PAYILAGAM

+91 8344777333 www.payilagam.com

JAVA FULL STACK FULL SYLLABUS

Java Course Syllabus

Module 1 : Introduction

- Fundamentals of OOP Concepts
- Benefits of OOP Programming.

Module 2 : Introduction to JAVA

- What is Java
- Features of Java.
- Advantages of Java.
- JDK, JVM Architecture.

Module 3 : Java Installation

- Java Installation
- Eclipse IDE Installation
- My First Java Program
- Sample Java Program

Module 4 : Data types, Variables and Operators

- Java Tokens, Alphabets (Character Set).
- Keywords, Java Reserved Keywords, Identifiers, Constants, Data types, Operators.

Module 5 : Control-flow Statements

- Control-Flow Statements
- Decision-Making Statements
- Switch
- IF else
- nested— if

Module 6 : Conditional Statements

- Looping statements
    - while, for, do.. while
    - break, continue.

Module 7 : Classes and Objects

- What is Class, Declaring a class, Rules for Naming Class
- Creating an Object
- new keyword

Module 8 : Exploring Methods

- Method Calling from main() method
- Return values from called method
- Significance of void, return keywords
- Argument Passing, Local Variable, Global Variable
- static, non-static variables and Methods

Module 9 : Polymorphism – Introduction

- Introduction to Polymorphism
- Methods with different no. and type of arguments
- Method Overloading – Compile Time Polymorphism

Module 10 : Encapsulation – Introduction

- Create one more class in same package– Class 2
- Call methods present in Class 1 from Class 2
- Significance of Access Modifiers – private, public and default

Module 11 : Encapsulation – Part 2

- Create one more class in another package – Class 2
- Try to call methods present in Class 1 from Class 2
- Significance of Access Modifier – protected.

Module 12 : Inheritance

- Need for Inheritance
- IS-A relationship

- Usage of extends keyword
- Simple, Multilevel and Hierarchical Inheritance
- Create child class in another package– Class 2
- Significance of Access Modifier – protected.

Module 13 : Polymorphism – Method Overriding

- Method Overriding
- Type Casting
- super and this keywords
- super() and this() keywords

Module 14 : Constructor

- What is Constructor
- How Constructor is being called
- Rules for Constructor
- Constructor Overloading

Module 15 : Getting inputs at runtime

- Scanner Class and its methods
- Converting all their previous programs with Scanner Class

Module 16 : Arrays

- Need for Array
- Types of Arrays
- Array Declaration – Two Ways
- 2D, 3 Dimensional arrays
- Int array, char array, String array
- Converting all their previous programs with Scanner Class

Module 17 : String

- String Class
- String Declaration – two ways
- Heap Memory, String Constant Pool Memory
- Difference between String literal and String Object
- Difference between == operator and equals method
- toString() method, hashCode() method

Module 18 : Abstraction – Introduction

- Details about abstract keyword
- Abstract Class
- Abstract Method
- Creating Object for abstract class not possible – Why
- Dynamic Binding / Late Binding
- final abstract class not possible – How

Module 19 : Inheritance – Interface

- What is Interface
- Difference between Abstract class and Interface
- Usage of implements keyword
- Interface, Sub Interface
- Dynamic Binding / Late Binding

Module 20 : Exception Handling

- What is Exception
- Difference between Exception and Error
- Syntax for Exception Handling Mechanism.
- try Block, Catch Block, Exception , Checked Exceptions
- Catch Block (or) Multiple Catch.
- Throw, Throws, The finally Block
- User Defined Exceptions.

