SYMBIOSIS SKILLS AND PROFESSIONAL UNIVERSITY

Certificate Course in Java and Advanced Java Programming Concepts

Program Structure

2021-22

Course Curriculum Pack

Program Name	**Certificate Course - Java and Advanced Java Programming Concepts**
Version No	**1.0** **Version Update date**
Pre-requisite	• **Knowledge of computer fundamentals**
Skills Students Acquire at end of the course	*Generic*: Communication skills (spoken, basic), Presentation Skills | *Technical*: Collate and Analyze Data, Use Java and Advanced Java for Programming.
Duration	3 Months (300 Hours)
Credits (Total)	15
Name of School	School of
Eligibility (Educational)	Graduate (BE All stream) MCA/ MSC (IT/CS) (BSC/ BCA Pass out 2021 or 2020 with minimum 50% from SSC to all semesters of graduation
Course Objective	• To learn why Java is useful for the design of desktop and web applications • To learn how to implement object-oriented designs with Java • To identify Java language components and how they work together in applications • To design and program stand-alone Java applications. • To learn how to design a graphical user interface (GUI) with Java Swing. • To understand how to use exception handling in Java applications • To understand how to design GUI components with the Java Swing API • To learn Java generics and how to use the JAVA collections API • To understand how to design applications with threads in Java. • To learn how to read and Write files in Java.
Course Learning Outcomes	After completing this programme, participants will be Be able to: • Codes basic programs in Java programming language. • Prints to the screen in Java language. • Makes relational operations in Java. • Constructs loops in Java. • Defines arrays in Java and uses them. • Uses objects and classes. • Declares objects and classes. • Distinguishes classes and objects.

	• Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs. • Read and make elementary modifications to Java programs that solve real-world problems. • Validate input in a Java program. • Identify and fix defects and common security issues in code. • Document a Java program using Javadoc. • Use a version control system to track source code in a project.
Intake Capacity	
Fees (if applicable)	

Credit Structure and weekly hour plan

Symbiosis Skills and Professional University										
School of Data Science										
Course Structure of Certificate Course - Java and Advanced Java Programming Concepts – 3 Months (12 Weeks)										
Sr.No.	**Module Code**	**Module Name**		**Hours**			**Credits**			
			Total Hours/ Week	**L**	**P**	**S**	**L**	**P**	**S**	**Total Credits**
1	JP101	Basic programing and Object Oriented Programing using Java	**4**	0	2	2	0	1	2	**3**
2	JP102	Java Destructor and Multithreading	**4**	0	2	2	0	1	2	**3**
3	JP103	Java Database Connectivity	**5**	0	2	3	0	1	3	**4**
4	JP104	Advance Java	**3**	0	0	3	0	0	3	**3**
5	JP105	JEE FULL STACK 2.0 WITH ANGULAR	**48/5**	0	2	3	0	1	3	**4**
6	JP106	Project	**2**	0	0	2	0	0	2	**2**
7	-	Continuous assessment	**2**	-	-	-	-	-	-	**-**
		Total	**25**	**0**	**6**	**12**	**0**	**3**	**12**	**19**
Total Credits: 19										

Total Hours (Weekly) : 26
12 weeks x 18 hrs/week = 468 hrs
26 weeks x 18 hrs/week = 468 hrs + 30 hrs for Examination and Evaluation = 498 hrs

Examination Scheme:

Module Code	Module Name	CAT	CAP	ESET	ESEP	SA	Total
JP101	Basic programing and Object Oriented Programing using Java	50	40	50	40	20	200
JP102	Java Destructor and Multithreading	50	40	50	40	20	200
JP103	Java Database Connectivity	50	40	50	40	20	200
JP104	Advance Java	50	0	50	0	20	120
JP105	JEE FULL STACK 2.0 WITH ANGULAR	50	0	50	0	20	120
JP105	Project	0	0	0	0	30	30
Total Mark							**840+30(Projet)= 870**

Evaluation / Grade criteria as per University norms

Instructional Activity Chart (mention what types of activities will be used to cover that LO)

Sr. No	Module/Unit	Learning Outcome	Theory / Lecture	Practical	Group Work	Lab Experiment (HW based)	Project Work	Seminar	Presentation	Research Work	Case Study Analysis	Group Discussions	Role Play	Prototype making	Other (Pl explain)
1	Getting Started	Lo1: Be able to Understand how to install and use a good Java development environment.													
		LO2: Understand Java programming basics	Y	Y	Y		Y		Y	Y	Y	Y			
		LO3: Begin using the Java programming language.	Y	Y	Y		Y		Y	Y	Y	Y			
		LO4: Learn basics of programming with a modern programming language, Java	Y	Y	Y		Y		Y	Y	Y	Y			
		LO5: Learn how to take a problem, figure out the algorithm to solve it, the write the code.													
		LO5: Understand Structure of java class													

		LO6: Be able to write Java programs													
2	Introduction to Class and Objects	LO6: learn Object-Oriented programming concepts and techniques using the Java programming language	Y	Y	Y		Y		Y	Y	Y	Y			
		LO7: Be able to identify classes, objects, members of a class and relationships among them needed for a specific problem	Y	Y	Y		Y		Y	Y	Y	Y			
3	Operators	LO8: Understand initialization of relational operators , logical operators, Arithmetic operators , Unary Operator, Ternary Operator , Assignment Operator	Y	Y	Y		Y		Y	Y	Y	Y			
		LO9: Understand how to evaluate relational operators, logical operators, Arithmetic operators, Unary Operator, Ternary Operator, and Assignment Operator.	Y	Y	Y		Y		Y	Y	Y	Y			
4	Conditional and Looping Statement's	LO10: Understand the concept of conditional looping	Y	Y	Y		Y		Y	Y	Y	Y	Y		
		LO11: Practical implementation of conditional and looping statements.													

		LO12: Determine the method of recursion	Y	Y	Y		Y		Y	Y	Y	Y	Y		
5	Arrays	LO13: Demonstrate how to initiate arrays													
		LO14: Understand the difference between a copy and an alias of an array													
		LO15: Understand initialization of add, update, read array elements													
6 &7	Object Oriented Programming	LO16: Understand the concept of Object Oriented Programming													
		LO17: Demonstrate the complete program using object-oriented programming concepts													
		LO18: Practical implementation of single and multilevel inheritance													
8	Abstract class and Abstract Methods	LO19: Be able to define, describe and correctly program classes and objects .													
		LO20: Be able to apply final variables, final methods and Final class													
9	Access Modifiers and Garbage Collection	LO21: A conceptual and practical implementation to the basic concepts and techniques of access modifiers													

		LO22: Demonstrate how to imports Static													
		LO23:To understand constructor chaining(with and without packages)													
		Lo24: Be able to understand JVM to run garbage collection.													
10 & 11	Wrapper Classes and String Class	LO25: Be able to importance of wrapper classes in Java													
		LO26: Practical demonstration on to sample classes to understand boxing & unboxing.													
12&13	Exception Handling	LO27:Be able to understand Exception Handling													
		LO28:Practical implementation to concept of Exception Handling													
		LO29: Understand the concept of classes, inheritance, inner classes, exception Handling and multi-threading to do parallel programming.													
14&15	java.io, java.nio and java.utils Package	LO30: Be able to describe java.io, java.nio and java.utils Package													

16,17 & 18	Collections	LO31: Be able to understand hierarchy in the Collections Framework of Java													
19 & 20	Multithreading & Synchronization	LO32: Objective of multithreading													
		LO33: Initialization understanding in synchronization necessary in multithreaded programming													
21,22 & 23	Database & SQL	LO34: Be able to understand basic SQL Syntax													
		LO35: Be able to write SQL commands to create tables and indexes, insert/update/delete data, and query data in a relational DBMS.													
		LO36: An ability to use and apply current technical concepts and practices in the Database & SQL													
24 & 25	J2EE Overview	LO37: learn the Internet Programming, using J2EE													
		LO36: Understand the JDBC architecture.													

