

Florida Department of Education
CURRICULUM FRAMEWORK

Program Title: Academy of Database and Programming Essentials
Program Type: Job Preparatory
Occupational Area: Business Technology Education
Components: 1 Occupational Completion Point

Program Number: Secondary
 8206400
CIP Number: 0507.039904
Grade Level: 11-12
Standard Length: 2 credits
Certification: Bus Ed or COMP SCI @6 @2 (core and prerequisite)
 ANY FIELD with appropriate two industry certification

CTSO: FBLA
 BPA

Coop Method: Yes
Apprenticeship: No
Facility Code: 212

- I. **PURPOSE:** This program is designed to train students for entry-level database and internet/web related positions. It is a locally-based education initiative that partners business and industry, education, and government to teach students database and Internet technologies skills using the latest industry tools. In addition to learning "value-added" technologies, students will discover a variety of higher education and career options that will prepare them to be more competitive in the 21st century skills for the marketplace.

Students should have completed Algebra 1, Computing for College and Careers or Keyboarding and Business Skills *plus* Computer and Business Skills or equivalent coursework prior to entering **Database Fundamentals**. A programming or flow-chart class is also recommended.

- II. **PROGRAM STRUCTURE:** This program is a two-year, four-course, 2-credit program with work-based learning experiences required. Curriculum is project-based. There is a certification exam given at completion of the first two courses and another at completion of the final two courses. Sequence is as follows:

Recommended prerequisite: Business Computer Programming 1

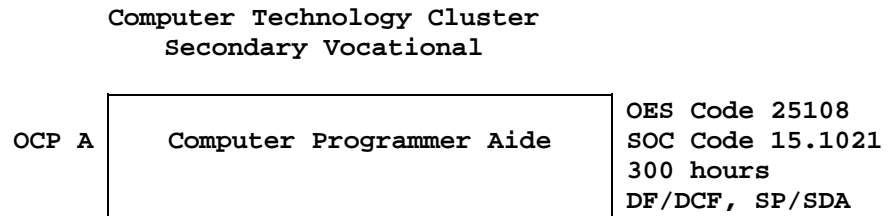
Core: Any career education core course accepted for Gold Seal requirements.

First year: Course 1 - 8206410 Database Fundamentals
 Course 2 - 8206420 Data Control and Functions
 Certification Exam
 [Possible internship in SQL & PL/SQL near end of second course]

Second year: Course 3 - 8206430 Specialized Programming
 Course 4 - 8206440 Specialized Database
 Applications
 Programming Certification Exam
 Work-based Learning Experiences

A student completing the entire program will earn OCP A:
 SOC Code - 15-1021 Computer Programmer Aide

The following diagram illustrates the Academy of Database and
 Programming Essentials program structure:



III. **LABORATORY ACTIVITIES:** Laboratory activities are an integral part of this program and include the use of computers, and peripheral equipment.

IV. **SPECIAL NOTES:** Future Business Leaders of America and Business Professionals of America are the appropriate Career and Technical Student Organizations (CTSO) for providing leadership training and for reinforcing specific career and technical skills. Career and Technical Student Organizations, when provided, shall be an integral part of the career and technical instructional program, and the activities of such organizations are defined as part of the curriculum in accordance with Rule 6A-6.065, FAC.

Work-based learning experiences are required. The following is required for each student employed in the Academy internship: an internship agreement among teacher/school, students, parents, and employers; a job performance skills plan signed by the student, ADPE teacher, and employer listing the competencies developed through classroom experiences and the on-the-job duties and tasks to be performed during the internship; documentation signed by the student and employer that the student worked the minimum hours required by the Academy; and an evaluative criteria of the student's on-the-job performance.

The ADPE teacher must visit each intern site a minimum of once during the term of the internship for the purpose of evaluating the student's progress in attaining the competencies listed in the job performance skills plan and in determining the student's grade for the internship course.

Federal and state legislation requires the provision of accommodations for students with disabilities to meet individual needs and ensure equal access. Adult students with disabilities must self-identify and request such services. Students with

disabilities may need accommodations in such areas as instructional methods and materials, assignments and assessments, time demands and schedules learning environment, assistive technology and special communication systems. Documentation of the accommodations requested and provided should be maintained in a confidential file.

SCANS Competencies: Instructional strategies for this program must include methods that require students to identify, organize, and use resources appropriately; to work with each other cooperatively and productively, to acquire and use information; to understand social, organizational, and technological systems; and to work with a variety of tools and equipment. Instructional strategies must also incorporate the methods to improve students' personal qualities and high-order thinking skills.

21st-Century Skills: Instructional strategies for this program must include methods that require students to acquire 1. Digital-Age Literacy - basic scientific, mathematical, and technological literacies - visual and information literacies - cultural literacy and global awareness; 2. Inventive Thinking - adaptability/ability to manage complexity - curiosity, creativity, and risk taking - higher order thinking and sound reasoning; 3. Effective Communication - teaming, collaboration, and interpersonal skills - personal and social responsibility - interactive communication; 4. High Productivity - ability to prioritize, plan, and manage for results - effective use of real-world tools - relevant, high-quality products.

Equipment List: A generic equipment list is available for this program.