Module 21 : Utility Classes – Introduction

- Collection Framework
- Collection Interfaces
- The List Interface and its implementation classes
- The Set Interface and its implementation classes
- The Queue Interface and its implementation classes
- Map Interface and Its Implementation Classes

Module 22 : Utility Classes – List Interface

- ArrayList and its methods
- LinkedList and its methods
- Difference between ArrayList and LinkedList
- Iterator Methods

Module 23 : Utility Classes – List Interface – Logical Programs

- Getting input from user and Sorting the ArrayList
- Searching in ArrayList
- Replacing element in ArrayList

Module 24 : Utility Classes – Map Interface – Logical Programs

- Find duplicate character in a given String using HashMap
- Find unique characters in a given String using HashMap
- Find count of characters in a given String using HashMap

Module 25 : Generics

- Need for Generics
- Simple Generics
- Sub Typing in generics

Module 26 : Packages

- Predefined Packages
- User Defined Packages

Module 27 : Multi Threading

- Threaded Application
- Thread states
- Runnable interface and Thread class
- Thread Priority
- Interrupting Threads (sleep(), join())
- Synchronization
- Intro. About Inter thread communication

Module 28 : File IO

- File Class
- How To read a File – FileReader, BufferedReader
- How to write in a file – FileWriter, BufferedWriter

Module 29 : Java New Features

- What is Functional Interface?
- Lambda Expression

HTML Essentials for Full Stack Developers

Module 1 : HTML Elements

- Introduction to HTML syntax
- Block-level vs Inline elements
- Common elements: , , <p>, <a>, <img>, <br>, <hr>
- Nesting elements correctly

Outcome: Mini Project #1

Module 2 : HTML Attributes

- Global attributes: id, class, title, style, lang
- data Specific attributes: href, src, alt, type, value, placeholder
- Boolean attributes: checked, disabled, readonly, required

Outcome: Mini Project #2

Module 3 : HTML Forms

- <form> element and its attributes (action, method, etc.)
- Input fields: <input>, <textarea>, <select>, <option>, <button>
- Fieldsets, legends, and labels
- Form validation (required, pattern, minlength, maxlength)

Outcome: Mini Project #3

Module 4 : HTML Lists

- Ordered Lists: <ol>,

- Unordered Lists: <ul>,
- Description Lists: <dl>, <dt>, <dd>
- Nesting and styling lists

Outcome: Mini Project #4

Module 5 : HTML Tables

- Creating tables with <table>, <tr>, <th>, <td>
- Table structure: <thead>, <tbody>, <tfoot>
- Merging cells with colspan and rowspan
- Table accessibility and semantics

Outcome: Mini Project #5

Module 6 : Semantic HTML

- Importance of semantic elements for SEO & accessibility
- Elements: <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>
- When and where to use semantic tags

Outcome: Mini Project #6

Module 7 : HTML Media

- Images: <img>, srcset, alt, picture
- Video: <video>, controls, autoplay, loop, muted
- Audio: <audio>, controls, autoplay, loop, preload
- Embedding YouTube and other media

Outcome: Mini Project #7

Module 8 : Input Types

- Text-based: text, email, password, search, url, tel
- Choice-based: checkbox, radio
- Date/time: date, datetime-local, time, month, week
- Others: file, color, range, number, hidden

Outcome: Mini Project #8

CSS Essentials for Full Stack Developers

Module 1 : CSS Selectors

- Basic selectors: element, class, ID
- Grouping and combining selectors
- Attribute selectors
- Combinators: descendant, child (>), adjacent sibling (+), general sibling (~)

Outcome: Mini Project #1

Module 2 : Box Model

- Understanding content, padding, border, and margin
- Using box-sizing: border-box
- Visualizing box dimensions with dev tools
- Margin collapsing

Outcome: Mini Project #2

Module 3 : Display Property

- block, inline, inline-block, none
- flex

- grid
- visibility: hidden vs display: none

Outcome: Mini Project #3

Module 4 : Positioning

- Static, Relative, Absolute, Fixed, Sticky
- Using top, right, bottom, left with positioning
- Z-index and stacking context

Outcome: Mini Project #4

Module 5 : Flexbox

- Introduction to display: flex
- Main axis vs cross axis
- Properties: justify-content, align-items, align-content, flex-wrap
- Flex item properties: flex, flex-grow, flex-shrink, flex-basis, order