		LO38: map Java classes and object associations to relational database tables													
		LO39: Apply the concepts of JDBC, Transaction processing, statement objects and Result set to perform operations on Database													
26	Architecture of Web	LO40: Understand how does internet works													
		LO41: Understand the development of web application architecture leading to a more modular approach													
		LO42: Be able to understand HTTP1.0, HTTP1.1 and HTTP2.0													
		LO43: An ability to use and apply Methods – GET, POST, HEAD, PUT, DELETE, etc													
27 & 28	HTML	LO44: learn the HTML programing													

		LO45:Be able to understand the concept of HTML tags and uses														
		LO46:Demostrate the features of HTML5														
		LO47: Be able to describe HTML Forms & Controls														
29 & 30	Cascading Style Sheets (CSS)	LO48: A conceptual and practical implementation of CSS														
		LO49: Practical implementation of CSS types														
31	Responsive Web Design	LO50: Identify the key functional elements of web pages														
		LO51: Use Bootstrap components to realize page designs														
		LO52: Be able to understand why users need to know where they are, where they can go and what is on a web page														
		LO53: Be able to Identify the key functional elements of web pages														
		LO54: Apply the concepts of various components.														

32 &33	JavaScript	LO55: Be able to describe and utilize Java script programming concepts such as variables, arrays, conditionals, and loops														
		LO56: Scripting or programming language that allows you to implement complex things on web pages														
34 & 35	JavaScript DOM	LO57:Learn about object model basics,														
		LO58: Utilizing JavaScript with HTML and CSS to create a web application.														
		LO59: Implement program logic using JavaScript.														
		LO60: Implement HTML5 APIs using JavaScript Capture user input using forms														
36,37 &38	Hibernate Framework	LO61: Understand about the Hibernate														
		LO62: Gain an understanding of the different ways of persistence, with a focus on the Hibernate Framework and its integration in Java applications.														

39, 40 & 41		LO63: Be able to understand comprehensive infrastructure support for developing Java applications.													
		LO64: Be able to understand an introduction to the spring framework where you will learn what spring is and its capabilities.													
42 & 43		LO65: Add advanced functionality to web applications using jQuery, AngularJS, Bootstrap, Type Script or Sass.													
		LO66: learn Spring framework offers hands-on experience building Spring Framework applications using Spring Boot.													
		LO67:Be able to create applications with Spring Boot, the modern way to create new spring													
44 & 45		LO68:Learn spring data module LO69: Learn the framework that sits on top of JPA and Hibernate and builds on both of these with Spring-centric functionality.													

46 & 47		LO70:Learning Spring data JPA with Spring Boot													
		LO71:Able to apply fundamentals of web services													
		LO72: Know and be able to describe building REST services with Spring													
48 & 49	Unit testing	LO73:Be able to improves the quality of the code													
		LO74:Learning to identifies every defect that may have come up before code is sent further for integration testing													
		LO75: Be able to writing tests before actual coding makes you think harder about the problem													
50	ES6 & Typescript	LO76: Understand the functions of arrow and default arguments													
		LO77:Be able to describe and utilize Typescript Fundamentals													
		LO78: Know and be able to determine when to use an interface or a class to define the structure of an object													

51	JEE Full Stack 2.0 with Angular **Agile SCRUM**	LO87:Learn Sprint 1 implementation with code reviews of L&D and BU trainer													
		LO88: Sprint 2 implementation with code reviews of L&D and BU trainer													
52	Core Java	LO89:Understand Declarations and Access Control													
		LO90:Learn Object Orientation													
		LO91:Learn operators													
		LO92: Learn Flow Control, Exceptions													
		LO93: Learn Gradle Fundamentals													
		LO94: Learn TDD with Junit 5													
		LO95: Learn Strings, I/O, Formatting, and Parsing													
		LO96: Learn Generics and Collections													
		LO97: Learn Threads													

		LO98: Understand Concurrent Patterns in Java														
		LO99: Understand Concurrent Collections														
		LO100: Understand Lambda expressions														
		LO101: Learn Stream API														
		LO102: Introduction to Design Pattern														
		LO103: Learn DevOps(Git,Sonarube,Maven,Jenkins)														
		LO104: **Database Using PostgreSQL**														
		LO105: **Introduction to JDBC Connection, Statement, PreparedStatement, ResultSet**														
		LO106:Understand JPA with Hibernate 3.0														
		LO107: Learn Spring 5.0														
		LO108: HTML 5, CSS 3 with Bootstrap,Javascript,TypeScript														
53	Angular	LO109:Introduction to Angular Framework														
		LO110:Essentials of Angular														
		LO111:Templates, Styles & Directives														

		LO112:Pipes,Services & Dependency Injection													
		LO113: Learn Template-Driven and Reactive Forms													
		LO114: Components Deep Dive / Routing													
		LO115: Http Requests / Observables													
		LO116:Understand Authentication and Route Protection													

Curriculum

Module/Unit	**Pre-requisite for module**	**Learning Outcome (LO to come in separate cells)**	**Sub Topics (There may be multiple topics to achieve one LO)**	**Instructional Activities (IAs should be mapped with sub topics as much)**	**Duration (hrs) (Hrs are as required for that IA)**
Getting	Java-Overview	Lo1: Be able to Understand how to install and use a good Java development environment.	• Setup development environment (JRE, JDK, eclipse) • Features of java • JVM Architecture	Demonstrate and lab activity how to write Java programs to:	
		LO2: Understand Java programming basics			

Started		LO3: Begin using the Java programming language.	• JDK and its usage • Structure of java class • Writing your first Java program • About main () method • Constructor in Java	• Print Hello World • Add two numbers/binary numbers/characters • Calculate compound interest • Calculate power of a number • Swap two numbers	
		LO4: Learn basics of programming with a modern programming language, Java			
		LO5: Learn how to take a problem, figure out the algorithm to solve it, the write the code.			
		LO5: Understand Structure of java class			
		LO6: Be able to write Java programs			
2	Introduction to Class and Objects	LO6: learn Object-Oriented programming concepts and techniques using the Java programming language	• Class & Object • Access Specifier • Java Data Types, Primitives and Binary Literals	Demonstrate and lab activity how to write Java programs to: • Calculate area of rectangle • Calculate area and circumference of circle using multiple classes • Java program to find ASCII value of a character	
		LO7: Be able to identify classes, objects, members of a class and relationships among them needed for a specific problem			
3	Operators	LO8: Understand initialization of relational operators , logical operators, Arithmetic operators , Unary Operator, Ternary Operator , Assignment Operator	• Arithmetic Operator • Relational Operator • Logical Operator • Unary Operator • Ternary Operator • Assignment Operator		
		LO9: Understand how to evaluate relational operators, logical operators, Arithmetic operators, Unary Operator, Ternary Operator, and Assignment Operator.			

4	Conditional and Looping Statement's	LO10: Understand the concept of conditional looping LO11: Practical implementation of conditional and looping statements. LO12: Determine the method of recursion	• If, else if, switch • break & continue keyword • for loop • while loop • do while loop • static & final keyword • Recursion	Demonstrate and lab activity how to write Java programs to: • Display prime numbers between 1 and 100 or 1 and n • Swap two variables without using the third variable • Find the factorial of a number • Check if a number is palindrome or not • Print Fibonacci series till n • Add two integer variables in 5 different ways using functions and control statement • Find square root of a number without sqrt method • Check Armstrong number • Calculate grades of students using their marks • Use switch case, recursion, print patterns, etc.	
		LO13: Demonstrate how to initiate arrays	• Initializing an Array in Java		

5	Arrays	LO14: Understand the difference between a copy and an alias of an array	• Two dimensional array in java • Java Variable Arguments explained • Add, update, read array elements • Sorting and searching in array • Java String Array to String • How to copy arrays in Java	Demonstrate and lab activity how to write Java programs to: • Calculate average of numbers using Array • Reverse an array • Sort an array in ascending order • Convert char Array to String • Add two Matrix using Multi-dimensional Arrays • Sort strings in alphabetical order • Find out the highest and second highest numbers in an array • Concatenate two arrays	
		LO15: Understand initialization of add, update, read array elements			
6 &7	Object Oriented Programming	LO16: Understand the concept of Object Oriented Programming	Introduction to OOP concepts Encapsulation Inheritance: single & multilevel Inheritance: Hierarchical Polymorphism: Compile time and runtime polymorphism Rules of overriding and overloading of methods super and this keywords Up casting & down casting of a reference variable	Demonstrate and lab activity to create a class Employee and encapsulate the data members. Demonstrate and lab activity create demo applications to illustrate different types of inheritance.	
		LO17: Demonstrate the complete program using object-oriented programming concepts			
		LO18: Practical implementation of single and multilevel inheritance			