- V. INTENDED OUTCOMES: After completing the following competencies, the student will be able to:

**OCCUPATIONAL COMPLETION POINT - DATA CODE A
COMPUTER PROGRAMMER AIDE - SOC Code 15-1021**

After completing the following competencies, the student will have achieved the occupational completion point of Computer Programmer Aide and the student will be able to:

DATABASE FUNDAMENTALS

- 269.0 Develop an awareness of the changes taking place in the Information Age and how they fit into an evolving society. [Student Performance Standards: 269.01, 269.02, 269.03, 269.04, 269.05]
- 270.0 Develop the "big picture" of database design and how to best organize data according to business rules and/or client needs. [Student Performance Standards: 270.01, 270.02, 270.03]
- 271.0 Develop the process of creating an entity by identifying relationships. [Student Performance Standards: 271.01, 271.02, 271.03, 271.04]
- 272.0 Formulate and assemble initial Entity Relationship by expanding on modeling concepts. [Student Performance Standards: 272.01, 272.02, 272.03]

- 210.0 Consider the degree and optionality of relationships of entities. [Student Performance Standards: 210.01, 210.02, 210.03, 210.04, 210.05]
- 211.0 Demonstrate proficiency in early construction stages of the data modeling process by using unique identifiers and Many-to-Many (M:M) relationships for building Entity Relationship Diagrams. [Student Performance Standards: 211.01, 211.02, 211.03, 211.04]
- 212.0 Demonstrate proficiency in advanced data constructs by analyzing business requirements and diagramming entities and relationships. [Student Performance Standards: 212.01, 212.02, 212.03, 212.04, 212.05]
- 213.0 Incorporate rules for determining item storage, retrieval and relationship employing standarding methods. [Student Performance Standards: 213.01, 213.02]
- 214.0 Demonstrate proficiency in data storage and dangers in designing the process for storage by adding complexity to an Entity-Relationship Model (ERM). [Student Performance Standards: 214.01, 214.02, 214.03, 214.04, 214.05, 214.06]
- 215.0 Apply the complex ERM information by fine tuning entities and the process for relating them. [Student Performance Standards: 215.01, 215.02, 215.03, 215.04, 215.05, 215.06]
- 216.0 Apply initial database design and normalization by following the set of house rules that determine how items are stored and retrieved. [Student Performance Standards: 216.01, 216.02, 216.03]
- 217.0 Demonstrate proficiency in the technique of normalization by labeling and organizing all items in a database in such a way as to prevent any confusion of mistakes. [Student Performance Standards: 217.01, 217.02, 217.03, 217.04, 217.05]
- 218.0 Demonstrate proficiency in table normalization by combining the techniques of an Entity Relationship Model or a top-down, business approach to data with Normalization or a bottom-up mathematical approach to data. [Student Performance Standards: 218.01, 218.02, 218.03, 218.04, 218.05]
- 219.0 Apply blueprint principles to begin designing a tool for creating a web-based interface access to a database. [Student Performance Standards: 219.01, 219.02, 219.03]
- 220.0 Extend the ERM presentation model by normalizing the data and mapping the management system. [Student Performance Standards: 220.01, 220.02, 220.03, 220.04, 220.05]
- 221.0 Apply techniques for building a storage management system by creating a website using templates and wizards. [Student Performance Standards: 221.01, 221.02, 221.03]
- 222.0 Demonstrate storage closet design and functionality by constructing a group business presentation. [Student Performance Standards: 222.01, 222.02]
- 223.0 Demonstrate comprehension of database modeling competency through group presentation. [Student Performance Standards: 223.01, 223.02, 223.03]

DATA CONTROL AND FUNCTIONS

- 224.0 Demonstrate comprehension that the database management software is a system for organizing the storage unit (or database) according to business needs and rules, through data integrity constraints [Student Performance Standards: 224.01, 224.02, 224.03, 224.04, 224.05]
- 225.0 Demonstrate comprehension of aspects of SQL Language interface by writing basic SQL statements. [Student Performance Standards: 225.01, 225.02, 225.03]
- 227.0 Demonstrate proficiency using SQL single row functions. [Student Performance Standards: 227.01, 227.02, 227.03, 227.04, 227.05]
- 228.0 Demonstrate proficiency displaying data from multiple tables. [Student Performance Standards: 228.01, 228.02, 228.03]
- 229.0 Demonstrate proficiency aggregating data using group functions. [Student Performance Standards: 229.01, 229.02, 229.03]
- 230.0 Demonstrate proficiency utilizing subqueries. [Student Performance Standards: 230.01, 230.02, 230.03, 230.04]
- 231.0 Demonstrate proficiency producing readable output with SQL language interface and reporting tool and manipulating data. [Student Performance Standards: 231.01, 231.02, 231.03, 231.04, 231.05, 231.06]
- 232.0 Demonstrate proficiency creating and managing database objects. [Student Performance Standards: 232.01, 232.02, 232.03]
- 233.0 Demonstrate proficiency altering tables and constraints implementing views. [Student Performance Standards: 233.01, 233.02, 231.03, 233.04]
- 234.0 Demonstrate mastery of creating and implementing views, synonyms, indexes and other database objects. [Student Performance Standards: 234.01, 234.02, 234.03, 234.04, 234.05, 234.06, 234.07, 234.08]
- 235.0 Demonstrate ability to control user access and SQL language interface and reporting tool. [Student Performance Standards: 235.01, 235.02, 235.03]
- 236.0 Demonstrate comprehension of bundling features of SQL. [Student Performance Standards: 236.01, 236.02, 236.03, 236.04]
- 237.0 Demonstrate comprehension working with composite data types by writing executable script files. [Student Performance Standards: 237.01, 237.02, 237.03, 237.04, 237.05]
- 240.0 Demonstrate proficiency handling exceptions. [Student Performance Standards: 240.01, 240.02, 240.03, 240.04, 240.05, 240.06]

SPECIALIZED PROGRAMMING

- 241.0 Demonstrate comprehension of the benefits of object-oriented technology by creating applications that use Unified Modeling Language to interact with a database. [Student Performance Standards: 241.01, 241.02, 241.03, 241.04]
- 242.0 Demonstrate understanding of object-oriented analysis in determining how to represent phases or activities in creating a software project. [Student Performance Standards: 242.01]

