Java Lectureflow

Module-1) SE - Overview of IT Industry	5

- Introduction of students
- Career in IT
- Understanding Student Login of TOPS ERP
- Using Lab
- What is Program
- What is programming?
- Types of Programming Language
- World Wide Web
- How Internet Works
- Network Layers on Client and Server
- Client And Servers
- Types of Internet Connections
- Protocols
- Application Security
- Software Applications and its types
- Software Architecture
- Layers in Software Architecture
- Software Environments
- Types of Programming Languages
- Source Code
- Github and introductions
- Student Account in Github
- Types of Software
- Introduction of Software
- Application software
- Software development process
- Software Requirement
- Software Analysis
- System Design
- Software Testing
- Maintenance
- Development
- Web Application
- Designing
- moble application
- DFD
- Desktop Application
- Flow Chart

Module 2) SE - Introduction to Programming - May 2024	15

- Overview of C Programming - What is C Programming? History and Evolution, Importance and Applications
- Setting Up Environment -Installing a C Compiler (e.g., GCC), Choosing an IDE (DevC++. VS Code, Codeblocks, etc) Writing Your First Program
- Basic Structure of a C Program - Structure of a C Program ,Comments in C, Data Types and Variables, Constants, Keywords and Identifiers
- Operators - Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Bitwise Operators, Conditional Operator
- Control Flow Statements - Decision-Making in C - If Statement, If_Else statement, If_elseif (Elseif Ladder), Nested if-else statement, Switch statement
- Looping in C - While Loop, For Loop, Do-While Loop
- Loop Control Statements - Break, Continue, Go to
- Functions in C - Introduction to Functions, Function Declaration and Definition, Function Call and Return
- Arrays in C - Introduction to Arrays, One-Dimensional Arrays, Multi-Dimensional Arrays
- Pointers in C - Introduction to Pointers, Pointer Declaration and Initialization
- Strings in C - Introduction to Strings, String Handling Functions, strlen, strcpy, strcat, strcmp, strcmpi,strchr, String Input and Output
- Structures in C - Introduction to Structures, Structure Declaration and Initialization, Array of Structure Nested Structures
- File Handling in C, File Operations (Opening, Closing, Reading, and Writing), File Pointers, File Handling Functions

Module-3) Introduction to OOPS Programming	8

- Introduction to C++ - Understanding the Basics of Programming, Introduction to C++ Language, POP Vs OOP, Advantages of OOP, Setting Up C++ Development Environment, Writing and Running Your First C++ Program, Input and Output Operations in C++
- Variables, Data Types, and Operators - Variables and Constants in C++, Data Types and Size Specifiers, Assignments, Arithmetic, Relational, Logical, and Bitwise Operators, Type Conversion in C++, Constants and Literals
- Control Flow Statements - Conditional Statements: if, if_else, else if ladder, nested if, Switch Statement, Loops: while, do-while, for, Break and Continue Statements, Nested Control Structures
- Functions and Scope - Introduction to Functions ,Function Prototypes and Definitions, Parameters and Return Values, Scope of Variables
- Arrays and Strings - Introduction to Arrays, Single-Dimensional and Multi-Dimensional Arrays, Array Initialization, Accessing Elements, and Manipulation, Introduction to Strings in C++, String Operations and Functions
- Introduction to Object-Oriented Programming -Understanding the Basics of Object-Oriented Programming, Advantages of OOP Paradigm, Key Concepts: Classes, Objects, Inheritance, Polymorphism, Encapsulation, Introduction to C++ as an Object-Oriented Language

- Classes and Objects - Declaring Classes and Objects in C++, Class Members: Data Members and Member Functions, Constructors and Destructors, Access Specifiers: Public, Private, Protected, Class Member Functions: Inline and Outside Definitions
- Inheritance - Concept of Inheritance and Reusability, Types of Inheritance: Single, Multiple, Multi-level, Hierarchical, Base and Derived Classes in C++, Access Control and Inheritance, Constructors and Destructors in Inheritance
- Polymorphism - Understanding Polymorphism in OOP, Compile-Time and Runtime Polymorphism, Function Overloading and Operator Overloading, Virtual Functions and Dynamic Binding, Abstract Classes and Pure Virtual Functions, Scope resolution operator, Static Keywords, Inline Function
- Encapsulation - Understanding Encapsulation and Information Hiding, Access Specifiers: Public, Private, Protected, Encapsulation in C++ Classes, Benefits of Encapsulation in OOP, Friend Functions and Friend Classes
- File Handling - Introduction to File Handling in C++, Opening, Closing, Reading, and Writing Files, Error Handling in File Operations, Working with File Streams