8	Abstract class and Abstract Methods	LO19: Be able to define, describe and correctly program classes and objects .	Abstract class and abstract methods Interface (implementing multiple interfaces) Final variables, final methods and final class Functional interface New interface features(Java 8 & above) Lambda expression and stream API Arrays Enumerations	Demonstrate and lab activity to create an Array of Employee class and initialize array elements with different employee objects. Try to understand the no of objects on heap memory when any array is created.	
		LO20: Be able to apply final variables, final methods and Final class			
9	Access Modifiers and Garbage Collection	LO21: A conceptual and practical implementation to the basic concepts and techniques of access modifiers	Access modifiers(public, private, protected and default) Packages and import statements. Static imports Constructor chaining (with and without packages) Accessing protected variables and methods outside the package Garbage collection in java Requesting JVM to run garbage collection Different ways to make object eligible for garbage collection: (Nulling a reference variable, Re-assigning a reference variable & island of isolation) Finalize method	Demonstrate and lab activity to create a demo application to understand the role of access modifiers. Implement multilevel inheritance using different packages. Access/invoke protected members/methods of a class outside the package. Override finalize method to understand the behavior of JVM garbage collector.	
		LO22: Demonstrate how to imports Static			
		LO23:To understand constructor chaining(with and without packages)			
		Lo24: Be able to understand JVM to run garbage collection.			
10 & 11	Wrapper Classes and String Class	LO25: Be able to importance of wrapper classes in Java	Wrapper classes and constant pools String class, StringBuffer & StringBuilder class String pool	Demonstrate and lab activity to create sample	

		LO26: Practical demonstration on to sample classes to understand boxing & unboxing.		classes to understand boxing & unboxing. Use different methods of java defined wrapper classes. Create StringDemo class and perform different string manipulation methods	
12&13	Exception Handling	LO27:Be able to understand Exception Handling	Exception hierarchy, Errors, Checked and un-checked exceptions Exception propagation try-catch-finally block , throws clause and throw keyword Multi catch block Creating user defined checked and unchecked exceptions	Demonstrate and lab activity to create user defined checked and unchecked exceptions.	
		LO28:Practical implementation to concept of Exception Handling			
		LO29: Understand the concept of classes, inheritance, inner classes, exception Handling and multi-threading to do parallel programming.			
14&15	java.io, java.nio and java.utils Package	LO30: Be able to describe java.io, java.nio and java.utils Package	Brief introduction to InputStream, OutputStream, Reader and Writer interfaces NIO package Serialization and de-serialization Shallow copy and deep copy Object Class & java.util Package Date, DateTime, Calendar class Converting Date to String and String to Date using SimpleDateFormat class Object Class: Overriding to String, equals & hashcode method	Demonstrate and lab activity to create a Demo class to Read & write image/text files. Create Serialization Demo class to illustrate serialization and de-serialization process.	
16,17 & 18	Collections	LO31: Be able to understand hierarchy in the Collections Framework of Java	Introduction to collections: Collection hierarchy List, Queue, Set and Map Collections List Collection: • ArrayList, LinkedList	Demonstrate and lab activity to create Date Manipulator class to convert String to	

			• Vector (insert, delete,search,sort, iterate, replace operations) Collections class Comparable and Comparator interfaces Queue collection	date, date to String and to find out number of days between two dates. Demonstrate and lab activity to create a List of java defined wrapper classes and perform insert/delete/search /iterate/sort operations. Create a collection of Employee class and sort objects using comparable and comparator interfaces. Implement Queue data structure using LinkedList and Queue collection.	
			Set Collection: • HashSet, LinkedHashSet & TreeSet collection • Backed set collections Map Collection: • HashTable, HashMap, LinkedHashMap & TreeMap classes • Backed Map collections Generics Concurrent collections	Demonstrate and lab activity to create an Employee HashSet collection and override equals & hash Code methods to understand how the set maintains uniqueness using these methods. Create a Sample class to understand generic assignments using	

				“? extends SomeClass” , “? super someclass ” and “?”.	
19 & 20	Multithreading & Synchronization	LO32: Be able to understand objective of multithreading	MultiThreading : Thread class and Runnable Interface sleep, join, yield, setPriority, getPriority methods ThreadGroup class Synchronization Deadlock Wait, notify and notify All methods Inner classes	Invoke private methods of some other class using reflection. Create multiple threads using Thread class and Runnable interfaces. Assign same task and different task to multiple threads. Understand sleep, join, and yield methods.	
		LO33: Initialization understanding in synchronization necessary in multithreaded programming		Demonstrate and lab activity to create a Deadlock class to demonstrate deadlock in multithreading environment. Implement wait, notify and notify All methods. Demonstrate how to share threadlocal data between multiple threads. Create multiple threads using	

				anonymous inner classes.	
21,22 & 23	Database & SQL	LO34: Be able to understand basic SQL Syntax	Introduction to Relational Model Understanding Basic SQL Syntax SELECT, INSERT, UPDATE, DELETE Querying Data with the SELECT Statement The SELECT List SELECT List Wildcard (*) The FROM Clause How to Constrain the Result Set DISTINCT and NOT DISTINCT • Filtering Results with the Where Clause WHERE Clause Boolean Operators The AND Keyword The OR Keyword Other Boolean Operators BETWEEN, LIKE, IN, IS, IS NOT • Shaping Results with ORDER BY and GROUP BY ORDER BY Set Functions Set Function And Qualifiers GROUP BY HAVING clause • Matching Different Data Tables with JOINs CROSS JOIN INNER JOIN OUTER JOINs LEFT OUTER JOIN RIGHT OUTER JOIN FULL OUTER JOIN SELF JOIN • Creating Database Tables CREATE DATABASE CREATE TABLE NULL Values PRIMARY KEY CONSTRAINT ALTER TABLE DROP TABLE		
		LO35: Be able to write SQL commands to create tables and indexes, insert/update/delete data, and query data in a relational DBMS.			
		LO36: An ability to use and apply current technical concepts and practices in the Database & SQL An ability to use and apply current technical concepts and practices in the Database & SQL			
24 & 25	J2EE Overview	LO37: learn the Internet Programming, using J2EE	J2EE Overview • J2EE Container • Packaging Web applications • J2EE compliant web application • Deployment tools. • Web application life cycle • Deploying web applications. • Web Services Support JDBC & Transaction Management	Perform database CRUD operations using JDBC classes and interfaces.	
		LO36: Understand the JDBC architecture.			
		LO38: map Java classes and object associations to relational database tables			

		LO39: Apply the concepts of JDBC, Transaction processing, statement objects and Result set to perform operations on Database	• Introduction to JDBC API • JDBC Architecture • JDBC Drivers • JDBC Classes & Interfaces: Driver, Connection, Statement, Prepared Statement, Result Set • Stored procedures and functions Invocation • Design Pattern: Data Access Object Pattern		
26	Architecture of Web	LO40:Understand how does internet works	Brief history of the Internet How does the Internet work? Internet Protocol; HTTP Domain Names; Domain Name Service servers HTTP Protocols o Difference between HTTP1.0, HTTP 1.1, and HTTP 2.0 o Methods – GET, POST, HEAD, PUT, DELETE, etc. Status codes Stateless nature of the protocol and HTTP Session o HTTPS Architecture of the Web Web servers – IIS, Apache server	Exploring different browsers Mozilla Firefox, Google Chrome, Safari Exploring different text editors o Windows: Notepad++, Linux: Gedit or Vim or Emacs	
		LO41: Understand the development of web application architecture leading to a more modular approach			
		LO42: Be able to understand HTTP1.0, HTTP1.1 and HTTP2.0			
		LO43: An ability to use and apply Methods – GET, POST, HEAD, PUT, DELETE, etc			
27 & 28	HTML	LO44: learn the HTML programing	Introduction to HTML Document Object Model (DOM) Basic HTML Tags o Alignment, Headings, Anchor, Paragraph, Image, Lists, Tables, and iFrames HTML5 o New features in HTML5 o New elements, new attributes, link relations, microdata, ARIA accessibility o HTML5 Validation o Audio & Video Support HTML Forms & Controls o Input, Text Area, Radio Button, Checkbox, Dropdown, Submit, Reset, Button, etc.	Demonstrate and lab activity to create a HTML form for building a resume.	
		LO45:Be able to understand the concept of HTML tags and uses			
		LO46:Demostrate the features of HTML5			
		LO47: Be able to describe HTML Forms & Controls			
29 & 30		LO48: A conceptual and practical implementation of CSS	Introduction to CSS, Styling HTML with CSS, Structuring pages with CSS, Inline	Demonstrate and lab activity to	