- 242.02, 242.03, 242.04, 242.05, 242.06, 242.07, 242.08, 242.09]
- 243.0 Demonstrate understanding of the ability of a Java application to be written once and run anywhere (WORA). [Student Performance Standards: 243.01. 243.02, 243.03, 243.04, 243.05, 243.06, 243.07, 243.08, 243.09]
- 244.0 Demonstrate an understanding of background object oriented programming working with classes and objects. [Student Performance Standards: 244.01. 244.02, 244.03, 244.04, 244.05]
- 245.0 Demonstrate comprehension of precise syntax for declaring the building blocks and controlling them by using eight primitive data types and operators to manipulate these types. [Student Performance Standards: 245.01. 245.02, 245.03, 245.04, 245.05, 245.06]
- 246.0 Demonstrate proficiency in programming in a command line environment using an Integrated Design Environment or IDE. [Student Performance Standards: 246.01. 246.02, 246.03, 246.04, 246.05, 246.06]
- 247.0 Demonstrate proficiency in controlling a program's flow by using a source code debugger that allows setting of breakpoints, evaluation of expressions, single-step through an application, etc., in working with decision-making constructs and loop statements. [Student Performance Standards: 247.01. 247.02, 247.03, 247.04, 247.05, 247.06]
- 248.0 Demonstrate proficiency controlling a program using classes and objects that can interact with each other. [Student Performance Standards: 248.01. 248.02, 248.03, 248.04, 248.05, 248.06, 248.07, 248.08, 248.09]
- 249.0 Demonstrate proficiency in modifying or extending classes and objects created by someone else using the java class library and javadoc tool. [Student Performance Standards: 249.01. 249.02, 249.03, 249.04, 249.05, 249.06, 249.07, 249.08]
- 250.0 Demonstrate further proficiency for control in Java Programming using the tools of Instance Methods, Overloading and Constructors. [Student Performance Standards: 250.01. 250.02, 250.03, 250.04, 250.05, 250.06, 250.07]
- 251.0 Demonstrate proficiency manipulating collections of variables in arrays and character strings using wrapper classes. [Student Performance Standards: 251.01. 251.02, 251.03, 251.04, 251.05, 251.06, 251.07, 251.08]
- 252.0 Demonstrate comprehension of class definition in terms of how it differs from an existing class (the superclass) using inheritance. [Student Performance Standards: 252.01. 252.02, 252.03, 252.04, 252.05]
- 253.0 Demonstrate comprehension of how abstract classes and interfaces are used to extend classes and establish protocol between classes. [Student Performance Standards: 253.01. 253.02, 253.03]
- 254.0 Demonstrate proficiency in basic concepts of exception handling. [Student Performance Standards: 254.01. 254.02, 254.03, 254.04]
- 255.0 Demonstrate proficiency in creating an application that is portable across different operating systems using file input and output. [Student Performance Standards: 255.01. 255.02, 255.03, 255.04, 255.05]

SPECIALIZED DATABASE APPLICATIONS

- 258.0 Demonstrate proficiency in how applets work with specific examples and hands-on demonstrations. [Student Performance Standards: 258.01, 258.02, 258.03, 258.04]
- 259.0 Demonstrate proficiency building and deploying simple Java applets. [Student Performance Standards: 259.01, 259.02, 259.03, 259.04]
- 261.0 Demonstrate proficiency using the syntax and semantics of multithreading in Java. [Student Performance Standards: 261.01, 261.02, 261.03, 261.04, 261.05, 261.06, 261.07, 261.08]
- 262.0 Demonstrate proficiency in preparation for Java final exam. 262.01, 262.02, 262.03, 262.04, 262.05, 262.06, 262.07, 262.08, 262.19, 262.10, 262.11, 262.12, 262.13, 262.14, 262.15, 262.16, 262.17, 262.18, 262.19, 262.20, 262.21, 262.22, 262.23, 262.24, 262.25, 262.26, 262.27, 262.28, 262.29, 262.30, 262.31, 262.32]
- 265.0 Create an interface that demonstrates most of the concepts learned throughout the program courses. [Student Performance Standards: 265.01, 265.02, 265.03, 265.04, 265.05, 265.07, 265.08, 265.09, 265.10, 265.11]

Florida Department of Education
STUDENT PERFORMANCE STANDARDS

Course Number: 8206410
Course Title: Database Fundamentals
Course Credit: .5

COURSE DESCRIPTION:

This data modeling course is designed to provide the foundation for future software engineers or database administrators. It transforms business requirements into an operational database utilizing a top-down systematic approach. Content includes creation of entity-relationship diagrams that accurately model an organization's needs and support the functions of a business, mapping of information requirements into a relational database design, creation of physical relational database tables to implement database design, construction of a website that interacts with a database and generates report using WebDB, and organization and composition of formal presentations, integrating multimedia software.

269.0 DEVELOP AN AWARENESS OF THE CHANGES TAKING PLACE IN THE INFORMATION AGE AND HOW THEY FIT INTO AN EVOLVING SOCIETY-The student will be able to:

- 269.01 Cite examples of jobs, salary, and opportunities he/she will have as a result of participating in the Academy.
- 269.02 Describe the role a database plays in a business and predict its evolution.
- 269.03 Demonstrate the difference between "data" and "information."
- 269.04 Understand the importance of clear communication when discussing business informational requirements.
- 269.05 Experiment with web-based email and explain how these services use a database.

270.0 DEVELOP THE "BIG PICTURE" OF DATABASE DESIGN AND HOW TO BEST ORGANIZE DATA ACCORDING TO BUSINESS RULES AND/OR CLIENT NEEDS-The student will be able to:

- 270.01 Identify and analyze the phases of the database development process.
- 270.02 Explain what conceptual data modeling and database design involve.
- 270.03 Compare database development Process with that of the application development process.

271.0 DEVELOP THE PROCESS OF CREATING AN ENTITY BY IDENTIFYING RELATIONSHIPS-The student will be able to:

- 271.01 Identify and model various types of entities.
- 271.02 Identify naming and drawing conventions for entities.
- 271.03 Sequence the steps that are necessary for creation of an entity.

