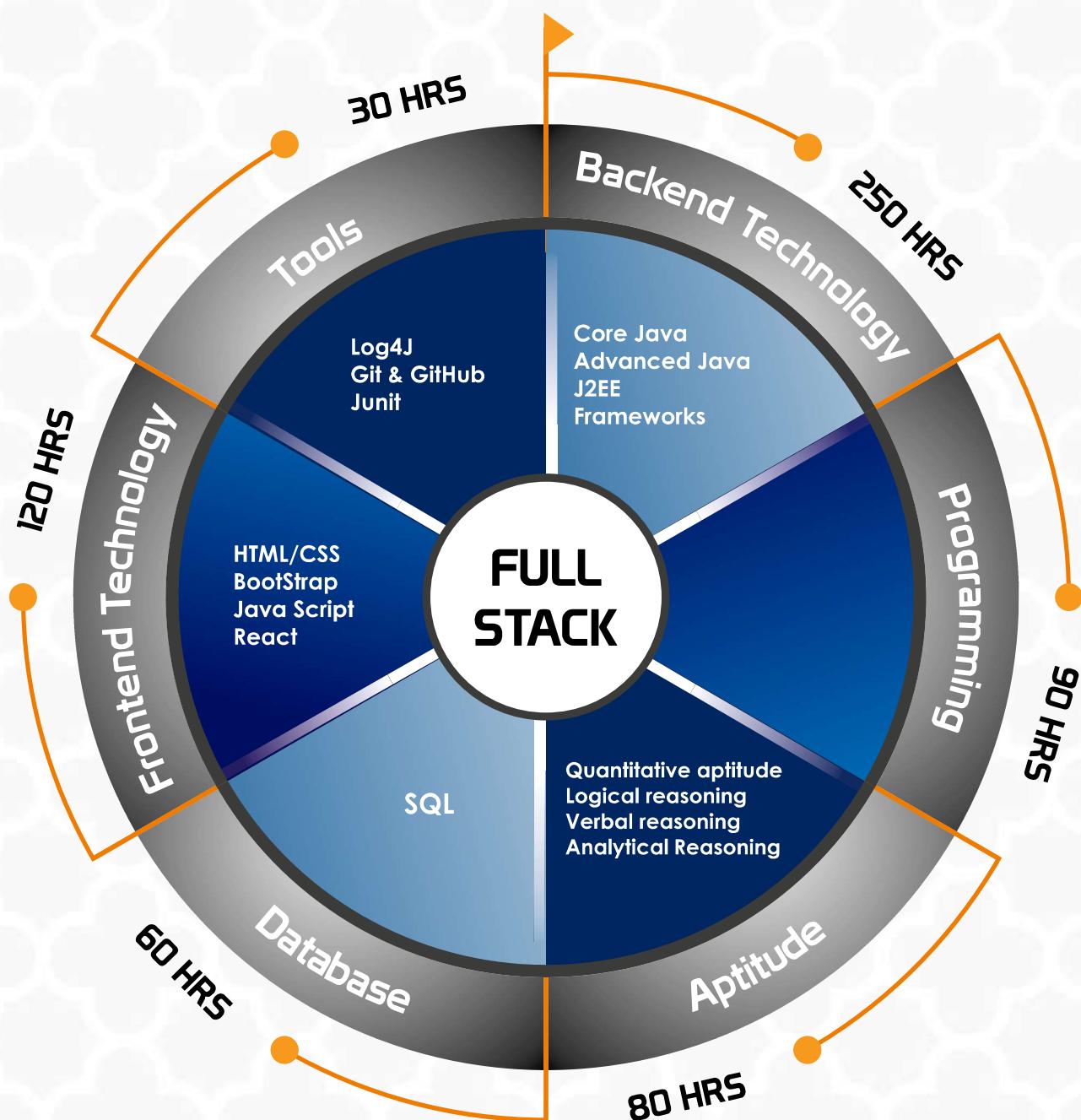




JAVA

FULL STACK

COURSE OVERVIEW



Introduction

- ◆ Evaluation of programming language.
- ◆ Types of programming language and paradigms.
- ◆ Java - what, where and why ?
- ◆ Comparison in Java with C and C++.
- ◆ Java's Magic Byte code.
- ◆ Java Virtual Machine(JVM) , JDK,JRE,JIT.