	Cascading Style Sheets (CSS)		CSS, Internal CSS, External CSS, Multiple styles, CSS Fonts CSS Box Model id Attribute, class Attribute HTML Style Tags Linking a style to an HTML document	apply inline, internal and external CSS to change colors of certain text portions, bold, underline, and italics certain words in the previously created HTML resume form.	
		LO49: Practical implementation of CSS types			
31	Responsive Web Design	LO50: Identify the key functional elements of web pages	Introduction of UI Scripting The Best Experience for All Users o Desktop, Tablet, Mobile Bootstrap o Overview of Bootstrap, Need to use Bootstrap o Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid o Typography o Components – Tables, Images, Jumbotron, Wells, Alerts, Buttons, Button Groups, Badges/Labels, Progress Bars, Pagination, List Groups, Panels, Dropdowns, Collapse, Tabs/Pills, Navbar o Forms, Inputs o Bootstrap Themes, Templates Lab	Demonstrate and lab activity to update the design of the Resume form using Bootstrap	
		LO51: Use Bootstrap components to realize page designs			
		LO52: Be able to understand why users need to know where they are, where they can go and what is on a web page			
		LO53: Be able to Identify the key functional elements of web pages			
		LO54: Apply the concepts of various components.			
32 &33	JavaScript	LO55: Be able to describe and utilize Java script programming concepts such as variables, arrays, conditionals, and loops	Introduction to JavaScript Variables in JavaScript Statements, Operators, Comments, Expressions, and Control Structures JavaScript Scopes Strings, String Methods Numbers, Number Methods Boolean Values Dates, Date Formats, Date Methods Arrays, Array Methods	Demonstrate and lab activity to Practice writing basic JavaScript programs for better understanding of the language constructs	

		LO56: Scripting or programming language that allows you to implement complex things on web pages	Objects, Object Definitions, Object Properties, Object Methods, Object Prototypes Functions, Function Definitions, Function Parameters, Function Invocation, Function Closures Object Oriented Programming o Method, Constructor, Inheritance, Encapsulation, Abstraction, Polymorphism	Demonstrate and lab activity to write a JavaScript program to sort a list of elements by implementing a sorting algorithm. Demonstrate and lab activity to write a JavaScript program to list the properties of a JavaScript object.	
34 & 35	JavaScript DOM	LO57:Learn about object model basics,	Document Object Model (DOM) o Object hierarchy in JavaScript o HTML DOM, DOM Elements, DOM Events o DOM Methods, DOM Manipulation Forms, Forms API, Forms Validation Regular Expressions Errors, Debugging Introduction to Browser Dev Tool Pushing code quality via JSLint tool	Demonstrate and lab activity to write a JavaScript function to get First and Last name from the previously created Resume form Validate the entire Resume form using client-side JavaScript Demonstrate and lab activity to write a JavaScript function to validate whether a given value is RegEx or not.	
		LO58: Utilizing JavaScript with HTML and CSS to create a web application.			
		LO59: Implement program logic using JavaScript.			

		LO60: Implement HTML5 APIs using JavaScript Capture user input using forms			
36,37 &38	Hibernate Framework	LO61: Understand about the Hibernate	• Hibernate Framework • Introduction to Hibernate Framework Architecture • Hibernate in IDE o Creating web application using Hibernate API o Lifecycle of Hibernate Entities • HB with annotation example • Hibernate Mappings and Relationships • Collection and Component Mapping • HQL, Named Queries, Criteria Queries Lab	Demonstrate Hibernate as standalone library in Java application • Develop a web application (Online Bookshop) using Hibernate Persistence	
		LO62: Gain an understanding of the different ways of persistence, with a focus on the Hibernate Framework and its integration in Java applications.			
39, 40 & 41		LO63: Be able to understand comprehensive infrastructure support for developing Java applications.	• What is Spring Framework • Overview of Spring Architecture • Spring Modules Overview • Understanding Spring 4 annotations(Basic Introduction)	• Design and deploy Library Management System using Spring	
		LO64: Be able to understand an introduction to the spring framework where you will learn what spring is and its capabilities.	• What is IoC (Inversion of Control) • IOC container • Dependency Injection • Spring Beans and its lifecycle • Autowiring Beans • Configuring collections • Spring Validations • Spring i18n, Localization, Properties • File Upload example		
42 & 43		LO65: Add advanced functionality to web applications using jQuery, AngularJS, Bootstrap, Type Script or Sass.	• Spring Boot essentials • Why Spring boot • Spring Boot Overview • Basic Introduction of MAVEN	Demonstrate and lab activity to create Hello World Spring Boot Web	

			• Building Spring application with Boot • Spring Boot in detail (Use Spring Boot for all demo & assignments here onwards) • Running a web application using Spring Boot with CRUD (with Static Data not DB)	application • Check Libraries imported by Spring Boot • Create Spring Boot CRUD application	
		LO66: learn Spring framework offers hands-on experience building Spring Framework applications using Spring Boot.			
		LO67:Be able to create applications with Spring Boot, the modern way to create new spring			
44 & 45		LO68:Learn spring data module LO69: Learn the framework that sits on top of JPA and Hibernate and builds on both of these with Spring-centric functionality.	Spring Data Module • Spring Data JPA (Repository support for JPA) •CrudRepository&JPARepository • Query methods • Using custom query (@Query)	• Add CRUD operations with Spring JPA etc. to earlier Spring Web application.	
46 & 47		LO70:Learning Spring data JPA with Spring Boot	Building REST services with Spring • Introduction to web services • SOAP Vs RESTful web services • RESTful web service introduction • Create RESTful web service in java using Spring Boot • RESTful web service JSONexample • RESTful web service CRUD example • Using POSTMAN client to invoke REST API's • REST service invocation using REST Template	Demonstrate and lab activity to create REST API for Employee Management using Spring Boot • Invoke it from POSTMAN app • Invoke it from another Spring Boot Web application using REST Template	
		LO71:Able to apply fundamentals of web services			
		LO72: Know and be able to describe building REST services with Spring			

48 & 49	Unit testing	LO73:Be able to improves the quality of the code	• Introduction to unit testing • Introduction to Junit - Fix the annotations - Assert Exceptions - Run Tests • Introduction to Mockito - Create DAO and BO Layer - Adding Mockito Dependency - Stubbing and Setting Expectation - Result verification • Unit Testing of Spring Service Layer • Integration Testing of Spring Applications: REST API	Demonstrate and lab activity	
		LO74:Learning to identifies every defect that may have come up before code is sent further for integration testing			
		LO75: Be able to writing tests before actual coding makes you think harder about the problem			
50	ES6 & Typescript	LO76: Understand the functions of arrow and default arguments	- Var, Let and Const keyword - Arrow functions, default arguments - Template Strings, String methods - Object de-structuring - Spread and Rest operator - Typescript Fundamentals - Types & type assertions, Creating custom object types, function types - Typescript OOPS - Classes, Interfaces, Constructor, et	Demonstrate and lab activity	
		LO77:Be able to describe and utilize Typescript Fundamentals			
		LO78: Know and be able to determine when to use an interface or a class to define the structure of an object			
51	**JEE FULL STACK 2.0 WITH ANGULAR - (8 WEEKS)** Agile	LO79:Learn Sprint 1 implementation with code reviews of L&D and BU trainer	Implementing Spring into the project Test case reviews Code reviews Performance monitoring during the sprint implementation and sharing the feedback **Sprint – 1 Evaluation 30min/participant**	Demonstrate and lab activity	
		LO80: Sprint 2 implementation with code reviews of L&D and BU trainer	**Creating front end for the project using Angular** **Code reviews** **Performance monitoring during the sprint implementation and sharing the feedback** **Sprint - 2 Evaluation**		

			30min/participant		
52	Core Java	LO81:Understand **Declarations and Access Control**	**Declarations and Access Control** Identifiers & JavaBeans Legal Identifiers Sun's Java Code Conventions JavaBeans Standards Declare Classes Source File Declaration Rules Class Declarations and Modifiers Concrete Subclass Declaring an Interface Declaring Interface Constants Declare Class Members Access Modifiers Nonaccess Member Modifiers Constructor Declarations Variable Declarations Declaring Enums	Demonstrate and lab activity	
		LO82:Learn Object Orientation	• Object Orientation o Encapsulation o Inheritance, Is-A, Has-A o Polymorphism o Overridden Methods o Overloaded Methods o Reference Variable Casting o Implementing an Interface o Legal Return Types o Return Type Declarations o Returning a Value o Constructors and Instantiation o Default Constructor o Overloaded Constructors o Statics o Static Variables and Methods o Coupling and Cohesion	Demonstrate and lab activity	

