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: Spring 2019

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

This is a programming course that goes beyond mere “desktop” database management. Participants will learn how to design and manage cloud based databases and explore dynamic applications that interact with databases using compiled and interpreted client/server programming languages. Learn databases concepts, relational database principles, database design/modeling techniques and Structured Query Language (SQL).

3.00 Units Lecture 1.00 Units Lab

Prerequisite
CS 1 - Computing Fundamentals I
with a minimum grade of C
or

CS 7 - Introduction to Computer Programming Concepts
with a minimum grade of C
or

CS 31 - Java Programming
with a minimum grade of C

Grading Methods:
Letter or P/NP

Discipline:
- Computer Science

	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. CS1
1. Design, create and compile C++ programs within multiple development environments and operating systems, including the use of command-line tools in Unix/Linux.
2. Interpret and apply C++ control structures for sequencing, selection and iteration.
3. Interpret and implement programmer-defined functions in C++.
4. Create and interpret expressions involving arithmetic and logical operators;
5. Interpret and apply arrays and simple programmer-defined data structures in C++.
6. Modify and expand short programs that use standard conditional and iterative control structures and functions.
7. Choose appropriate conditional and iteration constructs for a given programming task.
8. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
9. Analyze and explain the behavior of simple programs.
10. Describe, interpret and apply the mechanics of parameter passing.
11. Discuss and apply the concept of algorithms in problem-solving processes.
12. Judge the correctness and quality of algorithms, identifying necessary properties of good algorithms.
13. Describe and apply effective debugging strategies.
14. Identify properties of variables and apply different forms of variable binding, visibility, scoping, and lifetime management.
15. Explain, interpret and apply elements of syntax related variable types, including type-checking, abstraction, type incompatibility and type safety.
16. Design, implement, test, and debug programs using basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions.

B. CS7
1. Design simple algorithms to solve a variety programming problems.

2. Design and implement programs of short to medium length, using standard elements of programming languages such as variables, input/output, control structures, functions/methods and arrays.
3. Describe the software development life-cycle.
4. Describe the principles of structured and object-oriented programming and be able to describe, design, implement, and test structured and object-oriented programs using currently accepted methodology.
5. Explain what an algorithm is and its importance in computer programming.
6. Analyze and investigate program behavior to effectively alter or debug existing code.
7. Design and implement specific program steps and components to achieve desired program behavior.
8. Design and organize elements of a program using a structured representation such as pseudocode and/or flowcharts.
9. Design and implement simple graphical and command line user interfaces implementing the students algorithms.

C. CS31
1. Explain and apply basic principles of software engineering.
2. Design and implement both command line and graphical user interfaces using Java built-in classes.
3. Design and implement both built-in and complex Java data types and variables.
4. Understand and implement Java multi-dimensional arrays and vectors.
5. Design and implement multi-threaded Java applications.
6. Create java programs that implement standard data structures such as queues, linked lists, stacks, trees and hash tables.
7. Design and implement programs accessing network resources and client/server protocols.
8. Write, compile, test and debug java programs and applets using both command line and integrated development environments;
9. Design and implement event-driven programs;
10. Design and implement Java classes including inheritance within a class hierarchy.
11. Design and implement Java constructors and other methods within a class.
12. Apply file input/output within the implementation of a program using both sequential and random data access methodologies.
13. Create Java programs using JDBC and Relational Database Management Systems (RDBMS) such as MySQL, DB2 and/or Oracle.

IV. MEASURABLE OBJECTIVES:

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

A. Explain the use of databases and information systems in the business environment
B. Design multiple database systems using hierarchical, network, relational, and object-oriented data models
C. Explain database design concepts and the role of database components
D. Effectively use the database design process to implement a third normal formed database using industry standard relational database server engines (RDBMS)
E. Write, compile, test and debug programs which utilize RDBMS technologies using languages such as Python, Java and C/C++
F. Design, program and test user interfaces which utilize RDBMS technologies
G. Design and program Structured Query Language (SQL) statements including the SELECT statement, Projection, Joins, Data Definition Language, INSERT, MODIFY, DELETE and Built-in Functions
H. Design and program client-server applications using relational databases as the primary data store
I. Create/Modfy/Convert database(s) using the Database Definition Language (DDL) from multiple Database engines/technologies
J. Implement database views and stored procedures
K. Design and implement a security model for any RDBMS with focus on internet security intrusions
L. Explain and implement advanced database concepts such as data warehousing, Big-Data analytics, transaction management and concurrency control and distributed database systems
M. Select debugging and testing methodologies, and develop comprehensive test plans to be executed on previously designed and implemented projects

