SAN DIEGO COMMUNITY COLLEGE DISTRICT
CONTINUING EDUCATION
COURSE OUTLINE

SECTION I

SUBJECT AREA AND COURSE NUMBER

COMP 662

COURSE TITLE

PROGRAMMING DATABASES - PYTHON

TYPE COURSE

NON-FEE VOCATIONAL

CATALOG COURSE DESCRIPTION

This is an advanced course that introduces the learner to programming databases with Python. Topics covered include configuring drivers, creating a database, using Structured Query Language (SQL), coding in Python to retrieve data from and update a database, and use an Object Relational Mapping (ORM) language to simplify coding. Examples used are drawn from diverse areas such as financial data processing, gaming applications, and more. Students will be able to use this knowledge to land mid-level positions in such fields as data science, embedded programming, data analytics, and more. (FT)

LECTURE/LABORATORY HOURS

126

ADVISORIES

COMP 660 PROGRAMMING WITH PYTHON I; and
COMP 661 PROGRAMMING WITH PYTHON II

RECOMMENDED SKILL LEVEL

- Possess a 12th grade reading level
- Ability to communicate effectively in the English language
- Knowledge of math concepts at the 8th grade level and computer literacy

INSTITUTIONAL STUDENT LEARNING OUTCOMES

1. Social Responsibility
   SDCE students demonstrate interpersonal skills by leaning and working cooperatively in a diverse environment.
2. Effective Communication
   SDCE students demonstrate effective communication skills.

INSTITUTIONAL STUDENT LEARNING OUTCOMES (CONTINUED)

3. Critical Thinking
   SDCE students critically process information, make decisions, and solve problems independently or cooperatively.
4. Personal and Professional Development
   SDCE students pursue short term and life-long learning goals, mastering necessary skills and using resource management and self-advocacy skills to cope with changing situations in their lives.

COURSE GOALS

1. Gain advanced skills for working with Python and relational databases.
2. Learn how to install the necessary drivers for connecting Python to a relational database.
3. Use Python modules to connect to relational databases.
4. Learn how to create and model a database using a database management system.
5. Work with the Structured Query Language (SQL) to retrieve and add data within a database management system.
6. Use Python code to create, retrieve, update, and delete (CRUD) data within a relational database.
7. Use an Object Relational Mapping (ORM) language to in conjunction with Python to simplify Python coding with databases.

COURSE OBJECTIVES

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

1. Demonstrate an advanced understanding of Python and relational databases.
2. Install the necessary tools and drivers required for connecting Python to relational databases.
3. Use a Python module to connect to a relational database.
4. Architect a database using a database management system.
5. Demonstrate a basic understanding of using the Structured Query Language (SQL) to retrieve and add data within a database management system.
6. Demonstrate the ability to use Python code to create, retrieve, update, and delete (CRUD) data in a relational database.
7. Demonstrate a basic understanding of how to use an Object Relational Mapping (ORM) language in conjunction with Python code.

SECTION II

COURSE CONTENT AND SCOPE

1. Introduction to Python, Databases, and Tools
    - 1.1. Importance of data
    - 1.2. Introduction to Python
    - 1.3. Uses of Python
    - 1.4. Introduction to databases
    - 1.5. Database and Relational Database Management System (RDBMS) options
        - 1.5.1. SQLite and SQLite Studio
        - 1.5.2. MySQL and PhpMyAdmin
        - 1.5.3. SQL Server and Management Studio
    - 1.6. Object Database Connectivity (ODBC), Database Application Programming Interface (DB API), and pyodbc
    - 1.7. Installing Tools for the Course
        - 1.7.1. Installing Python
        - 1.7.2. Installing Eclipse
        - 1.7.3. Installing PyDev
        - 1.7.4. Installing the pyodbc Module
        - 1.7.5. Installing the MySQL Driver
        - 1.7.6. Installing the Microsoft ODBC Driver
2. Creating Databases and Tables
    - 2.1. Introduction to relational data
    - 2.2. Planning and designing a database
        - 2.2.1. Tables, records, and columns
        - 2.2.2. Assigning data types
        - 2.2.3. Normalization
        - 2.2.4. Keys and relationships
    - 2.3. Review a sample database structure
    - 2.4. Creating a database
    - 2.5. Establishing the relationships
    - 2.6. Manually adding data to the tables
3. Introduction to the Structured Query Language (SQL)
    - 3.1. SELECT command
    - 3.2. Filtering and ordering data with clauses
    - 3.3. Operators for advanced querying
    - 3.4. INSERT INTO Command
    - 3.5. UPDATE Command
    - 3.6. DELETE Command
    - 3.7. SQL expressions and aliases
    - 3.8. Common SQL functions
    - 3.9. Joining related tables
4. Python and Relational Databases
    - 4.1. Referencing the required modules
    - 4.2. Connecting to a different types of databases
    - 4.3. How to select data and return a result set
    - 4.4. Inserting new data into the database
    - 4.5. Updating data within the database