		LO83:Learn Assignments and Operators	• Assignments o Stack and Heap—Quick Review o Literals, Assignments, and Variables o Literal Values for All Primitive Types o Assignment Operators o Casting Primitives o Using a Variable or Array Element That Is Uninitialized and Unassigned o Local (Stack, Automatic) Primitives and Objects o Passing Variables into Methods o Passing Object Reference Variables o Does Java Use Pass-By-Value Semantics? o Passing Primitive Variables o Array Declaration, Construction, and Initialization o Declaring an Array o Constructing an Array o Initializing an Array o Initialization Blocks o Using Wrapper Classes and Boxing o An Overview of the Wrapper Classes o Creating Wrapper Objects o Using Wrapper Conversion Utilities o Autoboxing o Overloading o Garbage Collection o Overview of Memory Management and Garbage Collection o Overview of Java's Garbage Collector o Writing Code That Explicitly Makes Objects Eligible for Garbage Collection • Operators o Java Operators o Assignment Operators	Demonstrate and lab activity	

			o Relational Operators o instanceof Comparison o Arithmetic Operators o Conditional Operator o Logical Operators		
		LO84: Learn Flow Control, Exceptions	• Flow Control, Exceptions o if and switch Statements o if-else Branching o switch Statements o Loops and Iterators o Using while Loops o Using do Loops o Using for Loops o Using break and continue o Unlabeled Statements o Labeled Statements o Handling Exceptions o Catching an Exception Using try and catch o Using finally o Propagating Uncaught Exceptions o Defining Exceptions o Exception Hierarchy o Handling an Entire Class Hierarchy of Exceptions o Exception Matching o Exception Declaration and the Public Interface o Rethrowing the Same Exception o Common Exceptions and Errors	Demonstrate and lab activity	
		LO85: Learn Gradle Fundamentals	• Gradle Fundamentals o Introduction o Folder Structure o Install and Setup Gradle on Windows o Dependencies in Build Scripts		

			o Gradle Wrapper o Lifecycle Tasks: The Base Plug In o Using Project Info and the check command o Creating Variables and external properties o Creating a Build Scan o Dependencies		
		LO86: Learn TDD with Junit 5	• TDD with Junit 5 o Types of Tests o Why Unit Tests Are Important o What's JUnit? o JUnit 5 Architecture o IDEs and Build Tool Support o Setting up JUnit with Maven o Lifecycle Methods o Test Hierarchies o Assertions o Disabling Tests o Assumptions o Test Interfaces and Default Methods o Repeating Tests o Dynamic Tests o Parameterized Tests o Argument Sources o Argument Conversion o What Is TDD? o History of TDD o Why Practice TDD? o Types of Testing o Testing Frameworks and Tools o Testing Concepts o Insights from Testing o Mocking Concepts o Mockito Overview o Mockito Demo	Demonstrate and lab activity	

			o Creating Mock Instances o Stubbing Method Calls		
		LO87: Learn Strings, I/O, Formatting, and Parsing	• Strings, I/O, Formatting, and Parsing o String, StringBuilder, and StringBuffer o The String Class o Important Facts About Strings and Memory o Important Methods in the String Class o The StringBuffer and StringBuilder Classes o Important Methods in the StringBuffer and StringBuilder Classes o File Navigation and I/O o Types of Streams o The Byte-stream I/O hierarchy o Character Stream Hierarchy o RandomAccessFile class o The java.io.Console Class o Serialization o Dates, Numbers, and Currency o Working with Dates, Numbers, and Currencies o Parsing, Tokenizing, and Formatting o Locating Data via Pattern Matching o Tokenizing	Demonstrate and lab activity	
		LO88: Learn Generics and Collections	• Generics and Collections o Overriding hashCode() and equals() o Overriding equals() o Overriding hashCode() o Collections o So What Do You Do with a Collection? o List Interface o Set Interface	Demonstrate and lab activity	

			o Map Interface o Queue Interface o Using the Collections Framework o ArrayList Basics o Autoboxing with Collections o Sorting Collections and Arrays o Navigating (Searching) TreeSets and TreeMaps o Other Navigation Methods o Backed Collections o Generic Types o Generics and Legacy Code o Mixing Generic and Non-generic Collections o Polymorphism and Generics		
		LO89: Learn Threads	Threads Defining, Instantiating, and Starting Threads Defining a Thread Instantiating a Thread Starting a Thread Thread States and Transitions Thread States Preventing Thread Execution Sleeping Thread Priorities and yield( ) Synchronizing Code Synchronization and Locks Thread Deadlock Thread Interaction Using notifyAll( ) When Many Threads May Be Waiting	Demonstrate and lab activity	
		LO90: Understand Concurrent Patterns in Java	• Concurrent Patterns in Java	Demonstrate and lab activity	

			o Introducing Executors, What Is Wrong with the Runnable Pattern? o Defining the Executor Pattern: A New Pattern to Launch Threads o Defining the Executor Service Pattern, a First Simple Example o Comparing the Runnable and the Executor Service Patterns o Understanding the Waiting Queue of the Executor Service o Wrapping-up the Executor Service Pattern o From Runnable to Callable: What Is Wrong with Runnables? o Defining a New Model for Tasks That Return Objects o Introducing the Callable Interface to Model Tasks o Introducing the Future Object to Transmit Objects Between Threads o Wrapping-up Callables and Futures, Handling Exceptions		
		LO91: Understand Concurrent Collections	• Concurrent Collections o Implementing Concurrency at the API Level o Hierarchy of Collection and Map, Concurrent Interfaces o What Does It Mean for an Interface to Be Concurrent? o Why You Should Avoid Vectors and Stacks o Understanding Copy On Write Arrays o Introducing Queue and Deque, and Their Implementations	Demonstrate and lab activity	

			o Understanding How Queue Works in a Concurrent Environment o Adding Elements to a Queue That Is Full: How Can It Fail? o Understanding Error Handling in Queue and Deque o Introducing Concurrent Maps and Their Implementations o Atomic Operations Defined by the ConcurrentMap Interface o Understanding Concurrency for a HashMap o Understanding the Structure of the ConcurrentHashMap from Java 7 o Introducing the Java 8 ConcurrentHashMap and Its Parallel Methods o Parallel Search on a Java 8 ConcurrentHashMap o Parallel Map / Reduce on a Java 8 ConcurrentHashMap o Parallel ForEach on a Java 8 ConcurrentHashMap o Creating a Concurrent Set on a Java 8 ConcurrentHashMap o Introducing Skip Lists to Implement ConcurrentMap o Understanding How Linked Lists Can Be Improved by Skip Lists o How to Make a Skip List Concurrent Without Synchronization		
		LO92: Understand Lambda expressions	• Lambda Expressions o Introduction o Writing Lambda Expressions o Functional Interfaces o Types of Functional Interfaces o Method reference	Demonstrate and lab activity	

		LO93: Learn Stream API	• Stream API o Introduction o Stream API with Collections o Stream Operations	Demonstrate and lab activity	
		LO94: Introduction to Design Pattern	Introduction to Design Pattern Self learning with online links and explanation by Trainer with Demos o Creational Design Pattern ▯ Factory Pattern ▯ Singleton Pattern ▯ Prototype Pattern o Structural Design Pattern ▯ Decorator Pattern ▯ Facade Pattern o Behavioral Design Pattern ▯ Chain of Responsibility Pattern ▯ Iterator Pattern o Presentation Layer Design Pattern ▯ Intercepting Filter Pattern ▯ Front Controller Pattern o Business Layer Design Pattern ▯ Business Delegate Pattern ▯ Transfer Object Pattern o Integration Layer Design Pattern ▯ Data Access Object Pattern	Demonstrate and lab activity	
		LO95: Learn DevOps(Git,Sonarube,Maven,Jenkins)	• DevOps (Git, SonarQube, Maven, Jenkins) • Introduction to DevOps o Introduction of DevOps o Dev And Ops o Agile Vs DevOps o Continuous Integration & Delivery pipeline o Tools For DevOps	Demonstrate and lab activity	

			o Use-case walkthrough • GIT Hub o Working locally with GIT o Working remotely with GIT o Branching, merging & rebasing with GIT o Use Case walkthrough • Jenkins: o Introduction to Jenkins o Jenkins Objective o Introduction to continuous integration deployment & Jenkins-ci o Continuous Deployment & distribution builds with Jenkins • Sonar o Introduction to Sonar o Code quality Monitoring- Sonar o Use Case walkthrough		
		LO96: **Database Using PostgreSQL**	Database Using PostgreSQL Duration : 2 days Contents: • Introduction o The Relational Model o What is PostgreSQL? o PostgreSQL – Data Types o Arrays Functions and Operators • Understanding Basic PostgreSQL Syntax o The Relational Model o Basic SQL Commands - SELECT o Basic SQL Commands - INSERT o Basic SQL Commands - UPDATE o Basic SQL Commands – DELETE	Demonstrate and lab activity	