- 271.04 Analyze and model the relationships between entities.
- 272.0 FORMULATE AND ASSEMBLE INITIAL ENTITY RELATIONSHIP BY EXPANDING ON MODELING CONCEPTS-The student will be able to:
- 272.01 Analyze and model attributes.
 272.02 Identify unique identifiers for each entity.
 272.03 Develop an entity relationship diagram tagging attributes with optionality.
- 210.0 CONSIDER THE DEGREE AND OPTIONALITY OF RELATIONSHIPS OF ENTITIES-The student will be able to:
- 210.01 Create models and entity relationship information requirements and interviews.
 210.02 Begin to differentiate between one-to-many, many-to-many and one-to-one relationships.
 210.03 Identify relationship between two entities by reading a given diagram.
 210.04 Create a relationship between instances of the same entity.
 210.05 Read an entity relationship model in order to validate it.
- 211.0 DEMONSTRATE PROFICIENCY IN EARLY CONSTRUCTION STAGES OF THE DATA MODELING PROCESS BY USING UNIQUE IDENTIFIERS AND MANY-TO-MANY (M:M) RELATIONSHIPS FOR BUILDING ENTITY RELATIONSHIP DIAGRAMS-The student will be able to:
- 211.01 Identify the significance of an attribute that has more than one value for each entity instance.
 211.02 Evaluate appropriate methods of storing validation rules for attributes.
 211.03 Recognize unique identifiers inherited from other entities.
 211.04 Sequence the steps involved in resolving a many-to-many relationship.
- 212.0 DEMONSTRATE PROFICIENCY IN ADVANCED DATA CONSTRUCTS BY ANALYZING BUSINESS REQUIREMENTS AND DIAGRAMMING ENTITIES AND RELATIONSHIPS-The student will be able to:
- 212.01 Validate that an attribute is properly placed based upon its dependence on its entity's unique identifier (UID).
 212.02 Resolve many-to-many relationships with intersection entities.
 212.03 Model advanced data constructs including recursive relationships, subtypes, and exclusive relationships.
 212.04 Create exclusive entities and relationships by using subtypes and arcs, respectively.
 212.05 Identify initial layout for presentation and generate a list of action items for members of group.
- 213.0 INCORPORATE RULES FOR DETERMINING ITEM STORAGE, RETRIEVAL AND RELATIONSHIP EMPLOYING STANDARD METHODS-The student will be able to:
- 213.01 Model advanced data constructs including recursive relationships, subtypes, and exclusive relationships.

- 213.02 Develop an entity relationship model using subtypes, supertypes and an exclusive arc.
- 214.0 DEMONSTRATE PROFICIENCY IN DATA STORAGE AND DANGERS IN DESIGNING THE PROCESS FOR STORAGE BY ADDING COMPLEXITY TO AN ENTITY-RELATIONSHIP MODEL (ERM)-The student will be able to:
- 214.01 Revise an entity relationship model according to the diagramming techniques covered in this course.
 - 214.02 Formulate reasons for holding historical data.
 - 214.03 Recognize and resolve fan traps and chasm traps.
 - 214.04 Differentiate between transferable and non-transferable relationships.
 - 214.05 Deliver a professional, formal business style presentation.
 - 214.06 Evaluate and critique presentation layout, design and performance.
- 215.0 APPLY THE COMPLEX ERM INFORMATION BY FINE-TUNING ENTITIES AND THE PROCESS FOR RELATING THEM-The student will be able to:
- 215.01 Describe a relational database and how it is different from other database systems.
 - 215.02 Define primary keys and foreign keys and describe their purpose.
 - 215.03 Describe what data integrity refers to and list some constraints.
 - 215.04 Explain how database design fits into the database development process.
 - 215.05 Translate an entity-relationship model into a relational database design.
 - 215.06 Document a database design using table instance charts.
- 216.0 APPLY INITIAL DATABASE DESIGN AND NORMALIZATION BY FOLLOWING THE SET OF HOUSE RULES THAT DETERMINE HOW ITEMS ARE STORED AND RETRIEVED-The student will be able to:
- 216.01 Demonstrate ability to implement six steps for mapping entity relationship models.
 - 216.02 Document an initial database design on table instance charts.
 - 216.03 Recognize raw data and evaluate the steps for creating a data group in unnormalized form (UNF).
- 217.0 DEMONSTRATE PROFICIENCY IN THE TECHNIQUE OF NORMALIZATION BY LABELING AND ORGANIZING ALL ITEMS IN A DATABASE IN SUCH A WAY AS TO PREVENT ANY CONFUSION OR MISTAKES-The student will be able to:
- 217.01 Differentiate between unnormalized data and normalized.
 - 217.02 Move data from an unnormalized form through to a third normal form.
 - 217.03 Demonstrate ability to test data groups for third normal form compliance.
 - 217.04 Identify optimized data groups from given groups of normalized data.
 - 217.05 Illustrate a personal folder using WebDB and altering the navigation bar, banner and content area.

- 218.0 DEMONSTRATE PROFICIENCY IN TABLE NORMALIZATION BY COMBINING THE TECHNIQUES OF AN ENTITY RELATIONSHIP MODEL OR A TOP-DOWN, BUSINESS APPROACH TO DATA WITH NORMALIZATION OR A BOTTOM-UP MATHEMATICAL APPROACH TO DATA-The student will be able to:
- 218.01 Compare the normalization and entity relationship modeling (ERM) techniques in terms of strengths and weaknesses.
 - 218.02 Further define normalization and explain its benefits.
 - 218.03 Place tables in third normal form.
 - 218.04 Explain how conceptual data modeling rules ensure normalized tables.
 - 218.05 Specify referential integrity constraints and design indices.
- 218.1 APPLY BLUEPRINT PRINCIPLES TO BEGIN DESIGNING A TOOL FOR CREATING A WEB-BASED INTERFACE ACCESS TO A DATABASE-The student will be able to:
- 219.01 Evaluate the transformation of business requirements into an initial layout and design for a database.
 - 219.02 Construct simple web page design for personal work folder.
 - 219.03 Evaluate existing web sites and determine quality of design.
- 220.0 EXTEND THE ERM PRESENTATION MODEL BY NORMALIZING THE DATA AND MAPPING THE MANAGEMENT SYSTEM-The student will be able to:
- 220.01 Formulate a plan of action for the Database Project using skills previously learned in this course.
 - 220.02 Normalize an ERM to the third normal form (TNF).
 - 220.03 Create a table in the database using WebDB
 - 220.04 Demonstrate ability to edit tables using WebDB.
 - 220.05 Create forms that will display the table components created with WebDB.
- 221.0 APPLY TECHNIQUES FOR BUILDING A STORAGE MANAGEMENT SYSTEM BY CREATING A WEBSITE USING TEMPLATES AND WIZARDS-The student will be able to:
- 221.01 Create a web site that displays the database project home using WebDB.
 - 221.02 Link a web site to create a web-enabled interface to the industry database.
 - 221.03 Edit the forms created with WebDB and specify appropriate field labels for data entry.
- 222.0 DEMONSTRATE STORAGE CLOSET DESIGN AND FUNCTIONALITY BY CONSTRUCTING A GROUP BUSINESS PRESENTATION-The student will be able to:
- 222.01 Evaluate and generate criteria for a formal, business presentation.
 - 222.02 Construct a persuasive group presentation using the guidelines set forth in class.

