

Course Code	
Course Name	Programming Fundamentals
Duration (in days)	3.5
Pre-requisites	None

Proficiency Level	Fundamentals
Target Audience	Campus Hires

Learning Outcome

At the end of the program, participants will be able to learn:

- Define Software Engineering and its role in software development.
- Understanding Flowchart & Pseudocode.
- Database, DBMS, and RDBMS concepts
- DBMS Architecture
- Data Models
- Relation Database concepts
- Normalization Process and different Normal forms
- ER Diagram
- Key concepts of the Git source control system
- Install and configure the MySQL server and clients.
- Use Structured Query Language (SQL) to build your database.
- Understanding data types and their constraints.
- Querying datasets and perform data manipulation using SQL commands.
- Implementing join and set operations.
- Implementing SQL subqueries.
- Grouping & Aggregation Operations.
- Understanding Views and Stored Procedures

Day-wise Session Plan

Day	Unit	Objective(s)	Hours
1	Software Engineering & SDLC Phases	<ul style="list-style-type: none"> • Evolution of Software • Life Cycle Phases • Planning Analysis • Requirements Analysis • Design and Prototyping • Development of the Application • Testing and Deployment • Project Management 	4
1	Flow Chart and Pseudocode	<ul style="list-style-type: none"> • Pre-code planning • Pseudocode • Verify Algorithm 	4

		<ul style="list-style-type: none"> Flowchart 	
2	Architecture and Normalization Concepts	<p>[REFRESHER]</p> <ul style="list-style-type: none"> Describe a DBMS, its components, and advantages for users. Describe the features and characteristics of flat-file, hierarchical, and XML database models. Levels of a DBMS architecture Types of constraints Describe normalization in relation to designing a database. Perform first normal form when designing a database. Perform second normal form when designing a database. Perform third normal form when designing a database. Perform BCNF when designing a database. 	2
2	ER Diagram	<ul style="list-style-type: none"> Describe entity-relationship modeling for a RDBMS Define Entities, Attributes, Relationships Degree of relationships Cardinality of relationships Relational Database Model Create an ERD for a database based on a Scenario. 	4
2	Git Essentials	<p>[REFRESHER]</p> <ul style="list-style-type: none"> What is Git? How to Install Git on Windows? What is GitHub? Git commands. Git vs. GitHub. What is GitLab? Git Clone Commands. Git Push Commands. Git Pull Commands. Git History Branching and Merging Resolve Merge Conflicts in Git 	2
3	Introduction to SQL	<p>[REFRESHER]</p> <ul style="list-style-type: none"> What is a Database? What is SQL? What is MySQL? SQL Commands 	1

3	Database Connection	[REFRESHER] <ul style="list-style-type: none"> • Launch MySQL Workbench • Connect to MySQL Server • Creating a new Database • Data Types • CAST or CONVERT • Keys in SQL • Constraints 	1
3	DDL Commands	[REFRESHER] <ul style="list-style-type: none"> • DDL Commands • Add table to Database • Describe Table • Alter Table • Modify and Drop Clause • Data manipulation 	2
3	Query Clauses	<ul style="list-style-type: none"> • Database schema • Import Data • Query Clauses • Column Alias • Table Alias 	1
3	Query Multiple Tables	[REFRESHER] <ul style="list-style-type: none"> • Introduction to joins • Types of joins • Inner Join • Left Outer Join • Right Outer Join • Full outer Join • ANSI Join Syntax • Self-Join • Equi and non-equi Join • Set Operations 	1
3	Functions in SQL	[REFRESHER] <ul style="list-style-type: none"> • String Functions • Numeric Functions • Date Functions • Aggregate Functions • Generate Groups 	1
3	SQL Subqueries	[REFRESHER] <ul style="list-style-type: none"> • SQL subqueries • Correlated subqueries 	1

		<ul style="list-style-type: none"> • Non-correlated subqueries 	
4	Advanced Queries	<ul style="list-style-type: none"> • Views • Index • Transaction Control Commands • Stored procedures • Difference between Procedure and Function • Creating a Procedure • Creating a Function 	4
		Total	28