning
1. Recognize what skills and knowledge are needed to work on a specific project or in a specific environment
2. Assess personal skills and knowledge against identified needs
3. Assess objectively one's strengths and current knowledge to develop new skills
4. Leverage team resources to further one's own skills and knowledge
5. Evaluate the portability of one's skills to new areas of application

L. Professional Environment
1. Exhibit appropriate work habits and attitudes
2. Take pride in one's work and assume responsibility for personal actions
3. Accept constructive criticism
4. Display a positive attitude, project a professional image, and foster a productive environment
5. Identify characteristics of effective leaders, and demonstrate initiative and leadership skills
6. Explain the need and benefits of maintaining a strong professional network
7. Recognize and effectively use opportunities and forums to establish professional relationships
8. Create and develop successful professional relationships

M. Team Work
1. Explain and apply different team processes, roles and group dynamics, their purposes, advantages and disadvantages
2. Use effectively a variety of listening, communication and interactive styles and strategies, and recognize their appropriateness depending on the team environment and goals
3. Respect and work collaboratively with differences in backgrounds, opinions, and communication styles
4. Recognize and respect cultural, ethnic, and linguistic diversity
5. Recognize and leverage strengths in one's self and others to further the goals of the team
6. Create an environment that supports risk taking in freely disclosing ideas and opinions
7. Identify and support productive ideas and processes that contribute to the goals of the team
8. Function effectively in different roles within the team, and show flexibility in accepting others' leadership
9. Manage conflicts in a productive manner, and work collaboratively to set and accomplish team goals
10. Learn from and teach other members of the team

V. CONTENT:

A. Teams
1. Team Roles
2. Definition of Roles
3. Types of Teams

B. Creation of teams
1. Identifying members
2. Identifying members strengths and weaknesses

C. Selecting a project
1. Recognizing and defining the problem
2. Define the scope of the project
3. Requirements Analysis
4. Software selection
5. Hardware selection

D. Design the solution
1. Identify the project parts
2. Assigning project parts to members

E. Project Implementation
1. Project timeline and milestones
2. Evaluating project at each milestone
3. Milestone documentation and testing

F. Writing project report

G. Creating the project presentation

H. Project Presentation

I. Assessing the Project
1. Presentation assessment
2. Written team report
3. Team assessment
4. Members assessment

VI. METHODS OF INSTRUCTION:

A. **Lecture** -

B. **Projects** - Programming Project in Teams

C. **Discussion** -
D. **Demonstration** -

VII. TYPICAL ASSIGNMENTS:
A. Design a plan for software and hardware selection for new firm B. Work with a bank to develop a database system to capture needed data on customers, average deposit size, and other key characteristics. C. Develop a simple learning game application for an elementary school. D. Develop a program to track key data on global warming for the federal government.

VIII. EVALUATION:
A. **Methods**

B. **Frequency**

1. Frequency of evaluation
    a. Recommend 2 or 3 exams plus final examination
    b. Recommend programming assignment to cover each topic within course content. Contents can be combined.
2. Types of Exam Questions
    a. Identify the timeline components to a programming project.
    b. List the roles of the team members and give a brief description of each.
    c. Identify the major components to a project presentation.

IX. TYPICAL TEXTS:
1. Ashley Friedlein *Web Project Management.*, Morgan Kaufmann Publisher (Academic Press) , 2001.
2. Mary Ellen Guffey *Essentials of Business Communication.* 5th ed., South-Western College Publishing, 2000.
3. Gary Martin *Welcome to the Professional World.* 2nd ed., Mocha Enterprises, 2000.

X. OTHER MATERIALS REQUIRED OF STUDENTS:
A. Access to a software compiler (C++, JAVA, PASCAL, etc).
B. Access to internet.