Module-4) SE - Introduction to DBMS	**9**

- Introduction to SQL - SQL Overview, Definition of SQL, Importance of SQL in Database Management, DBMS-RDBMS
- SQL Syntax - Basic SQL Syntax, Structure of SQL Statements
- SQL Constraints - Types of Constraints (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY), Implementing Constraints in Tables
- Main SQL Commands and Sub-commands - Data Definition Language (DDL), CREATE Command, Creating Tables, Specifying Column Names, Data Types, and Constraints
- ALTER Command - Modifying Existing Tables, Adding, Modifying, or Dropping Columns
- DROP Command - Deleting Tables from the Database
- Data Manipulation Language (DML) - INSERT Command, Adding Data into Tables Specifying Column Names and Values
- UPDATE Command - Modifying Existing Data in Tables, Changing Values in Specific Columns
- DELETE Command - Removing Data from Tables, Deleting Specific Rows with WHERE Clause
- Data Query Language (DQL) - SELECT Command, Retrieving Data from Tables, Filtering Data with WHERE Clause, Sorting Data with ORDER BY Clause, Limiting Results with LIMIT or FETCH FIRST Clause
- Data Control Language (DCL) -GRANT Command, Granting Privileges to Users or Roles, Granting SELECT, INSERT, UPDATE, DELETE Permissions
- REVOKE Command - Revoking Privileges from Users or Roles, Removing SELECT, INSERT, UPDATE, DELETE Permissions
- Transaction Control Language (TCL) - COMMIT Command, Saving Changes Permanently to the Database
- ROLLBACK Command - Reverting Uncommitted Changes, ?SAVEPOINT Command - Creating Intermediate Points in a Transaction for Rollback
- SQL Joins - Inner Join, Left Join, Right Join, Full Outer Join
- SQL Group By - Grouping Data in SQL Queries

- SQL Stored Procedure - Definition and Purpose of Stored Procedures, Creating and Executing Stored Procedures
- SQL View - Creating Views in SQL, Advantages of Using Views
- SQL Trigger - Introduction to Triggers, Types of Triggers (INSERT, UPDATE, DELETE)
- Introduction to PL/SQL - Definition and Purpose of PL/SQL, Benefits of Using PL/SQL
- PL/SQL Syntax - Structure of PL/SQL Blocks, Variables, Constants, and Data Types in PL/SQL
- PL/SQL Control Structures - IF-THEN, IF-THEN-ELSE, CASE Statements ,Loops (WHILE, FOR)
- SQL Cursor - Introduction to Cursors in PL/SQL, Implicit vs. Explicit Cursors
- Rollback and Commit Savepoint - Transaction Management in PL/SQL

Module 6) - Java - Core Java	**18**

- Conditional Statements (If, If Else, Nested If Else If)
- Introduction of Core Java
- Practical Example : 1. Odd-Even, 2. Prime Number, 3. Max out of three, 4. Student's grade system
- Eclipse IDE
- (Switch Case)
- JVM,JDK,JRE
- Practical Example : 1. Mini Calculator
- Class, Object Constructor
- Loops (While, Do While, For)
- Class, Object, Method
- Practical Example : 1. Sum of n numbers, 2. patterns, 3. prime numbers for a range
- Constructor
- Break and Continue
- Garbage Collection
- Practical Example : 1. exit or continue from loop using break & continue
- Finalize
- Project Analysis
- Source File Layout
- Analysis In Details
- Package Management, Modifiers- Public, Private, Protected, Default
- Import Statement
- Context Level
- Data types
- First Level
- Primitive Types
- Second Level
- Reference Types
- Array Introduction
- Data Dictionary
- Modifiers - Public, Private, Protected, Default
- Why Array? Advantages
- Flow Chart