Java Architecture

Language Fundamentals

- ◆ Installation process.
- ◆ Compilation and Executions procedure using different editor & IDE.
- ◆ Java Source File Structure.
- ◆ Naming conventions.

Data types and Variables

- ◆ Primitive data type Integers ->byte, short, int, long. Floating point->float , double. Character , Boolean.
- ◆ Variable declaration and initialization.
- ◆ Value type and reference type.
- ◆ Types and scope of variables.

Control statements in Java

- ◆ Selection statements -- if, if else, switch.
- ◆ Iteration statements -- while, do-while, for, for-each, nested loop
- ◆ jump statements -- using break, using continue.

Arrays

- ◆ Defining array.
- ◆ Single Dimension. Initializing and Accessing array.
- ◆ Multi Dimension, Jagged Array.
- ◆ Sorting ,inserting, delete dynamically.

The Java Library: String Handling

- ◆ String operations.
- ◆ Character extractions.
- ◆ String comparison.
- ◆ Modifying a String.
- ◆ Changing the case of characters.
- ◆ String Buffer class and its methods.
- ◆ String Builder class and its methods.





Object Oriented programming

- ◆ Class Fundamentals.
- ◆ Object & Object reference.
- ◆ Life time of object & Garbage Collection.
- ◆ Constructor & initialization code block.
- ◆ Nested, Inner Class & Anonymous Classes.
- ◆ Defining Methods, Argument Passing Mechanism.
- ◆ Method Overloading, Recursion.
- ◆ Data Encapsulation.
- ◆ Inheritance.
- ◆ Method overriding
- ◆ Access modifier
- ◆ Polymorphism
- ◆ Association, Aggregation, Composition.
- ◆ Abstract classes.

Interface

- ◆ Implementing interface.
- ◆ Extending interface.
- ◆ Tagging interface.
- ◆ static and default methods.
- ◆ Design pattern-Factory Pattern.

Utilize the CPU time using Multithreaded Programming

- ◆ Java threaded model.
- ◆ Thread priorities.
- ◆ Creating Multiple threads by extending Thread class / by implementing Runnable interface.
- ◆ Using isAlive() and join().
- ◆ Daemon thread.
- ◆ Synchronization.
- ◆ Race condition.
- ◆ Obtaining a Thread State.
- ◆ Inter Thread Communication



The Collection Framework

- ◆ Collections Overview.
- ◆ The Collection interfaces –The collection interface, The List interface, The Set interface, The Queue interface, The Deque interface.
- ◆ The Collection classes – ArrayList class, LinkedList class, HashSet class, Linked Hash Set class, TreeSet class, Priority Queue, Array Deque, Generics.
- ◆ Maps -key value pairs.
- ◆ Maps Hierarchy.
- ◆ Map interface.
- ◆ TreeMap class.
- ◆ HashMap class.
- ◆ LinkedHashMap class.
- ◆ Sorting Objects - using Collection.sort().
- ◆ Sorting complex Objects -using Comparator interface.
- ◆ Using Comparable interface.