223.0 DEMONSTRATE COMPREHENSION OF DATABASE MODELING COMPETENCY THROUGH GROUP PRESENTATION-The student will be able to:

- 223.01 Deliver a formal business presentation for the class that discusses an entity-relationship model and initial database design.
- 223.02 Demonstrate the functionality of the database and the layout/design capabilities of WebDB.
- 223.03 Self-assess learning experience through the presentation and demonstration of their final database project.

**Florida Department of Education
STUDENT PERFORMANCE STANDARDS**

Course Number: 8206420
Course Title: Data Control and Functions
Course Credit: .5

COURSE DESCRIPTION:

This course introduces data-server technology. Structured Query Language (SQL) is the standardized language that creates a medium for companies to compete in the building of databases or data management systems. Content of this course includes creation and maintenance of database objects and storage, retrieval and manipulation of data using SQL and Programming Language (PL) PL/SQL programming languages. At completion of Database Fundamentals and Data Control and Functions, students will be able to create blocks of application code that can be shared by multiple forms, reports and data management applications and to sit for the SQL and PL/SQL certification exam.

224.0 DEMONSTRATE COMPREHENSION THAT THE DATABASE MANAGEMENT SOFTWARE IS A SYSTEM FOR ORGANIZING THE STORAGE UNIT (OR DATABASE) ACCORDING TO BUSINESS NEEDS AND RULES, THROUGH DATA INTEGRITY CONSTRAINTS-The student will be able to:

- 224.01 Identify the structural elements of a relational database table.
- 224.02 List and describe the system development life cycle.
- 224.03 Describe the industry implementation of the relational database management system (RDBMS) and object relational database management system (ORDBMS).
- 224.04 Explain how SQL and languages that extend SQL are used in the industry product set.
- 224.05 Identify the advantages of a database management system.

225.0 DEMONSTRATE COMPREHENSION OF ASPECTS OF SQL language interface and reporting tool BY WRITING BASIC SQL STATEMENTS-The student will be able to:

- 225.01 List the capabilities of SQL SELECT statements.
- 225.02 Execute a basic select statement.
- 225.03 Differentiate between SQL statements and language commands that extend SQL.

227.0 DEMONSTRATE PROFICIENCY USING SQL SINGLE ROW FUNCTIONS-The student will be able to:

- 227.01 Perform calculations on data.
- 227.02 Modify individual data items.
- 227.03 Use character, number and date functions in SELECT statements.
- 227.04 Format data and numbers for display purposes.
- 227.05 Convert column data types.

- 228.0 DEMONSTRATE PROFICIENCY DISPLAYING DATA FROM MULTIPLE TABLES-The student will be able to:
- 228.01 Construct select statements to access data from more than one table using equality and nonequality joins.
 - 228.02 Use outer joins through viewing data that generally does not meet a join condition.
 - 228.03 Join a table to itself.
- 229.0 DEMONSTRATE PROFICIENCY AGGREGATING DATA USING GROUP FUNCTIONS-The student will be able to:
- 229.01 Identify the available group functions and describe their use.
 - 229.02 Demonstrate the ability to group data through the use of the GROUP BY clause.
 - 229.03 Demonstrate the ability to include or exclude grouped rows by using the having clause.
- 230.0 DEMONSTRATE PROFICIENCY UTILIZING SUBQUERIES-The student will be able to:
- 230.01 Write a query with an embedded subquery.
 - 230.02 Evaluate and perform a multiple-column subquery.
 - 230.03 Describe and explain the behavior of subqueries when null values are retrieved.
 - 230.04 Create a subquery in a from clause.
- 231.0 DEMONSTRATE PROFICIENCY PRODUCING READABLE OUTPUT WITH SQL language interface AND MANIPULATING DATA-The student will be able to:
- 231.01 Produce queries that require an input variable.
 - 231.02 Customize the SQL language interface and reporting environment using SET commands for control.
 - 231.03 Produce more readable output through the use of the column and break commands.
 - 231.04 Identify data manipulation language (DML) and describe various DML statements.
 - 231.05 Utilize data manipulation language (DML) through inserting, updating and deleting rows from a table.
 - 231.06 Control transactions using *commit* and *rollback* statements.
- 232.0 DEMONSTRATE PROFICIENCY CREATING AND MANAGING DATABASE OBJECTS-The student will be able to:
- 232.01 Describe the main database objects.
 - 232.02 Create tables and alter their definitions.
 - 232.03 Describe the datatypes that can be used when specifying column definition.
- 233.0 DEMONSTRATE PROFICIENCY ALTERING TABLES AND CONSTRAINTS IMPLEMENTING VIEWS-The student will be able to:
- 233.01 Create, drop, rename and truncate tables using SQL.
 - 233.02 Identify and describe various constraints including not null, unique, primary key, foreign key, and check.

