Java Lectureflow

Module 12) - Java -Introduction	1

- Introduction Lecture
- Introduction of students
- Understanding Student Login of TOPS ERP
- Working on Project and Assignment
- Using Lab
- Career in IT

Module 13) - Java - Core Java	15

- 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. 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. One dimensional array(get data by scanner and print it). 2. Array array elements in asceding and desceding order
- Practical Example: 1. Create 2 two dimensional array and perform matrix addition, subtractio and multiplication
- Practical Example: 1. Perform singleinheritance. 2. Multilevel. 3. Hierarchical.
- Practical Example : 1. Perform constructor chaining
- Practical Example: 1. Method overloading. 2. Method overriding 3. Dynamic method dispatch to solve method override
- 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. 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: 1. String class & its method 2.Perform StringBuffer class methods
- 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. 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. Write a progra to write 1 string data into the file using FileOutputStream and read that file using FileInputStream.
- 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.
- Practical Example: 1. Pass the 2 integer values through command line and print the maximum number from this.
- 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. 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. Create swing GUI with id, fname, lname & email and perform CRUD operation with mysql datanase.
- 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
- SDLC Process
- Project Analysis
- Source File Layout
- Analysis In Details
- DFD (with practical)

- Package Management, Modifiers- Public, Private, Protected, Default
- Introduction of DFD
- Import Statement
- Rules for Drawing DFD
- 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 Other Classes
- Generics
- Comparator and Comparables
- JAVA GUI
- AWT (Introduction only) & Swing (in Details)
- Components, Containers, Frame, Window, Panel, Layout
- All Components
- Events, Event Handling

Module 14) Java - RDBMS & Database Programming With JDBC	**5**

- Database
- DBMS and RDBMS
- Introduction MYSQL
- Mysql IDE
- Query Types
- DDL, DML, DQL, DCL
- Constraints : Primary Key, Foreign Key, Unique Key
- Normalizations: 1NF 2NF 3NF
- Joins: All Joins Types
- Advance Database: Indexers Views Procedures Functions Cursor, Triggers
- 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 UL, Tag LI, Tag a, Tag IMG, tag Table, TR, TD, tag
- Form tags with Attributes
- All input tags CSS
- Types of CSS Pseudo- Classes Margins and Puddings
- CSS background
- CSS using ID and Class
- JavaScript Events
- Validations with Regular Expressions
- Firebug Template Integration
- Practical Example: 1. Basic HTML Tags 2. Create Registration form and perform required and regular expression validation for firstname(only alphabets allowd), email(standard email id), mobile number(only 10 digits). 3. Perform all type of css, class & id, pseudo code.
- 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 - Rest Framework - Industry	**10**

- Design Pattern
- MVC Design Pattern with Example
- AJAX Programming With Example
- Practical Example: 1. Perform dynamic search operation in project using AJAX. 2. Register user with unique email using AJAX
- Introduction to Distributed Technologies RMI, EJB and WEB Services Introduction Types of Web Services What is Restful Web Services? Restful Web Services Annotations Restful Web Services with Exampl
- ractical Example: 1. Restful web service CRUD operation
- RESTful API: Representational State Transfer (REST) is a widely used architectural style for building web services. Understanding REST principles and being able to create RESTful APIs is essential. CRUD API: CRUD stands for Create, Read, Update, and Delete, which are the basic operations performed on data. Creating APIs that allow these operations is fundamental to backend development. Authentication and Authorization API: Knowing how to implement user authentication and authorization mechanisms is crucia
- OpenWeatherMap API: This API provides weather data for various locations worldwide. You can retrieve current weather conditions, forecasts, and historical weather data.
- Google Maps Geocoding API: This API allows you to convert addresses into geographic coordinates (latitude and longitude) and vice versa. You can use it to retrieve location data, calculate distances between points, and display maps.
- GitHub API: GitHub provides an API that enables you to interact with repositories, issues, pull requests, and more. You can perform actions like retrieving repository information, creating issues, and accessing user data.
- Twitter API: Twitter offers an API that allows you to integrate Twitter functionality into your applications. You can fetch tweets, post tweets, retrieve user information, and perform searches.
- REST Countries API: This API provides information about countries, including details like population, languages spoken, currencies, and more. You can retrieve country-specific data and use it for various applications.

- Social authentication (For eg; Login with Google, Login with Facebook...etc)
- Email sending APIs (For eg; Mailchimp, Mailgun...etc)
- SMS sending APIs (For eg; Twilio)
- Normal payments (For eg; Paypal, Stripe)
- - Google Maps API

Module 17) java - Frameworks - Industry	**16**

- All Core Interface Query and Criteria Named Query
- Relationships Many to Many
- Relationships One to Many
- Relationships Many to One
- Hibernate Introduction
- Relationships One to One
- Hibernate Architecture
- All Database Operations with hibernate
- Practical Example: 1. CRUD Operation with hibernate using xml files. 2. CRUD operation with hibernate using annotation. 3. Perform onetoone relationship(Employee class with eid, uname and password & EmployeePersonalINfo class with epid, fname, lname,email). 4. Perform onetomany & manytoone relationship(Employee class with eid, fname,lname,email & Department class with deptno, dname,location). 5. Perform manytomany relationship(Student class with sid, sname & Course class with cid, cname)
- Introduction of Spring Framework Architecture
- Overview Of Spring Framework
- Core Container AOP
- Spring DAO (Data Integration)
- Spring Using IDE, Using Library Spring Hello World Example
- Practical Example: 1. Hello world spring app to introduce spring framework
- 1) Spring IOC Container 2) Bean Factory 3) Application Context Spring Bean Definition 4) Configuration 5) Life Cycle 6) Inheritance 7) Scopes
- Practical Example: 2. Perform spring inheritance, life cycle & abstraction. 3. Perform singleton & prototype scope to use spring beans variable
- 1) Spring Dependency Injection 2) Constructor based 3) Setter Getter based 4) Inner Beans , Aliases and ID-ref Collections and References 5) Auto Wiring
- Practical Example : 4. Demonstrate spring dependency injection by setter method.5. Demonstrate spring dependency injection by constructor. 6. Demonstrate spring dependency injection by object. 7. Perform inner bean concept in xml file. 8. Use all type of collection refences in spring xml file. 9. Minimize spring xml file using spring auto wire concept
- 1) Spring AOP 2) AOP Term 3) Write the Aspects 4) Configure Where the Aspects
- Practical Example : 10. Perfomr AOP(aspect oriented programming concept(login, perform, logout sequence)
- Spring ORM
- Practical Example : 11. Perform CRUD operation in spring web using hibernate integration

- 1) Spring MVC Web Forms 2) Spring Form Handling 3) Spring Form Tags 4) Spring Controller XML and Annotation Based
- Practical Example : 12. Create spring MVC pattern using dispatcher servlet. 13. CRUD operation using Spring MVC+ORM
- Spring MVC with Session Management
- Practical Example : 14. Spring MVC+ORM+Session