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

Course Outline for CS 45

DATABASE PROGRAMMING

Effective: Fall

I. CATALOG DESCRIPTION:
CS 45 — DATABASE PROGRAMMING — 4.00 units

This is a programming course that goes beyond mere “desktop” database management. Participants explore dynamic applications that interact with a database using client-side scripts; server-side scripts, and compiled server programs. Learn databases concepts, relational database principles, and Structured Query Language (SQL).

3.00 Units Lecture 1.00 Units Lab

Prerequisite
CS 2 - Computing Fundamentals II
or

CS 30 - C++ Programming
or

CS 31 - Java Programming
or

CS 32 - Visual Basic Programming
with a minimum grade of C

Grading Methods:
Letter or P/NP

Discipline:

	MIN
Lecture Hours:	54.00
Lab Hours:	54.00
Total Hours:	108.00

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. CS2
B. CS30
1. Present the elements and features of the development environment
2. Explain and use the design process
3. Define and use functions and storage classes
4. Define and explain trends in programming standards
5. Write, compile, test and debug programs
6. Present the characteristics of object-oriented programming
7. Define and use data types and variables
8. Define and use multi-dimensional arrays
9. Define and use constructor and destructor functions
10. Define and use function overloading
11. Define and use operator overloading
12. Define and use inheritance mechanisms in OOP
13. Define and use user interfaces
14. Define and use file I/O
15. Define and develop class modules
16. Develop and use event-driven programs
17. Use pointers to data, objects and functions
18. Use dynamic memory allocation
19. Systems Analysis
20. Develop high-level systems and functional specifications
21. Define general scope of work to meet requirements and constraints

22. Systems Design
23. Specify major subsystems and interfaces
24. Develop detail design specifications
25. Select design methodology and tools
26. Identify maintenance requirements
27. Technical Documentation
28. Write in a concise and precise form appropriate for technical documentation
29. Explain and use the processes and techniques of technical documentation
30. Record system specifications accurately and completely
31. Testing and Debugging
32. Select debugging and testing methodology, and develop comprehensive and systematic test plan
33. Develop testing procedures
34. Conduct tests in the most efficient way
35. Test programs, and document errors and solutions
36. User Interface Design
37. Define the requirements for the user interface
38. Detail the development process and methods best suited for the project
39. Develop user interface (UI) to meet user requirements
40. Test UIs
41. Problem Solving
42. Recognize a wide range of problems, and assess their impact on the system
43. Use a wide range of troubleshooting methods and tools to isolate problems
44. Select the appropriate approach to identify causes of the problem based on the given situation
45. SPECIFIC: These outcomes are detailed specifically for this course. Upon completion of the course students should be able to: Give an overview of the evolution and present state of programming languages
46. Describe and employ basic principles of software engineering.
47. Define and use abstract data types in program applications.
48. Define and employ overloading of functions and operators.
49. Write functions implementing iteration.
50. Define and illustrate encapsulation, inheritance and polymorphism in C++.
51. Write programs that use file I/O techniques.

C. CS31

D. CS32

1. present the elements and features of the development environment;
2. explain and use the design process;
3. define and use functions;
4. define and explain trends in programming standards;
5. write, compile, test and debug programs;
6. present the characteristics of object-oriented programming;
7. define and use data types and variables;
8. define and use multi-dimensional arrays;
9. define and use user interfaces;
10. define and use file I/O;
11. define and develop class modules;
12. develop and use event-driven programs;
13. Systems Analysis
14. develop high-level systems and functional specifications;
15. define general scope of work to meet requirements and constraints;
16. Systems Design
17. specify major subsystems and interfaces;
18. develop detail design specifications;
19. select design methodology and tools;
20. identify maintenance requirements;
21. Technical Documentation
22. write in a concise and precise form appropriate for technical documentation;
23. explain and use the processes and techniques of technical documentation;
24. record system specifications accurately and completely;
25. Testing and Debugging
26. select debugging and testing methodology, and develop comprehensive and systematic test plan;
27. develop testing procedures;
28. conduct tests in the most efficient way;
29. test programs, and document errors and solutions;
30. User Interface Design
31. define the requirements for the user interface;
32. detail the development process and methods best suited for the project;
33. develop user interface (UI) to meet user requirements;
34. test Uis;
35. Problem Solving
36. recognize a wide range of problems, and assess their impact on the system;
37. use a wide range of troubleshooting methods and tools to isolate problems;
38. select the appropriate approach to identify causes of the problem based on the given situation;
39. SPECIFIC: These outcomes are detailed specifically for this course. Upon completion of the course students should be able to: write programs using the basic grammar and syntax of Visual Basic.