- 233.03 Create and maintain constraints including adding, dropping, enabling, disabling, and cascading.
- 233.04 Recognize views and explain how they are created, how they retrieve data and how they perform DML operations.
- 234.0 DEMONSTRATE MASTERY OF CREATING AND IMPLEMENTING VIEWS, SYNONYMS, INDEXES AND OTHER DATABASE OBJECTS-The student will be able to:
 - 234.01 Create views, retrieve data through a view, alter the definition of a view and drop a view.
 - 234.02 Categorize information by using Top-N queries to retrieve specified data.
 - 234.03 Identify the features of a sequence and display sequence values using a data dictionary view.
 - 234.04 Identify the characteristics of a cached sequence.
 - 234.05 Modify and remove a sequence using a SQL statement.
 - 234.06 Identify the features of private and public synonyms.
 - 234.07 Identify characteristics of an index and describe different types.
 - 234.08 Create and remove an index using a SQL statement.
- 235.0 DEMONSTRATE ABILITY TO CONTROL USER ACCESS AND SQL language interface and reporting tool -The student will be able to:
 - 235.01 Identify the features of database security.
 - 235.02 Create users using SQL statements.
 - 235.03 Grant and revoke object privileges using a SQL language interface and reporting tool.
- 236.0 DEMONSTRATE COMPREHENSION OF BUNDLING FEATURES OF SQL -The student will be able to:
 - 236.01 List and describe the benefits of extensions to SQL.
 - 236.02 Recognize the basic SQL block and its sections.
 - 236.03 Declare SQL variables and describe their significance.
 - 236.04 Execute a SQL block.
- 237.0 DEMONSTRATE COMPREHENSION WORKING WITH COMPOSITE DATATYPES BY WRITING EXECUTABLE script files -The student will be able to:
 - 237.01 Recognize the significance of the executable section and decide when to use it.
 - 237.02 Write statements in the executable section.
 - 237.03 Describe the rules of nested blocks.
 - 237.04 Identify and utilize appropriate coding conventions.
 - 237.05 Create a script that will select, insert, and update data in a table.

**Florida Department of Education
STUDENT PERFORMANCE STANDARDS**

Course Number: 8206430
Course Title: Specialized Programming
Course Credit: .5

COURSE DESCRIPTION:

This is the first of two .5 credit programming courses designed to teach the basics of object-oriented programming and the Java programming language. This first course, Specialized Programming, focuses on practical object oriented principles, syntax, and command line statements.

Content includes development of Java applications using command line prompts. In addition to coding in Java, graphical capabilities of Java IDE's to aid rapid application development are covered.

241.0 DEMONSTRATE COMPREHENSION OF THE BENEFITS OF OBJECT-ORIENTED TECHNOLOGY BY CREATING APPLICATIONS THAT USE UNIFIED MODELING LANGUAGE TO INTERACT WITH A DATABASE-The student will be able to:

- 241.01 Explain the principles and terminology of object oriented development.
- 241.02 Apply a development process to move a project through analysis and design.
- 241.03 Identify simple objects and classes.
- 241.04 Describe relationships between objects.

242.0 DEMONSTRATE UNDERSTANDING OF OBJECT-ORIENTED ANALYSIS IN DETERMINING HOW TO REPRESENT PHASES OR ACTIVITIES IN CREATING A SOFTWARE PROJECT-The student will be able to:

- 242.01 Demonstrate the core concepts behind object-oriented programming: objects, methods, classes, and inheritance.
- 242.02 Compare traditional project life cycles with iterative project life cycles.
- 242.03 Describe the main activities of each phase in an object oriented project.
- 242.04 Identify the popular object oriented tools currently available.
- 242.05 List the strengths and weaknesses of object-oriented programming languages and Java.
- 242.06 Identify how the strengths and weaknesses of object-oriented programming languages—and Java translate to code.
- 242.07 Download and install a Java compiler.
- 242.08 Set the path and classpath environment variables in a Windows operating system.
- 242.09 Explore the Java API library.

243.0 DEMONSTRATE UNDERSTANDING OF THE ABILITY OF A JAVA APPLICATION TO BE WRITTEN ONCE AND RUN ANYWHERE (WORA)-The student will be able to:

- 243.01 Understand how the Java compiler, bytecodes and Virtual Machine work together.
 - 243.02 Identify the ways in which Java is suited to network environments
 - 243.03 Identify the components of the Java Software Development Kit (SDK) and their purpose.
 - 243.04 Identify the key elements of Java.
 - 243.05 Describe the Java Virtual Machine (JVM).
 - 243.06 Compare the use of Java for building applets and applications.
 - 243.07 Identify the key components of the Java Development Kit (JDK).
 - 243.08 Explain Java and a database.
 - 243.09 Create a sample applet.
- 244.0 DEMONSTRATE AN UNDERSTANDING OF BACKGROUND OBJECT ORIENTED PROGRAMMING WORKING WITH CLASSES AND OBJECTS-The student will be able to:
- 244.01 Describe aCCCraction.
 - 244.02 Create an object-oriented hierarchy using "is-a" and "has-a" relationships.
 - 244.03 Describe object-oriented concepts: inheritance, encapsulation, polymorphism.
 - 244.04 Identify simple objects and classes.
- 245.0 DEMONSTRATE COMPREHENSION OF PRECISE SYNTAX FOR DECLARING THE BUILDING BLOCKS AND CONTROLLING THEM BY USING EIGHT PRIMITIVE DATA TYPES AND OPERATORS TO MANIPULATE THESE TYPES-The student will be able to:
- 245.01 Describe primitive datatypes.
 - 245.02 Declare and initialize primitive variables.
 - 245.03 Use operators to manipulate the value in a primitive variable.
 - 245.04 Define and initialize arrays.
 - 245.05 Construct literal String values using quoted or unicode values.
 - 245.06 Identify when an object becomes eligible for garbage collection.
- 246.0 DEMONSTRATE PROFICIENCY IN PROGRAMMING IN A COMMAND LINE ENVIRONMENT USING AN INTEGRATED DESIGN ENVIRONMENT (IDE)-The student will be able to:
- 246.01 Create new workspaces, projects and applications.
 - 246.02 Build and execute Java applications.
 - 246.03 Modify Java application source code.
 - 246.04 Demonstrate familiarity and develop comfort using an integrated design environment.
 - 246.05 Re-compile sample applications from previous sections.