Outcome: Mini Project #5

Module 6 : CSS Grid

- Introduction to grid layout
- Creating columns and rows using grid-template-columns and grid-template-rows
- Placing items with grid-column, grid-row
- Grid gap, alignment, and nested grids

Outcome: Mini Project #6

JavaScript Essentials for Full Stack Developers

Module 1 : Variables & Data Types

- Understand var, let, and const when and why to use each
- Explore JavaScript's powerful dynamic typing system
- Work with strings, numbers, booleans, null, undefined, and more

Module 2 : Operators & Logic

- Arithmetic, assignment, comparison, and logical operators
- Master the building blocks of computation and condition

Module 3 : Conditionals & Control Flow

- Decision-making with if, else, else if, and switch
- Write clean, readable branching logic

Module 4 : Loops & Iteration

- for, while, and do...while loops
- Introduction to forEach() for working with arrays

Module 5 : Functions

- Function declarations vs expressions
- Arrow functions: cleaner syntax, smarter scopes
- Parameters, return values, and function composition

Module 6 : Events & Interaction

- Add interactivity with addEventListener()
- Handle clicks, inputs, mouse events

React Essentials for Full Stack Developers

1. JSX - JavaScript XML

Understand JSX syntax and how it integrates HTML-like code within JavaScript.
Learn JSX rules, expressions, and embedding JavaScript logic into markup.

2. Components: Functional & Class

Build UI with reusable functional components (primary focus).
Introduction to class components and when they are used.
Understand component structure, export/import, and nesting.

3. Props and State

Pass dynamic data to components using props.
Manage component data using state in functional components.
Understand unidirectional data flow and state lifting.

4. Conditional Rendering

Render elements based on logic using if-else, ternary, && operators.
Show loading states, error messages, or user-specific views.

5. List Rendering with map()

Use JavaScript's map() to dynamically generate lists.
Assign unique keys to improve rendering performance.
Render nested data or display lists with interaction.

6. Event HandlingReact Essentials for Full Stack Developers

Attach event listeners like onClick, onChange, onSubmit.
Write custom event handlers and pass data using functions.
Use synthetic events in Reacts cross-browser environment.

7. Component Lifecycle (Class Components)

Learn lifecycle methods: componentDidMount, componentDidUpdate, componentWillUnmount.
Use lifecycle concepts to manage effects like API calls or cleanup.
Understand lifecycle phases: mounting, updating, unmounting.

8. React Hooks

useState: manage local component state.
useEffect: run side effects like fetching data or subscriptions.
useContext: access global state without prop drilling.
useRef: persist values and access DOM elements.

9. React Router

Set up routing using BrowserRouter, Routes, and Route.
Navigate between pages with Link and useNavigate.
Create dynamic routes using URL parameters.React Essentials for Full Stack Developers

10. Fetching Data (APIs)

Fetch data from backend APIs using fetch() or Axios.
Display data using useEffect and handle loading/error states.
Create reusable API utility functions.
Use API data to populate components dynamically.

Postgresql Essentials for Full Stack Developers

Module 1 : Introduction to Databases and PostgreSQL

- What is a Database? Types of databases
- RDBMS vs NoSQL (with examples)
- Why PostgreSQL for Full Stack development?
- Installing PostgreSQL and pgAdmin
- Introduction to PostgreSQL ecosystem (CLI tools, GUI, drivers)

Module 2 : SQL Basics – Table Operations & CRUD

- Filtering with WHERE, AND, OR, IN, BETWEEN, LIKE
- Sorting Results: ORDER BY
- Limiting Data: LIMIT, OFFSET
- Aggregate Functions: COUNT, SUM, AVG, MIN, MAX
- Grouping Data:
- GROUP BY, HAVING

Module 3 : Advanced Joins and Subqueries

- Types of Joins:
    - INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN
- Writing Nested Queries and Subqueries
- Using subqueries in SELECT, WHERE, FROM
- Use cases in reporting and data analysis