			• Querying Data with the SELECT Statement o Wildcards (%, _) o The SELECT List o SELECT List Wildcard (*) o The FROM Clause o How to Constrain the Result Set o DISTINCT and NOT DISTINCT • Arrays Functions and Operators o array_append o array_cat o array_lower o array_to_string o array_agg o every,Count,sum,avg o Array Operators • Filtering Results with the Where Clause o WHERE Clause o Boolean Operators o The AND Keyword o The OR Keyword o Other Boolean Operators BETWEEN, LIKE, IN, IS, IS NOT • Shaping Results with ORDER BY and GROUP BY o ORDER BY o Set Functions o Set Function And Qualifiers o GROUP BY o HAVING clause • Matching Different Data Tables with JOINs o Table Aliases		

			o CROSS JOIN o INNER JOIN o OUTER JOINs o LEFT OUTER JOIN o RIGHT OUTER JOIN o FULL OUTER JOIN o SELF JOIN o Natural Join • Creating Database Tables o CREATE DATABASE o CREATE TABLE o NULL Values o PRIMARY KEY o CONSTRAINT o ALTER TABLE o DROP TABLE • PostgreSQL Transactions o BEGIN, COMMIT, ROLLBACK • PostgreSQL Constraints o CHECK, UNIQUE, NOT NULL Introduction to JDBC o Connection, Statement, PreparedStatement, ResultSet		
		LO97: **Introduction to JDBC Connection, Statement, PreparedStatement, ResultSet**	Introduction - Introduction & overview of data persistence - Overview of ORM tools - Understanding JPA - JPA Specifications Entities	Demonstrate and lab activity	

			- Requirements for Entity Classes - Persistent Fields and Properties in Entity Classes - Persistent Fields - Persistent Properties - Using Collections in Entity Fields and Properties - Validating Persistent Fields and Properties - Primary Keys in Entities - Managing Entities - The EntityManager Interface - Container-Managed Entity Managers - Application-Managed Entity Managers - Finding Entities Using the EntityManager - Managing an Entity Instance's Lifecycle - Persisting Entity Instances - Removing Entity Instances - Synchronizing Entity Data to the Database - Persistence Units - Querying Entities - Java Persistence query language (JPQL) - Criteria API - Entity Relationships - Direction in Entity Relationships - Bidirectional Relationships - Unidirectional Relationships - Queries and Relationship Direction - Cascade Operations and Relationships		

		LO98:Understand JPA with Hibernate 3.0	Introduction - Introduction & overview of data persistence - Overview of ORM tools - Understanding JPA - JPA Specifications Entities - Requirements for Entity Classes - Persistent Fields and Properties in Entity Classes - Persistent Fields - Persistent Properties - Using Collections in Entity Fields and Properties - Validating Persistent Fields and Properties - Primary Keys in Entities Managing Entities - The EntityManager Interface - Container-Managed Entity Managers - Application-Managed Entity Managers - Finding Entities Using the EntityManager - Managing an Entity Instance's Lifecycle - Persisting Entity Instances - Removing Entity Instances - Synchronizing Entity Data to the Database - Persistence Units Querying Entities - Java Persistence query language (JPQL) - Criteria API Entity Relationships	Demonstrate and lab activity	

			- Direction in Entity Relationships - Bidirectional Relationships - Unidirectional Relationships - Queries and Relationship Direction - Cascade Operations and Relationships		
		LO99: Learn Spring 5.0	1. Spring Core Spring Core Introduction / Overview - Shortcomings of Java EE and the Need for Loose Coupling - Managing Beans, The Spring Container, Inversion of Control - The Factory Pattern - Configuration Metadata - XML, @Component, Auto-Detecting Beans - Dependencies and Dependency Injection (DI) with the BeanFactory - Setter Injection Spring Container - The Spring Managed Bean Lifecycle - Autowiring Dependencies Dependency Injection - Using the Application Context - Constructor Injection - Factory Methods - Crucial Namespaces ‘p’ and ’c’ - Configuring Collections Metadata / Configuration - Annotation Configuration @Autowired, @Required, @Resource - @Component, Component Scans. Component Filters - Life Cycle Annotations	Demonstrate and lab activity	

			- Java Configuration, @Configuration, XML free configuration - The Annotation Config Application Context 2. Spring Boot SPRING BOOT Introduction - Spring Boot starters, CLI, Gradle plugin - Application class - @SpringBootApplication - Dependency injection, component scans, Configuration - Externalize your configuration using application.properties - Context Root and Management ports - Logging Using Spring Boot - Build Systems, Structuring Your Code, Configuration, Spring Beans and Dependency Injection, and more. Spring Boot Essentials - Application Development, Configuration, Embedded Servers, Data Access, and many more - Common application properties - Auto-configuration classes - Spring Boot Dependencies 3. Spring Data JPA - Spring Data JPA Intro & Overview - Core Concepts, @RepositoryRestResource - Defining Query methods - Query Creation - Using JPA Named Queries		

			- Defining Repository Interfaces - Creating Repository instances - JPA Repositories - Persisting Entities - Transactions 4. Spring Data REST - Introduction & Overview - Adding Spring Data REST to a Spring Boot Project - Configuring Spring Data REST - Repository resources, Default Status Codes, Http methods - Spring Data REST Associations - Define Query methods 5. Introduction to Spring Security with Demo 6. Introduction to Spring Microservices with Demo		
		LO100: HTML 5, CSS 3 with Bootstrap,Javascript,TypeScript	HTML 5: • HTML Basics o Understand the structure of an HTML page. o New Semantic Elements in HTML 5 o Learn to apply physical/logical character effects. o Learn to manage document spacing. • Tables o Understand the structure of an HTML table. o Learn to control table format like cell spanning, cell spacing, border • List o Numbered List o Bulleted List • Working with Links	Demonstrate and lab activity	

			o Understand the working of hyperlinks in web pages. o Learn to create hyperlinks in web pages. o Add hyperlinks to list items and table contents. • Image Handling o Understand the role of images in web pages o Learn to add images to web pages o Learn to use images as hyperlinks • Frames o Understand the need for frames in web pages. o Learn to create and work with frames. • HTML Forms for User Input o Understand the role of forms in web pages o Understand various HTML elements used in forms. o Single line text field o Text area o Check box o Radio buttons o Password fields o Pull-down menus o File selector dialog box • New Form Elements o Understand the new HTML form elements such as date, number, range, email, search and datalist o Understand audio, video, article tags CSS 3 □ Introduction to Cascading Style Sheets 3.0 - What CSS can do - CSS Syntax		

			- Types of CSS - Working with Text and Fonts - Text Formatting - Text Effects - Fonts - CSS Selectors - Type Selector - Universal Selector - ID Selector o Class selector - Colors and Borders - Background - Multiple Background - Colors RGB and RGBA - HSL and HSLA - Borders - Rounded Corners - Applying Shadows in border - Implementing CSS3 in the "Real World" o Modernizr o HTML5 Shims o SASS, and Other CSS Preprocessors o CSS Grid Systems o CSS Frameworks		
		LO101: Learn Bootstrap	- Introduction to Bootstrap - Introduction - Getting Started with Bootstrap - Bootstrap Basics - Bootstrap grid system - Bootstrap Basic Components - Bootstrap Components - Page Header - Breadcrumb - Button Groups	Demonstrate and lab activity	

			- Dropdown - Nav & Navbars □ JavaScript Essentials □ ES6 & Typescript - Var, Let and Const keyword - Arrow functions, default arguments - Template Strings, String methods - Object de-structuring - Spread and Rest operator - Typescript Fundamentals - Types & type assertions, Creating custom object types, function types - Typescript OOPS - Classes, Interfaces, Constructor, etc		
53	Angular	LO102:Introduction to Angular Framework	□ Introduction to Angular Framework - Introduction to Angular Framework, History & Overview - Environment Setup, Angular CLI, Installing Angular CLI - NPM commands & package.json - Bootstrapping Angular App, Components, AppModule - Project Setup, Editor Environments - First Angular App & Directory Structure - Angular Fundamentals, Building Blocks - MetaData □ Essentials of Angular - Component Basics - Setting up the templates - Creating Components using CLI - Nesting Components - Data Binding - Property & Event Binding, String Interpolation, Style binding	Demonstrate and lab activity	