- 247.0 DEMONSTRATE PROFICIENCY IN CONTROLLING A PROGRAM'S FLOW BY USING A SOURCE CODE DEBUGGER THAT ALLOWS SETTING OF BREAKPOINTS, EVALUATION OF EXPRESSIONS, SINGLE-STEP THROUGH AN APPLICATION, ETC., IN WORKING WITH DECISION-MAKING CONSTRUCTS AND LOOP STATEMENTS-The student will be able to:
- 247.01 Use decision-making constructs.
 - 247.02 Perform Loop operations through writing loops and nested loops.
 - 247.03 Write nested if/else and switch statements.
 - 247.04 Use logical operators in and identify valid arguments for if and switch statements.
 - 247.05 Use break and continue with and without labels.
 - 247.06 Debug an application using an IDE debugger.
- 248.0 DEMONSTRATE PROFICIENCY CONTROLLING A PROGRAM USING CLASSES AND OBJECTS THAT CAN INTERACT WITH EACH OTHER-The student will be able to:
- 248.01 Declare classes, instances of the classes and constants.
 - 248.02 Identify the two primary components that make up the implementation of a class.
 - 248.03 Make appropriate use of the public, private, and protected keywords.
 - 248.04 Write statements that declare a class, create instance variables of the given class and declare constants to store constant values.
 - 248.05 Implement nested classes.
 - 248.06 Determine the scope of a variable in a Java program.
 - 248.07 Identify the statement for casting primitive data types and object types.
 - 248.08 Copy the contents of an object to another object.
 - 248.09 Match the math class methods with the values they return.
- 249.0 DEMONSTRATE PROFICIENCY IN MODIFYING OR EXTENDING CLASSES AND OBJECTS CREATED BY SOMEONE ELSE USING THE JAVA CLASS LIBRARY AND JAVADOC TOOL-The student will be able to:
- 249.01 Demonstrate an understanding of the organization of the Java Application Programming Interfaces (API).
 - 249.02 Identify the packages that make up the Java Software Development Kit (SDK).
 - 249.03 Use the Javadoc tool to generate API documentation from Java source files.
 - 249.04 Utilize the API documentation.
 - 249.05 Create new classes with an IDE.
 - 249.06 Use packages to group related classes.
 - 249.07 Define instance variables and methods.
 - 249.08 Create objects and call methods.
- 250.0 DEMONSTRATE FURTHER PROFICIENCY FOR CONTROL IN JAVA PROGRAMMING USING THE TOOLS OF INSTANCE METHODS, OVERLOADING AND CONSTRUCTORS-The student will be able to:
- 250.01 Declare fields and methods using the final and static modifiers.
 - 250.02 Demonstrate an understanding of how inner classes work.

- 250.03 Use final and static to create constants.
 - 250.04 Define overloaded methods in a class.
 - 250.05 Provide constructors for a class.
 - 250.06 Use class variables and class methods where appropriate.
 - 250.07 Use initializers to initialize instance and class variables.
- 251.0 DEMONSTRATE PROFICIENCY MANIPULATING COLLECTIONS OF VARIABLES IN ARRAYS AND CHARACTER STRINGS USING WRAPPER CLASSES-The student will be able to:
- 251.01 Define and initialize arrays.
 - 251.02 Construct literal String values using quoted or unicode values.
 - 251.03 Identify when an object becomes eligible for garbage collection.
 - 251.04 Apply all assignment, arithmetic, relational, bitwise, and logical operators.
 - 251.05 Understand how type conversion occurs when using operators.
 - 251.06 Create and manipulate strings in Java.
 - 251.07 Use wrapper classes to convert strings to primitives and primitives to strings.
 - 251.08 Implement resizable arrays - vectors.
- 252.0 DEMONSTRATE COMPREHENSION OF CLASS DEFINITION IN TERMS OF HOW IT DIFFERS FROM AN EXISTING CLASS (THE SUPERCLASS) USING INHERITANCE-The student will be able to:
- 252.01 Define inheritance and use inheritance to define new classes.
 - 252.02 Explain how to use polymorphism in Java classes.
 - 252.03 Identify when and how a default constructor is created for a class.
 - 252.04 Use the instance of operator when casting class objects.
 - 252.05 Override methods in the superclass.
- 253.0 DEMONSTRATE COMPREHENSION OF HOW ACCRACT CLASSES AND INTERFACES ARE USED TO EXTEND CLASSES AND ESTABLISH PROTOCOL BETWEEN CLASSES-The student will be able to:
- 253.01 Define aCCCract classes and aCCCract methods.
 - 253.02 Declare classes using the aCCCract modifier.
 - 253.03 Define and implement interfaces.
- 254.0 DEMONSTRATE PROFICIENCY IN BASIC CONCEPTS OF EXCEPTION HANDLING-The student will be able to:
- 254.01 Handle standard Java exceptions.
 - 254.02 Explain the basic concepts of exception handling.
 - 254.03 Write programs using try, catch and finally blocks.
 - 254.04 Create exception classes.

255.0 DEMONSTRATE PROFICIENCY IN CREATING AN APPLICATION THAT IS PORTABLE
ACROSS DIFFERENT OPERATING SYSTEMS USING FILE INPUT AND OUTPUT-
student will be able to:

- 255.01 Read from and write to the console.
- 255.02 Create programs that read and write text and data files.
- 255.03 Use input and output streams.
- 255.04 Write programs that operate on files and directories.
- 255.05 List the contents of a directory.

Florida Department of Education
STUDENT PERFORMANCE STANDARDS

Course Number: 8206440
Course Title: Specialized Database Applications
Course Credit: .5

COURSE DESCRIPTION:

This is the second in a two-course programming curriculum designed to teach the basics of object-oriented programming and the Java programming language.

