DTC - Microsoft Sql Server

Sifal, Kathmandu Phone No.: 01-446-7153

Dataasbe Design And Implementation: Level 1

OBJECTIVES

The DTC - Dataasbe design and Implementation – Level 1 course is targeted for beginners who want to:

- Learn how to think and write meaningful piece of code in Dataasbe design and Implementation.
- Learn how to read Dataasbe design and Implementation code that has been written by somebody else.
- Learn how to map literary description of a problem (requirement) to an application/library coded in Dataasbe design and Implementation. In summary, this course teaches how to program using Dataasbe design and Implementation programming language.

This is a core basic level course that is essential for anyone who have no prior programming experience but wish to be a professional Dataasbe design and Implementation engineer in future

TARGET GROUP

- Anyone who has some basic knowledge about programming and wants to learn to write applications in Dataasbe design and Implementation for any purpose e.g. curiosity, hobby, to complete an academic project, to work towards a career as Dataasbe design and Implementation programmer, to help in project management, etc.

Prerequisites:

- Basic knowledge about programming, bits/bytes, procedures, classes, computer architecture, etc. If you just have a theoretical knowledge that is perfectly okay but you should have strong convictions on what programming is, and what you hope to achieve from this class.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Dataasbe design and Implementation (self-study and practice).
- There is no prior educational level requirement for this course. Anyone from 10+2 student to someone who is doing her PHD in Genetic Engineering is welcome to take this course.
- If you are only interested in theory and have no interest/patience in spending at least 10 hours every week throughout the duration of the course, then this course is clearly not for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six -odd months, then this class may not be for you.

TRAINING METHOD

The course is spread over 40 hours that consists of lecture and lab work. There will be approximately 10 hours of lectures and 30 hours of hands-on lab work.

- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So, in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but especially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Dataasbe design and Implementation.

COURSE DURATION

- 40 hours
- Classes
  - ✓ Morning/Evening

COURSE BREAKDOWN

Theory:

1. RELATIONAL DATABASE FUNDAMENTALS
    - Overview of Relational Database Concepts
    - Relational Databases and Relational Database Management Systems
    - Data Normalization

2. CONCEPTUAL DATA MODELING
    - Problems with File-based System
    - Concept of Data Model
    - 3-Tier Architecture
    - Data Mapping
    - Data Model and its Types
    - THE RELATIONAL DATA MODEL
    - Data Modeling Using ERD,
    - Problems of Using ERDs and Solutions,
3. EERD and Chen Notation,
    - Relational Database Model Terminologies and their Implementation,
    - Database Relations and their Characteristics,
    - Relational Keys and Integrity Constraints

4. RELATIONAL DATABASE DESIGN
    - Database Design Methodologies,
    - Conceptual, Logical and Physical Database Designs,
    - Mapping ERD into Relational Schema

5. CREATING A DATABASE
    - Database Development Methodology Overview
    - Building a Logical Data Model
    - Identifying Entities and Attributes
    - Isolating Keys
    - Relationships Between Entities
    - Creating Entity-Relationship Diagrams
    - Transforming to Physical Design
    - Migrating Entities to Tables
    - Selecting Primary Keys
    - Defining Columns
    - Enforcing Relationships with Foreign Keys
    - Constructing the Database Using DDL
    - Creating Tables, Indexes, Constraints and Views
    - Dropping Tables, Indexes, Constraints and Views
    - Modifying Tables, Indexes, Constraints and Views

6. WRITING BASIC SQL QUERIES
    - Displaying Table Structures
    - Retrieving Column Data from a Table or View
    - Selecting Unique Values
    - Filtering Rows Using the WHERE Clause
    - Sorting Results Using ORDER BY
    - Joining Multiple Tables
    - Using Column and Table Aliases

7. MANIPULATING QUERY RESULTS
    - Using Row Functions
    - Character
    - Numeric
    - Date and Time
    - Data Conversion (CAST and CONVERT)

8. USING THE CASE FUNCTION
    - Handling Null Values

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

DISCLAIMER

Please note that Deerwalk Training Center serves the right to change the course syllabus of DTC - Dataasbe design and Implementation – Level 1 course at any time without prior notification.