IV. MEASURABLE OBJECTIVES:

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

A. Several of these outcomes are being developed throughout the entire programming sequence. Upon completion of the course, to an advanced level, students should be able to: -Technical Competencies -Programming Skills

1. Explain and use the design process
2. Define and use functions and storage classes
3. Define and explain trends in programming standards
4. Write, compile, test and debug programs
5. Present the characteristics of object-oriented programming
6. Define and use user interfaces
7. Define and use file I/O
8. Understand Structured Query Language (SQL) programming
9. Develop ability to program client-server applications using a relational database
10. Understand database views and stored procedures

11. Develop ability to program server-side applications using a relational database
12. Create database using Database Definition Language (DDL) or database application interface

B. Database Design
1. Explain database design concepts and the role of database components
2. Create and customize forms and reports
3. Explain the use of databases and information in the business environment
4. Develop database business applications

C. SQL Topics
1. The SELECT statement
2. Projection
3. Joins
4. Data Definition Language
5. DML: INSERT, MODIFY, DELETE
6. Operators & Built-in Functions

D. Systems Design
1. Specify major subsystems and interfaces
2. Develop detail design specifications
3. Explain database capability of access/storage/security
4. Identify physical requirements for systems implementation
5. Understand hardware requirements of client/server resources

E. 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

F. Testing and Debugging
1. Select debugging and testing methodology, and develop comprehensive and systematic test plan
2. Design testing programs to uncover hardware compatibility problems during the development phase of the project
3. Develop testing procedures
4. Conduct tests in the most efficient way
5. Test programs, and document errors and solutions

G. User Interface Design
1. Define the requirements for the user interface
2. Develop and test prototypes
3. Construct user interfaces for flexibility and adaptability

H. 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

I. Task Management
1. Break down projects and activities into a series of tasks

V. CONTENT:

A. Introduction – Database Processing
B. Relational Database Concept
C. Structured Query Language (SQL)
1. The SELECT statement
2. Projection
3. Joins
4. Data Definition Language
5. DML
a. INSERT
b. MODIFY
c. DELETE

D. User Interface
E. Servers
F. Using Visual Basic for Database applications
G. Client-side + Server-side Scripts & Compiled Server Programs

VI. METHODS OF INSTRUCTION:

A. **Lecture** -
B. **Demonstration** -
C. **Projects** - Optional: Programming project completed in teams
D. Lab Programming Assignments
E. **Discussion** -

VII. TYPICAL ASSIGNMENTS:

A. Write a program to create a database of customers: 1. The customer record contains the following data a. Name (20 char) b. Address (50 char) c. Phone (20 char) d. Number of Employees e. Average salary of employees 2. The application should be capable of creating, modifying, and deleting all customer records. B. Write a team programming project to create a WEB application that will act as a web browser and as a database which contains users and their web sites visited: 1. The browser application behaves just as normal browsers behave but that each user is added to the list of users and the web sites visited are tracked within the database. 2. The system administrator to the Web application can view the list of web sites and disable selected sites. 3. When a user starts the application and attempts to visit a restricted site they are denied access.

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. Write SQL command to open the file “Customers.dat” from the server located on drive F.
b. What is meant by ‘deadly embrace’ in relationship to two users trying to open the same file at the same time?
c. Write the SQL command to save the record Customer to a file.
d. Identify the major concepts to database programming as described in class. Give a brief description for each.

IX. TYPICAL TEXTS:

1. Deitel, Harvey M. & Paul J, & T.R Nieto *Internet & World Wide Web: How To Program.* 2nd ed., Prentice-Hall, 2001.
2. Morrison, Mike & Joline *Database, Driven Web Sites.*, Course Technology, 2000.
3. Mukhar, Kevin, Todd Lauinger, & John Carnell *Beginning Java Databases: JDBC, SQL, J2EE, EJB, JSP, XML.*, Perfect Paperback, 2001.
4. Roman, Steven *Access Database: Design and Programming.*, O'Reilly, 1999.

X. OTHER MATERIALS REQUIRED OF STUDENTS:

A. Software 1. Compiler/Interpreter –Visual Basic & Visual InterDev (latest versions) 2. Database – Oracle or Access (latest versions) 3. For online learning – WebCT engine and textbook companion
B. Hardware 1. Hardware (minimum) – 64MB RAM, 1GB HD, 500MHz computer systems 2. Internet – T1 line with multiple port (64+ connections)