			- Two-way data binding - Input Properties, Output Properties, Passing Event Data □ Templates, Styles & Directives - Template, Styles, View Encapsulation, adding bootstrap to angular app - Built-in Directives, Creating Attribute Directive - Using Renderer to build attribute directive - Host Listener to listen to Host Events - Using Host Binding to bind to Host Properties □ Pipes, Services & Dependency Injection - In-built Pipes, Creating a Custom Pipes - Services & Dependency Injections - Creating Data Service - Understanding Hierarchical Injector □ Template-Driven and Reactive Forms - Template-Driven vs Reactive Approach - Understanding Form State - Built-in Validators & Using HTML5 Validation - Grouping Form Controls - FormGroup, FormControl, FormBuilder - Forms with Reactive Approach - Predefined Validators & Custom Validators - Showing validation errors		

			Components Deep Dive / Routing - Component Life Cycle Hooks - Reusable components in angular using <ng-content> - Navigating with Router links - Understanding Navigation Paths - Navigating Programmatically - Passing Parameters to Routes - Passing Query Parameters and Fragments - Setting up Child (Nested) Routes - Outsourcing Route Configuration (create custom module) Http Requests / Observables - HTTP Requests - Sending GET Requests - Sending a PUT Request - Using the Returned Data - Catching Http Errors - Basics of Observables & Promises		
		LO103:Essentials of Angular	Introduction to Angular Framework - Introduction to Angular Framework, History & Overview - Environment Setup, Angular CLI, Installing Angular CLI - NPM commands & package.json - Bootstrapping Angular App, Components, AppModule - Project Setup, Editor Environments - First Angular App & Directory Structure - Angular Fundamentals, Building Blocks - MetaData Essentials of Angular - Component Basics		
		LO111:Templates, Styles & Directives			
		LO104:Pipes,Services & Dependency Injection			
		LO105: Learn Template-Driven and Reactive Forms			
		LO106: Components Deep Dive / Routing			
		LO107: Http Requests / Observables			
		LO108:Understand Authentication and Route Protection			

			- Setting up the templates - Creating Components using CLI - Nesting Components - Data Binding - Property & Event Binding, String Interpolation, Style binding - Two-way data binding - Input Properties, Output Properties, Passing Event Data □ Templates, Styles & Directives - Template, Styles, View Encapsulation, adding bootstrap to angular app - Built-in Directives, Creating Attribute Directive - Using Renderer to build attribute directive - Host Listener to listen to Host Events - Using Host Binding to bind to Host Properties □ Pipes, Services & Dependency Injection - In-built Pipes, Creating a Custom Pipes - Services & Dependency Injections - Creating Data Service - Understanding Hierarchical Injector □ Template-Driven and Reactive Forms - Template-Driven vs Reactive Approach - Understanding Form State - Built-in Validators & Using HTML5 Validation - Grouping Form Controls		

			- FormGroup, FormControl, FormBuilder - Forms with Reactive Approach - Predefined Validators & Custom Validators - Showing validation errors - Components Deep Dive / Routing - Component Life Cycle Hooks - Reusable components in angular using <ng-content> - Navigating with Router links - Understanding Navigation Paths - Navigating Programmatically - Passing Parameters to Routes - Passing Query Parameters and Fragments - Setting up Child (Nested) Routes - Outsourcing Route Configuration (create custom module) - Http Requests / Observables - HTTP Requests - Sending GET Requests - Sending a PUT Request - Using the Returned Data - Catching Http Errors - Basics of Observables & Promises		
54	**Project**				
		Project	Student will independently work on the project	Well-equipped computer lab with projector, internet connection	2
		Training 318 (60 + 60 + 75 + 45 + 30+48) + 30 Examination = 348 Hours Total			

Teaching Plan and Resource Requirement

Day	Instructional Activity (conducted on day)	Description (Actual activity details)	Equipment / HW / SW / Consum Be able / Resource	Hours
1	Demonstrate and lab activity how to write Java programs to: • To Print Hello World • Add two numbers/binary numbers/characters • Calculate compound interest • Calculate power of a number • Swap two numbers	Give the demonstration and lab activity on how to write Java Programs 1. Print the world, Addition of two numbers/characters , Calculate compound interest	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	2
2	Demonstrate and lab activity how to write Java programs to: • Calculate area of rectangle • Calculate area and circumference of circle using multiple classes • Java program to find ASCII value of a character	Give the demonstration and lab activity on how to write Java Programs	Well-equipped computer lab with projector, internet connection, Computer, MS Office using (PPT)	2
3	Demonstrate and lab activity how to write Java programs to: • Display prime numbers between 1 and 100 or 1 and n • Swap two variables without using the third variable • Find the factorial of a number • Check if a number is palindrome or not • Print Fibonacci series till n • Add two integer variables in 5 different ways using functions and control statement	Give the demonstration and lab activity on how to write Java Programs	Computer, MS Office using (PPT). Well-equipped computer lab with projector, internet connection.	**2**

	• Find square root of a number without sqrt method • Check Armstrong number • Calculate grades of students using their marks • Use switch case, recursion, print patterns, etc.			
4	Demonstrate and lab activity how to write Java programs to: • Calculate average of numbers using Array • Reverse an array • Sort an array in ascending order • Convert char Array to String • Add two Matrix using Multi-dimensional arrays • Sort strings in alphabetical order • Find out the highest and second highest numbers in an array • Concatenate two arrays	Give demonstration on Java programs • Calculate average of numbers using Array • Reverse an array • Sort an array in ascending order • Convert char Array to String • Add two Matrix using Multi-dimensional arrays • Sort strings in alphabetical order • Find out the highest and second highest numbers in an array • Concatenate two arrays	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
5	Demonstrate and lab activity to create: • A class Employee and encapsulate the data members. • Create demo applications to illustrate different types of inheritance.	Demonstrate how to use inheritance Assign lab activity.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
6	Demonstrate and lab activity to create • An Array of Employee class and initialize array elements with different employee objects.	Demonstrate and lab activity how to create arrays Perform different operations to arrays	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
7	Demonstrate and lab activity to: • Create a demo application to understand the role of access modifiers. Implement multilevel inheritance using different packages. Access/invoke protected members/methods of a class outside the package. Override finalize method	Explain the role of modifiers. Assign lab activity.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**

	to understand the behavior of JVM garbage collector.			
8	Demonstrate and lab activity to create sample classes to understand boxing & unboxing. Use different methods of java defined wrapper classes. Create StringDemo class and perform different string manipulation methods	Demonstrate and lab activity how to create set and dictionary. Perform different operations on it. Assign lab activity.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
9	Demonstrate and lab activity to create user defined checked and unchecked exceptions.	Give the demonstration how to create User-defined exceptions. Assign lab activity.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
10	Demonstrate and lab activity to create • A demo class to Read & write image/text files. • Create Serialization Demo class to illustrate serialization and de-serialization process.	Assign lab activity.	Well-equipped computer lab with projector, internet connection.	**2**
11	Demonstrate and lab activity to create • Date Manipulator class to convert String to date, date to String and to find out number of days between two dates. • A List of java defined wrapper classes and perform insert/delete/search/iterate/sort operations. • A collection of Employee class and sort objects using comparable and comparator interfaces. Implement Queue data structure using LinkedList and Queue collection.	Demonstrate and lab activity how to create a classes using Java. Perform different operations on it. Ask students to do the same.	Well-equipped computer lab with projector, internet connection.	**2**
12	Demonstrate and lab activity how to write Java programs: • Calculate average of numbers using Array • Reverse an array • Sort an array in ascending order • Convert char Array to String • Add two Matrix using Multi-dimensional Arrays	Demonstrate and lab activity how to create Multi-dimensional Arrays. Ask students to do the same.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**