V. CONTENT:

A. Introduction – Database Systems and Database Design
B. Relational Database Concepts and Database Modeling
C. Implementation and Management of Relational Database Servers
D. Structured Query Language (SQL)
   1. The SELECT statement
   2. Projection
   3. Joins
   4. Data Definition Language
   5. Procedural and Embedded SQL
   6. Query Optimization
E. RDBMS Programming Concepts (Java, Python, C/C++, PHP)
F. User Interface design and implementation
G. Distributed and Cloud-based Database Management Systems
H. Business Intelligence and Data Warehousing
I. Big-Data Analytics using C++/Python/R

VI. METHODS OF INSTRUCTION:

A. **Student Presentations** - Development and Presentation of a fully defined RDBMS design for client/server applications
B. **Demonstration** -
C. **Lecture** -
D. **Classroom Activity** - Implementation of example database systems and designs as individuals or in-class teams
E. Lab Programming Assignments
F. **Projects** - Design and Programming projects completed individually and in teams
G. **Discussion** - Overview of design ideas and implementation of team based RDBMS projects

VII. TYPICAL ASSIGNMENTS:

A. Design a cross-platform database implemented using MySQL, DB2 or Oracle database engines and present the database design to the class as a whole.
B. Design and implement a program which uses the previously approved design with either a CLI or GUI user interface, showing the full capabilities of the database in a cloud based environment.
C. Write a browser-based database application which uses a previously designed and approved database solution and compare/contrast the differences it has with cloud-based applications.

VIII. EVALUATION:

Methods/Frequency

A. Exams/Tests
   Midterm Exam Final Exam
B. Quizzes
   Quizzes for each chapter of the book and for extra material provided by the instructor. Frequency: this could be done weekly, or

when a chapter is completed.

C. Projects
   Any number of projects which will demonstrate the student's ability to perform the tasks taught during the course. Frequency: Projects will normally be required for each chapter of the text, so weekly if a chapter is finished in this amount of time.
D. Class Participation
   Do the students ask questions, work in teams, contribute to on-line discussion boards and do they assist other students in the course work materials?
E. Class Work
   During class, in-class projects will be assigned to strengthen the student's ability to apply concepts learned during lecture. Frequency: Each class
F. Home Work
   Programming and design projects will be given to students during off-hours to assure they understand concepts taught during class. Frequency: Projects (Homework) will normally be required for each chapter of the text, so weekly if a chapter is finished in this amount of time.
G. Final Class Performance
   A measurement of final class performance will be done using all tests, quizzes and assignments as well as an end-of-course assignment showing the student's ability to apply all concepts learned during class lecture and class activities. Frequency: Mid-Term and Final exams will be major measurements of final class performance, along with the projects assigned and class participation.

IX. TYPICAL TEXTS:

1. Thomson, Welling. *PHP and MySQL Web Development.* 5th ed., Addison-Wesley Professional, 2017.
2. Navathe, Shamkant. *Fundamentals of Database Systems.* 7th ed., Pearson, 2016.
3. Coronel, Carlos, and Steven Morris . *Database Systems, Design, Implementation and Management.* 12 ed., Cengage, 2017.
4. MySQL Client/Server. Oracle, (5.x).
5. DB2 Developers Edition. IBM, (11.1).

X. OTHER MATERIALS REQUIRED OF STUDENTS:

A. USB thumb drive, memory stick, disk drive or cloud based storage device for saving class work
B. Optional: A personal computer (Windows or Macintosh) to be able to work from home