Module 4 : Views, Indexes, and Query Optimization

- Creating and using VIEWs
- Materialized Views: use cases and refreshing
- Creating Indexes: B-tree, expression-based indexes
- Query optimization basics with EXPLAIN ANALYZE
- When and how indexes improve performance

Module 5 : Transactions and Data Integrity

- Transactions: BEGIN, COMMIT, ROLLBACK
- Ensuring ACID properties
- Savepoints and nested transactions
- Handling transaction errors
- Use cases: banking systems, form submissions

Module 6 : Functions, Stored Procedures, and Triggers

- Defining SQL & PL/pgSQL functions
- Returning values, variables, and conditional logic
- Stored Procedures
- Triggers:
    - BEFORE and AFTER triggers for INSERT, UPDATE, DELETE

Module 7 : Relational Modeling and Database Design

- Normalization:
    - 1NF, 2NF, 3NF with examples
- Entity Relationship Diagrams (ERD)
- Implementing:
    - One-to-One
    - One-to-Many
    - Many-to-Many (junction tables)
- Using UUID as Primary Keys

Module 8 : Access Control and Security

- Creating database users and roles
- Granting and revoking permissions:
    - GRANT, REVOKE
- Role-based access control (RBAC)
- Best practices for securing PostgreSQL in production
- Schema-level and table-level permissions

Springboot Project Oriented Training for Full Stack Developers

Module 1: Spring MVC Architecture

- What is a server, basic client-server architecture?
- Understanding HTTP methods.
- What is Spring and Spring Boot?
- Learning the architecture diagram of Spring MVC, understanding the flow of a Spring boot.

Module 2: Maven Key Concepts

- How to build a maven project.
- Maven Life Cycle.
- Building Jar and War.
- Difference between Jar and War.

Module 3: Registration Form Using Spring Boot.

- Understanding the basic Annotations used for building a Monolithic Enterprise Application.
- Creating Pojo, using Lombok dependency.
- Learn the basics of Thymeleaf
- Connecting Controller to View in MVC Model
- Validation of incoming request.

Module 4: Color Picker using Spring Boot

- Getting Inputs from Thymeleaf View
- Passing it to Controller
- Learn the basics of Thymeleaf
- Passing List of String Objects to View

Module 5: Simple Calculator with Spring Boot

- Getting Inputs from View
- Passing inputs to Controller
- Processing inputs at Controller Side

Module 6: Product Catalogue for E-Commerce Site

- Adding Entity Class
- Passing Model Values to Controller
- Transforming Objects to View from Controller

Module 7: Quiz Application using Spring Boot

Creating Quiz Application using Thymeleaf / React and Springboot and publishing the results

Module 8: To Do List

- Learning about HTTPSession Maintenance
- Creating To Do List with Thymeleaf / React

- Add To Do Item, Delete To Do Item
- Delete All To Do Items

Module 9: REST API Integration with SpringBoot

- What is REST API?
- How to Access API Response
- Accessing REST results through PostMan
- Open Weather Map API Integration
- Getting Weather Details

Module 10: REST API Integration with SpringBoot

- Swagger UI
- Open API

Module 11: Spring Exception Handling

- Inbuilt Exception Handling in Spring boot.
- Handling default exception and user defined exception.
- Presenting proper error messages as per the exception occurred.

Module 12: Usage of Log4j

- Usage of Log4j
- Adding Log4j in the project
- Generation of log for all the necessary services.

Module 13: Spring JPA

- What is a Repository and its types?
- Difference between CRUD and JPA repository.
- User-defined Query.
- Joining tables and do operations on the data.

Module 14: Spring Connectivity with PostgreSQL

- Usage of JDBC Driver.
- Setting path in application.properties file.

Module 15: Spring Security

- How spring security works.
- Explaining Spring Default authentication.
- Exploring Basic Auth methods.
- Create our own authentication credentials.
- Creating various roles for users.
- Authorization and Authentication as per roles.

Module 16: E-Commerce Project

- Creating JSON

- Product List Page
- Add To Cart Page
- Cart To Database
- Delete From Cart
- View Cart