- Types of Array
- Resizing Array
- Copying Array
- Primitive types and Reference type Arrays
- Encapsulations
- Advantages of Inheritance
- Types of Inheritance
- Practical of Inheritance
- Practical of Inheritance with Constructor
- Polymorphism
- Types of Polymorphism
- Method Overloading and Method Overriding
- Abstract and Interface - Introduction and Difference
- Keywords - This, Static, Final, Super
- Classes
- Object Class(only Important Methods)
- String Class (Only Important Methods)
- String Buffer & String Builder
- Wrapper Classes
- Exceptions
- Introduction - Why Exceptions
- Types of Exceptions
- Try catch and Finally Block
- Multi catch Exceptions
- Throw and Throws keywords
- Method Overriding with Exceptions
- Custom Exceptions
- FILE I/O
- What is Stream and Types of Stream
- File Input Output Streams and Its Methods
- File class
- Command Line Arguments
- Thread-Introduction
- Thread Life Cycle
- Creating Threads
- Thread Class Methods (Only Important Methods)
- Runnable Interface
- Synchronized block and Synchronized Methods
- Collection Framework - Introduction
- Collection API
- Hierarchy of Collections
- List and Set and Map Collections
- Array list, vector and •Linkedlist, Hashset, Hashmap, Map, Hashtable
- Generics

- Comparator and Comparables
- •Java 8 Features •Lambda Expressions, Functional Interfaces, Stream API, Date/Time API, Collection API Improvements, Concurrency API Improvements •JAVAFX - 2D Shapes -line, rectangle , circle •Javafx color, JavaFX Layout, JAVAFX UI components, JAVAFX chart-pie •JAVAFX Event Handling, JAVAFX playing audio and video
- Javafx and Java8 api
- All Components
- Events, Event Handling
- Practical Example : 1. Create class box with three variable height, width, depth. Create default, parameterized and copy constructor. Create one method called volume to show width*height*depth. Call all constructor and volume method for all constructor
- Practical Example : 1. Create class named student with variable rno, fname, lname, email, mobile. create 2 methods to get student data and print them.
- Practical Example : 1. One dimensional array(get data by scanner and print it). 2. Array array elements in asceding and desceding order
- Practical Example : 1. Perform constructor chaining
- Practical Example : 1. Print the current thread that is by default available and then change it's name and again print it. 2. Create a thread usinf Runnable interface. 3. Create a thread using Thread class. 4. Create multiple thread and execute it in main method. 5. Create multiple thread and execute them simultaneouly and achive synchronization. 6. Create two synchronized thread and perform deadlock
- Practical Example: 1. Create 2 two dimensional array and perform matrix addition, subtractio and multiplication
- Practical Example: 1. Create abstract class RBI with one abstract method interest rate and extend this class in three class SBI, HDFC, Kotak to implement abstract method. 2. Create interface with 2 method. Implement this method in two class. 3. Create program for inheritance of interface. 4. Create program to implment static method in interface and call it in a class
- Practical Example: 1. Create custome excepton insufficientfund. Create class named bank and create two methods deposite and withdraw. If withdrawal amount is greater than balance then throw user defined exception and handle it.
- Practical Example: 1. Create swing GUI with id, fname, lname & email and perform CRUD operation with mysql datanase.
- Practical Example: 1. Demonstrate the devide by zero, input mismatch exception and arrayindexoutofbounds exception in a multi catch and multi try statement. 2. Create a method called demo and enter user defind integer value at runtime, if user enters negative value ask again to put value using recursion others wise throw an exception and handle it. 3. Create above program using throws clause without recusrion. 4. Demonstrate the finally block. 5. How to use exception in method override
- Practical Example: 1. Make a ArrayList with different type of data and perform it's different method. 2. Iterate ArrayList data in both direction from first to last and last to first. 3. Demonstrate HashSet with it;s method. 4. Demonstrate HashMap and iterate it's data. 5. Perform enumeration with Vector class. 6. Create generic method to print different types of array of diffent wrapper classes. 7. Demonstrate Comparator 8. Demonstrate Comparable.
- Practical Example: 1. Method overloading. 2. Method overriding 3. Dynamic method dispatch to solve method override

- Practical Example: 1. Pass the 2 integer values through command line and print the maximum number from this.
- Practical Example: 1. Perform singleinheritance. 2. Multilevel. 3. Hierarchical.
- Practical Example: 1. String class & its method 2.Perform StringBuffer class methods
- Practical Example: 1. Write a progra to write 1 string data into the file using FileOutputStream and read that file using FileInputStream.
- Practical Example: 1. Write a program to show the use of this keyword in assigning values to variable, as argument in constructor and method, call the default constructor in parameterized constructor using this, and call the method using this. 2. Write a program to demonstrate the difference between static and non static variable. 3. Write a program to to create a static method and static block. 4. Demonstrate teh use of final variable, mthod and class. 5. Access the variable, methods and constructor from d
- Practical Example: 2. Do above operation using FileWritter & FileReader
- Practical Example: 3. Create one class name student with rno, fname, lname & email and store values of variable into object and then write that object into file and read it.
- Practical Example: 4. Print all the basic property of file that is available in your c:\ drive. You create tops1.txt and put some text into it.