COURSE CONTENT AND SCOPE

- 4.6. Deleting data from the database
- 4.7. How to test database code
- 4.8. How to handle database exceptions

5. Using Object Relational Mapping (ORM) Tools with Python
    - 5.1. Introduction to different types of ORM tools
    - 5.2. Connecting to a database with an ORM tool
    - 5.3. Declare a mapping
    - 5.4. Create a schema
    - 5.5. Create an instance of the mapped class
    - 5.6. Creating a session
    - 5.7. Adding and updating objects
    - 5.8. Rolling back
    - 5.9. Querying
        - 5.9.1. Common filter operators
        - 5.9.2. Returning lists and scalars
        - 5.9.3. Using Textual SQL
        - 5.9.4. Counting
    - 5.10. Building a relationship
    - 5.11. Working with related objects
    - 5.12. Querying with joins
        - 5.12.1. Using aliases
        - 5.12.2. Using subqueries
        - 5.12.3. Selecting entities from subqueries
        - 5.12.4. Using EXISTS
        - 5.12.5. Common relationship operators
    - 5.13. Eager Loading
        - 5.13.1. Subquery load
        - 5.13.2. Joined load
        - 5.13.3. Explicit join and eagerload
    - 5.14. Deleting
        - 5.14.1. Configuring delete/delete-orphan Cascade
    - 5.15. Building a Many-To-Many relationship

APPROPRIATE READINGS

Reading assignments may include, but are not limited to assigned readings from textbooks, supplemental reading assignments, industry-related periodicals or magazines, manuals, online help pages, articles posted on the Internet, and information from Web sites, online libraries and databases. Topics should be related to Python programming with databases and include techniques for connecting to a database using Python, and performing create, update, retrieve, and delete operations.

WRITING ASSIGNMENTS

Writing assignments may include, but are not limited to, completing assigned reports, providing written answers to assigned questions, performing internet research and reporting on that research. An example would include a case study about how an organization uses Python and a Python-connected relational database for inventory management, order fulfillment, human resources, and client support.

OUTSIDE ASSIGNMENTS

Outside assignments may include, but are not limited to, appropriate internet research, reading from assigned textbooks and completing the assignments at the end of each chapter, and studying as needed to perform successfully in class. An appropriate assignment for instance, would include the creation of an application that stores support requests made from clients for an organization’s customer support department.

APPROPRIATE ASSIGNMENTS THAT DEMONSTRATE CRITICAL THINKING

Assignments, which demonstrate critical thinking, may include but are not limited to designing and building an application that uses Python to connect to a relational database. Students will also be expected to participate in online class discussion posts, in-class discussions and project reviews.

EVALUATION

Evaluation that a student has met the course competencies will include multiple measures of performance related to the course objectives. Evaluation methods may include, but are not limited to performance in a variety of activities and assignments, such as completing a research project individually or in a group, hands-on projects, and demonstration of use of the internet, quizzes, class participation, written and practical tests, attendance and punctuality.

Upon successful completion of the course, a Certificate of Course Completion will be issued. Upon successful completion of all courses in the program, a Certificate of Program Completion will be issued.

METHOD OF INSTRUCTION

Methods of instruction, may include, but are not limited to, lecture, in-class and online discussions, hands-on demonstrations, computer-assisted instruction, field trips, and laboratory assignments.

This course, or sections of this course, may be offered through distance education.

TEXTS AND SUPPLIES

Essential SQLAlchemy: Mapping Python to Databases, Jason Myers and Rick Copeland, O'Reilly Press, current edition,
MySQL Connector/Python Revealed, Krogh and Jesper Wisborg, Apress, current edition,

Web Resources:
Udemy: Programming with Python, https://www.udemy.com/programming-with-python/learn/v4/overview
Python Org, https://www.python.org/
Wikibooks: Python Programming, https://en.wikibooks.org/wiki/Python_Programming

Supplies: Journal (composition book), USB Drive or other storage media

PREPARED BY Zak Ruvalcaba DATE June 5, 2019

REVISED BY ____________________ DATE ____________________

Instructors must meet all requirements stated in Policy 3100 (Student Rights, Responsibilities and Administrative Due Process), and the Attendance Policy set forth in the Continuing Education Catalog.

REFERENCES:

San Diego Community College District Policy 3100
California Community Colleges, Title 5, Section 55002
Continuing Education Catalog