Dataasbe Design and Implementation: Level 2

OBJECTIVES

The DTC - Dataasbe design and Implementation – Level 2 course is targeted for trainees:

- Who have had some prior beginner level hands-on programming experience in Dataasbe design and Implementation programming language.
- Who have programming experience in some other programming language (e.g. Dataasbe design and Implementation, Obj-C, PHP, C, C++, etc.) and want to learn Dataasbe design and Implementation .

TARGET GROUP

- High school and university students (undergraduate, graduate, etc.) who want to do coursework (e.g. project, etc.) in DATAASBE DESIGN AND IMPLEMENTATION.
- Someone who has experience in some other programming language (e.g. C/C++, PHP, Perl, etc.), but has never done programming in ANDROID.
- Someone who is already working as a professional VB.NET developer and wants to switch to ANDROID.
- Someone who did her undergraduate in Economics, has been working in Media sector since graduation, and also working as a professional freelance PHP developer.
- Electrical/Electronic undergraduates in their 3rd semester who want to beef up their software skills prior to graduation.

Prerequisites:

- Successfully complete the entrance test with score of at least 40% (for trainees directly applying to this level).
- Successfully complete the DWIT Training - Dataasbe design and Implementation – Level 1 course (not applicable to trainees directly applying to this level).
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Dataasbe design and Implementation (self-study and practice).

TRAINING METHOD

- The course is spread over 40 hours that consists of approximately 15 hours of lecture and 25 hours of hands-on lab work.

- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So, in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but especially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Dataasbe design and Implementation.

COURSE DURATION

- 40 hours
- Classes
  - ✓ Morning/Evening

COURSE BREAKDOWN

1. ADVANCED QUERY TECHNIQUES
   - Inner Joins
   - Outer Joins (Left, Right, Full)
   - Joining a Table to Itself
   - Subqueries
   - Tips for Developing Complex SQL Queries
   - Using Aggregate Functions
     - AVG
     - COUNT
     - SUM
     - MIN
     - MAX
   - Aggregating Results Using GROUP BY
   - Restricting Groups with the HAVING Clause

2. USER-DEFINED FUNCTIONS
   - Definition and Benefits of Use
   - CREATE FUNCTION
     - Syntax
     - RETURN Clause and the RETURNS Statement
     - Scalar vs. Table Functions
     - Comparison with Stored Procedures
     - Returning Scalar Values and Tables
   - ALTER and DROP FUNCTION

3. MANIPULATING TABLE DATA USING SQL'S DATA MANIPULATION LANGUAGE (DML)
   - Inserting Data into Tables
   - Updating Existing Data
   - Deleting Records
   - Truncating Tables
   - Implementing Data Integrity with Transactions
     - Beginning Explicit Transactions
     - Committing Transactions
     - Rolling Back Transactions

4. STORED PROCEDURES
   - Definition and Benefits of Use
   - CREATE PROCEDURE
     - Syntax
     - Variables and Parameters
   - Control of Program Flow
   - ALTER and DROP PROCEDURE
     - Implementation Differences

5. WORKING WITH VIEWS
   - Benefits of Using Views
   - Creating Views
     - ALTER and DROP View

6. TRIGGERS
   - Definition and Benefits of Use
   - Alternatives (e.g., Constraints)
   - CREATE TRIGGER
     - Syntax
     - Trigger Types
   - "Inserted" (or "NEW") and "Deleted" (or "OLD") Tables
   - Event Handling and Trigger Execution
     - ALTER and DROP TRIGGER

7. DATA WAREHOUSING
   - Database Warehouse Architecture,
   - RAID,
   - Parallelism and Partitioning,
   - ETL,
   - Data Mining

8. DATABASE RECOVERY AND SECURITY
   - Logical and Physical Database Security,
   - SQL Injections,
   - Database Recovery Mechanism,
   - Deferred and Immediate Update

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

DISCLAIMER

Please note that Deerwalk Training Center reserves the right to change the course syllabus of DTC - Dataasbe design and Implementation – Level 2 course at any time without prior notification.