Module 7) Java - RDBMS & Database Programming With JDBC	**6**

- JDBC (Insert, Update, Select, Delete)
- Introduction of JDBC
- Driver Types
- Steps for Creating Connections
- Types of Statements (Statements, prepared Statements and Callable Statements)
- Result Set Interface
- Database Metadata
- Result Set Metadata
- Practical Examples: SQL Queries
- Practical Example : 1. Create swing GUI with id, fname, lname & email and perform CRUD operation with mysql datanase. 2. Demonstrate callble statement in & out parameter.

Module 15) Java - Web Technologies In Java	**12**

- HTML Tags anchor,form,table,image,all list tags,paragraph,break,label
- CSS –Inlince CSS,Interbal CSS,External CSS • Margin and Paddings • Psecudo class • Css id and class
- Introduction of Client Server Architecture
- HTTP Protocol overview with Request and Response header explanation
- J2EE Architecture Overview
- Web Component Development In Java CGI Programming Process Advantage and Disadvantage
- Servlet Programming Introductions Advantage and Disadvantage

- Servlet Versions, Types of Servlets
- Difference between HTTP Servlet an Generic Servlet
- Servlet Life Cycle
- Creating Servlets Servlet Entry in web.xml
- Logical URL Servlet Config Interface
- Request Dispatcher Interface Forward and Include Methods
- Request Dispatcher Interface
- Servlet Context Interface Web Application Listener Scope of Objects, Request and Response Application (Context)
- Practical Example: 1. Fetch data from web.xml to particular servlet using ServletConfig. 2. Fetch data from web.xml to multiple servlet using ServletContext. 3. Create one registration form in jsp and send data to servlet, from servlet again send data to jsp using RequestDispatcher. 4. Create login form in jsp and after login send uname & password to servlet, check data if not blank go forward and if blank then include login.jsp page to servlet.
- Java Filters - Introduction What are the needs Filter Life Cycle Process of Execution Filter Applying Filter Entry in web.xml URL Pattern with Filter
- Practical Example: 1. Perform server side validation using filter.
- Action JSTL Custom Tags
- Comments
- Declaration Implicit Objects
- Directives - Scriplets
- Expression
- JSP Life Cycle
- JSP Translation
- Practical Example: JSP Translation JSP Life Cycle Comments Directives Scrip lets Expression Declaration Implicit Objects Action JSTL Custom Tags
- Cookies Session
- Hidden Form Fields
- Session Management - Introduction
- Session Tracking Technique
- URL Rewriting
- What are needs?
- Practical Example: 1. Create registration form, after validation insert data to database and redirect to login form, if successfull login manage session data and logout. 2. Create complete CRUD operation for user profile management.

Module 16) Java - Software Design Pattern And Project	**15**

- Software Design Pattern and Project MVC + DAO
- Session Management (Session, Cookie, Hidden Form Field , URL Rewriting)
- Project Cover Topics: 1) Template Integration 2) Image Up/Dwn 3) Mail Integration 4) OTP Via Mail Integration 5) Online Payment Integration 6) AJAX

Module 17) Java - Hibernate Framework	5

- Introduction Hibernate Architecture
- Hibernate Relationship(one to one, one to many , many to one , many to many)
- Hibernate CRUD Example

Module 18) Java - Spring	7

- Introduction
- BeanFactory and application Context
- Container Concepts
- Spring Data JPA Template
- MVC

Module 19) Java - Spring Boot	5

- Introduction to STS
- MVC (Template Integration, CRUD , Form Validation , Pagination), AOP
- Security, Role based Authentication, OAuth2, Token based authentication

Module 20) Java - Spring WebServices	6

- Introduction to WebService
- Basics of REST APIs
- MVC, AOP
- Spring REST(CRUD API, Pagination, Fetch from Multiple Table , Image Up/API)

Module 21) Java - Microservices with Spring Boot, Spring Cloud	6

- Microservices with Spring Boot, Spring Cloud
- Introduction to MicroService architecture
- Advantages with MicroService over Monolithinc architecture
- Develop and Deploy Microservice application in localhost Introduction to Service Discovery ,Eureka server
- Client side Discovery pattern
- Server side Discovery pattern
- Load Balancing configuration