	• Sort strings in alphabetical order • Find out the highest and second highest numbers in an array • Concatenate two arrays			
13	Demonstrate and lab activity to create • A Demo class to Read & write image/text files. • Create Serialization Demo class to illustrate serialization and de-serialization process.	Demonstrate Serialization and Deserialization in Java with Example Assign Lab activity to students.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
14	Demonstrate and lab activity to create • Date Manipulator class to convert String to date, date to String and to find out number of days between two dates. • List of java defined wrapper classes and perform insert/delete/search/iterate/sort operations. • Create a collection of Employee class and sort objects using comparable and comparator interfaces. Implement Queue data structure using LinkedList and Queue collection.	Ask students to solve the questions given in the activity on classes Assign Lab activity to students.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
15	Demonstrate and lab activity to create • A class Employee and encapsulate the data members. • Demo applications to illustrate different types of inheritance.	Demonstrate java inheritance. Ask students to solve the questions given in the activity.	Well-equipped computer lab with projector, internet connection.	**2**
16	Demonstrate and lab activity to create • An Array of Employee class and initialize array elements with different employee objects.	Demonstrate how to Create Array of Objects in Java. Assign Lab activity to students.	Well-equipped computer lab with projector, internet connection.	**2**
17	Demonstrate and lab activity to create • A demo application to understand the role of access modifiers. Implement multilevel inheritance using different packages. Access/invoke protected members/methods	Demonstrate java multilevel inheritance. Ask students to solve the questions given in the activity.	Well-equipped computer lab with projector, internet connection.	**2**

	of a class outside the package. Override finalize method to understand the behavior of JVM garbage collector.			
18	Demonstrate and lab activity to create • Sample classes to understand boxing & unboxing. Use different methods of java defined wrapper classes. Create StringDemo class and perform different string manipulation methods	Demonstrate and lab activity wrapper classes for primitive data types Assign lab activity to students.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
19	Demonstrate and lab activity to create user defined checked and unchecked exceptions.	Demonstrate and lab activity how to create user defined checked and unchecked exceptions Ask students to work on the plotting activity	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
20	Demonstrate and lab activity to create a Demo class to Read & write image/text files. Create Serialization Demo class to illustrate serialization and de-serialization process.	Demonstrate and lab activity how create files. Ask students to work on the files	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
21	Demonstrate and lab activity to create Date Manipulator class to convert String to date, date to String and to find out number of days between two dates. • A List of java defined wrapper classes and perform insert/delete/search/iterate/sort operations. Create a collection of Employee class and sort objects using comparable and comparator interfaces. Implement Queue data structure using Linked List and Queue collection.	Demonstrate and lab activity on conversions. • Class to convert • String to date • Date to String Assign programs to students for performing the same.	Well-equipped computer lab with projector, internet connection. Computer.	**2**
22	Demonstrate and lab activity how to write Java programs to: • Calculate average of numbers using Array • Reverse an array • Sort an array in ascending order • Convert char Array to String • Add two Matrix using Multi-dimensional Arrays • Sort strings in alphabetical order	Explain how to perform arrays in a program. Demonstrate and lab activity on functions of arrays	Well-equipped computer lab with projector, internet connection. Computer.	**2**

	• Find out the highest and second highest numbers in an array • Concatenate two arrays			
23	Demonstrate and lab activity to create a class Employee and encapsulate the data members. • Create demo applications to illustrate different types of inheritance.	Perform on how to create class and inheritance. Demonstrate and lab activity on classes using Java.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
24	Demonstrate and lab activity to create • An Array of Employee class and initialize array elements with different employee objects.	Demonstrate and lab activity the process of arrays.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
25	Demonstrate and lab activity to create • A demo application to understand the role of access modifiers. Implement multilevel inheritance using different packages. Access/invoke protected members/methods of a class outside the package. Override finalize method to understand the behavior of JVM garbage collector.	Demonstrate and lab activity to create inheritance Excute java program, JVM creates three threads. 1) main thread 2) Thread Scheduler 3) Garbage Collector Thread. Assign programs to use define finalize() method	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
26	Demonstrate and lab activity to create Date Manipulator class to convert String to date, date to String and to find out number of days between two dates. • A List of java defined wrapper classes and perform insert/delete/search/iterate/sort operations. Create a collection of Employee class and sort objects using comparable and comparator interfaces. Implement Queue data structure using LinkedList and Queue collection.	Mechanism *to convert primitive into object and object into primitive*. To execute a program using class and object	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**

27	Demonstrate and lab activity to create • An Employee HashSet collection and override equals & hash Code methods to understand how the set maintains uniqueness using these methods. Create a Sample class to understand generic assignments using "? extends SomeClass" , "? super someclass " and "?".	Demonstrate and lab activity	Well-equipped computer lab with projector, internet connection. Computer	**2**
28	Invoke private methods of some other class using reflection. Create multiple threads using Thread class and Runnable interfaces. Assign same task and different task to multiple threads. Understand sleep, join, and yield methods.	Demonstrate and lab activity handling dataset using multiple threads.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
29	Perform database CRUD operations using JDBC classes and interfaces.	Learn basic database operations (CRUD - Create, Retrieve, Update and Delete) using JDBC (Java Database Connectivity) API.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**2**
30	Exploring different browsers Mozilla Firefox, Google Chrome, Safari Exploring different text editors o Windows: Notepad++, Linux: Gedit or Vim or Emacs	Assignment	-	**2**
31	Demonstrate and lab activity to create a HTML form for building a resume.	Execute HTML Program using html tags to building a resume	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**
32	Demonstrate and lab activity to apply inline, internal and external CSS to change colors of certain text portions, bold, underline, and italics certain words in the previously created HTML resume form.	Create webpages using concepts of CSS(Inline, External and Internal)	Well-equipped computer lab with projector, internet connection.	**3**
33	Demonstrate and lab activity to update the design of the Resume form using Bootstrap	Demonstrate and lab activity to develop a single page HTML resume using Bootstrap	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**

34	Demonstrate and lab activity to Practice writing basic JavaScript programs for better understanding of the language constructs	Create a web page using dynamically updating content, control multimedia, animate images. Using JavaScript	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**
35	Demonstrate and lab activity to: Write a JavaScript program to sort a list of elements by implementing a sorting algorithm. Write a JavaScript program to list the properties of a JavaScript object.	Demonstrate and lab activity the following and ask students to perform: • A list of elements by implementing a sorting algorithm • List the properties of a JavaScript object.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**
36	Demonstrate and lab activity to: Write a JavaScript function to get First and Last name from the previously created Resume form Validate the entire Resume form using client-side JavaScript Write a JavaScript function to validate whether a given value is RegEx or not.	Demonstrate and lab activity the following and ask students to perform: • Validation of program using client-side JavaScript • To validate whether a given value is RegEx or not.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**
37	Demonstrate Hibernate as standalone library in Java application • Develop a web application (Online Bookshop) using Hibernate Persistence	Demonstrate and lab activity the following and ask students to perform to Develop a web application (Online Bookshop) using Hibernate Persistence	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**
38	Design and deploy Library Management System using Spring	Assignment	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**
39	Demonstrate and lab activity to create Hello World Spring Boot Web application • Check Libraries imported by Spring Boot • Create Spring Boot CRUD application	Explain how should use Spring Boot. covers topics such as build systems, auto-configuration, and how to run your applications also cover some Spring Boot best practices.	Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	**3**
		Assessment of final Work.		
		JEE FULL STACK 2.0 WITH ANGULAR - (8 WEEKS)		**48**
40	Demonstrate and lab activity **Agile SCRUM**		Well-equipped computer lab with projector, internet	

			connection. Computer, MS Office using (PPT)	
41	Demonstrate and lab activity Core Java 8		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	
42	Demonstrate and lab activity DevOps (Git, SonarQube, Maven, Jenkins)		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	
43	Demonstrate and lab activity **Database Using PostgreSQL**		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	
44	Demonstrate and lab activity **JPA with Hibernate 3.0**		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	
45	Demonstrate and lab activity Spring 5.0		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	
46	Demonstrate and lab activity HTML 5, CSS 3 with Bootstrap, Javascript, TypeScript		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	
47	Demonstrate and lab activity Bootstramp		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	
48	Demonstrate and lab activity Angular 7		Well-equipped computer lab with projector, internet connection. Computer, MS Office using (PPT)	

	Assessment and Examination			

Text Books / Reference Books / Online Resources

Sr. No	Title of the Book / Link	Author / Webiste	Edition / volume	Text (T) Reference (R)
1.	Introduction to Java programming b	Y. Daniel Liang.		
2	Java The Complete Reference	Herbert Schildt		
3				
4				
5				
6				
7				

8		

Unique Equipment Required:

White Board, Marker, Projector, Laptop/desktop
Computer Lab with wifi and software: OS: Windows, LINUX. Data Analysis tools: SQL, Apache Spark. JDK (Java Development Kit) JDBC driver. Database Tools: MySQL.