256.0 DEMONSTRATE PROFICIENCY IN HOW APPLETS WORK WITH SPECIFIC EXAMPLES AND HANDS-ON DEMONSTRATIONS-The student will be able to:

- 258.01 Write an applet using a subclass of the applet class.
- 258.02 Draw graphics with different colors and characters with different fonts.
- 258.03 Identify the functions of the applet's execution methods.
- 258.04 Load images and sounds in an applet.

257.0 DEMONSTRATE PROFICIENCY BUILDING AND DEPLOYING SIMPLE JAVA APPLETS-The student will be able to:

- 259.01 Write an applet and embed it in an HTML page.
- 259.02 Describe which methods are called by the browser, and when.
- 259.03 Pass parameters into an applet from an HTML page.
- 259.04 Deploy an applet.

258.0 DEMONSTRATE PROFICIENCY USING THE SYNTAX AND SEMANTICS OF MULTITHREADING IN JAVA-The student will be able to:

- 261.01 Distinguish between multithreading and multiprocessing.
- 261.02 Design classes that extend the thread class.
- 261.03 Design applets that implement the runnable interface.
- 261.04 Avoid synchronization problems which can occur in multithreaded programs.
- 261.05 Explain the basic concepts of multithreading.
- 261.06 Create multiple threads.
- 261.07 Use the synchronized keyword.
- 261.08 Describe the life cycle of a thread.

259.0 DEMONSTRATE PROFICIENCY IN PREPARATION FOR JAVA FINAL EXAM-The student will be able to:

- 262.01 Pass the quizzes associated with the Specialized Programming course.
- 262.02 Demonstrate understanding of Java concepts.
- 262.03 Practice exam format and questions.
- 262.04 Write code that declares, constructs, and initializes arrays of any base type using any of the permitted forms both for declaration and for initialization.

- 262.05 Declare classes, inner classes, methods, instance variables, static variables, and automatic (method local) variables making appropriate use of all permitted modifiers (such as public, final, static, aCCCract, etc.).
- 262.06 State the significance of each of the permitted modifiers (public, final, static, aCCCract, etc.) both singly and in combination, and state the effect of package relationships on declared items qualified by these modifiers.
- 262.07 For a given class, determine if a default constructor will be created, and if so, state the prototype of that constructor.
- 262.08 State the legal return types for any method given the declarations of all related methods in this or parent classes.
- 262.09 Write code using if and switch statements and identify legal argument types for these statements.
- 262.10 Write code using all forms of loops including labeled and unlabeled use of break and continue, and state the values taken by loop control variables during and after loop execution.
- 262.11 Write code that makes proper use of exceptions and exception handling clauses (try, catch, finally) and declares methods and overriding methods that throw exceptions.
- 262.12 State the behavior that is guaranteed by the garbage collection system, and write code that explicitly makes objects eligible for collection.
- 262.13 Identify correctly constructed source files, package declarations, import statements, class declarations (of all forms including inner classes), interface declarations and implementations (for java.lang.runnable or other interface described in the test), method declarations (including the main method that is used to start execution of a class), variable declarations and identifiers.
- 262.14 State the correspondence between index values in the argument array passed to a main method and command line arguments.
- 262.15 Identify all Java programming language keywords and correctly constructed identifiers.
- 262.16 State the effect of using a variable or array element of any kind when no explicit assignment has been made to it.
- 262.17 State the range of all primitive data types and declare literal values for string and all primitive types using all permitted formats, bases, and representations.
- 262.18 Determine the result of applying any operator, including assignment operators, instanceof, and casts to operands of any type, class, scope, or accessibility, or any combination of these.
- 262.19 Determine the result of applying the boolean equals(object) method to objects of any combination of the classes java.lang.string, java.lang.Boolean, and java.lang.object.
- 262.20 In an expression involving the operators &, ξ, &&, ξξ, and variables of known values, state which operands are evaluated and the value of the expression.
- 262.21 Determine the effect upon objects and primitive values of passing variables into methods and performing assignments or other modifying operations in that method.

- 262.22 State the benefits of encapsulation in object oriented design and write code that implements tightly encapsulated classes and the relationships "is a" and "has a."
- 262.23 Write code to invoke overridden or overloaded methods and parental or overloaded constructors and describe the effect of invoking these methods.
- 262.24 Write code to construct instances of any concrete class including normal top-level classes, inner classes, static inner classes, and anonymous inner classes.
- 262.25 Write code to define, instantiate, and start new threads using both `java.lang.thread` and `java.lang.runnable`.
- 262.26 Identify conditions that might prevent a thread from executing.
- 262.27 Write code using `synchronized`, `wait`, `notify`, or `notifyall`, to protect against concurrent access problems and to communicate between threads.
- 262.28 Define the interaction between threads and between threads and object locks when executing `synchronized`, `wait`, `notify`, or `notifyall`.
- 262.29 Write code using component and container of the `java.awt` package
- 262.30 Write code using the following methods of the `java.lang.math` class: `abs()`, `ceil()`, `floor()`, `max()`, `min()`, `random()`, `round()`, `sin()`, `cos()`, `tan()`, `sqrt()`.
- 262.31 Describe the significance of the immutability of string objects.
- 262.32 Make appropriate selection of collection classes/interfaces to suit specified behavior requirements.

260.0 CREATE AN INTERFACE THAT DEMONSTRATES MOST OF THE CONCEPTS LEARNED THROUGHOUT THE PROGRAM COURSE-The student will be able to:

- 265.01 Identify a local business or select a case study to be used for final database project.
- 265.02 Create a list of questions that will provide a basis for developing an entity-relationship diagram.
- 265.03 Develop and normalize an entity relationship diagram.
- 265.04 Document an initial database design on Table Instance Charts.
- 265.05 Create a graphical user interface that allows a user to connect to the database.
- 265.07 Evaluate and generate criteria for a formal, business presentation.
- 265.08 Construct a persuasive group presentation using the guidelines set forth in class.
- 265.09 Deliver a formal business presentation for the class that discusses the group's proposed solution.
- 265.10 Demonstrate the functionality of the database application.
- 265.11 Self-assess learning experience through the presentation and demonstration of their final database project.