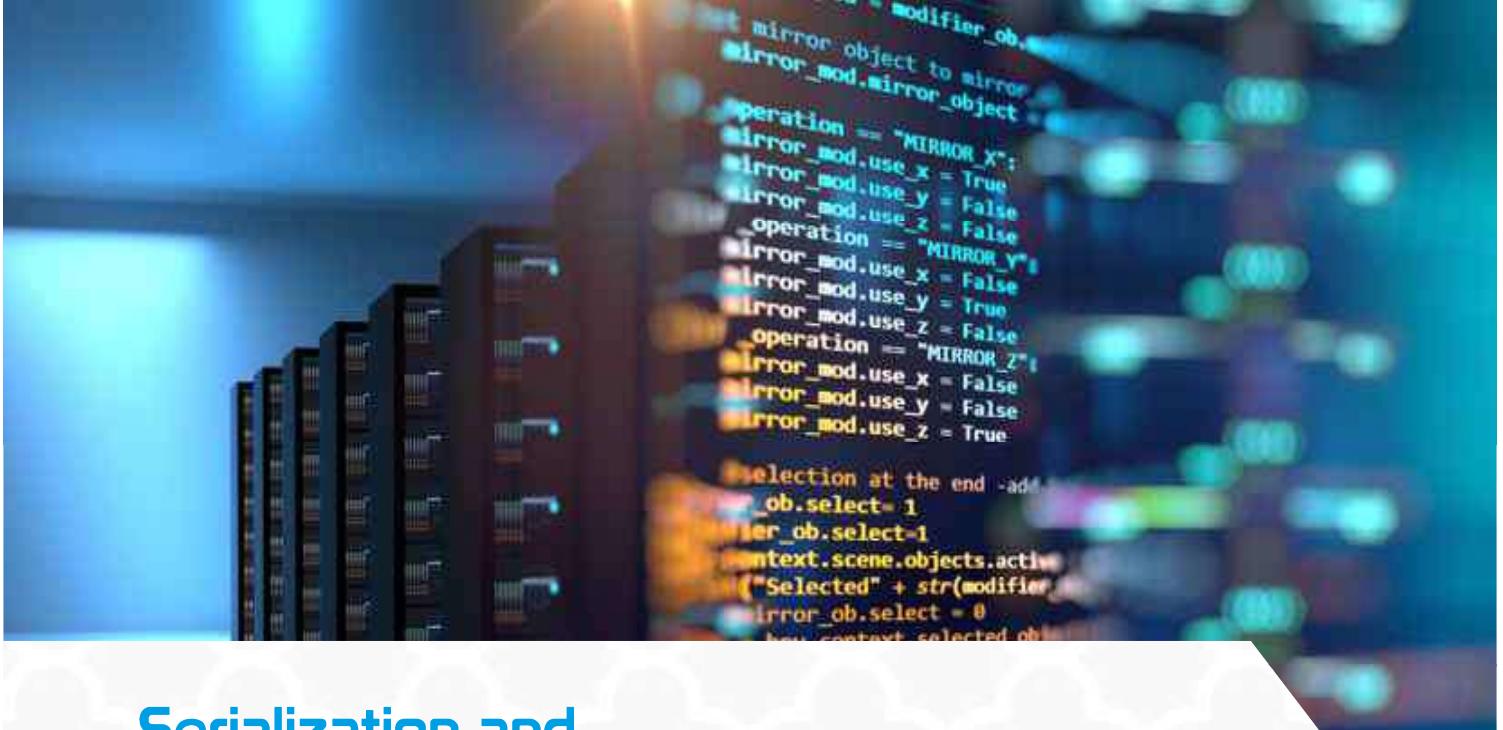
Lambda Expression

- ◆ Introduce Lambda Expression fundamentals.
- ◆ Functional Interface.
- ◆ Lambda Expressions and Exceptions.
- ◆ Method reference to static and instance methods.
- ◆ Constructor reference
- ◆ Predefined functional Interface.
- ◆ Enumerations.
- ◆ Annotations.

Input/Output operations in Java

- ◆ Understanding the streams.
- ◆ Creating files and folders using Java code.
- ◆ File Input Stream & File Output Stream.
- ◆ Buffered Reader and Buffered Writer class.
- ◆ Data Input Stream and Data Output Stream.
- ◆ Flushing Buffered Streams.





Serialization and Deserialization

- ◆ Serializing an Object, Deserializing an Object.
- ◆ Selective serialization
- ◆ Externalization interface.
- ◆ Secure Hash Algorithm.

JDBC

- ◆ Introduction to JDBC , JDBC API to connect Java and Database.
- ◆ Drivers and types of Driver. Complete Architecture of JDBC.
- ◆ Steps to connect Two applications.
- ◆ Connectivity with Oracle, Connectivity with MYSQL.
- ◆ Application to create tables.
- ◆ Application to insert ,delete, update, drop table.
- ◆ Metadata extraction from ResultSet.
- ◆ SQL commands to receive data from DB. -Statement, -Prepared Statement, -Callable Statement.
- ◆ Scrollable Result Set.
- ◆ Batch Updates.
- ◆ Transaction management.
- ◆ Stored procedure.

Network Programming

- ◆ Networking basics.
- ◆ Client-server Architecture. 1-tier,2-tier, 3-tier and n-tier architecture.
- ◆ Socket programming -one way communication. -two way communication.
- ◆ URL classes.

J2EE Server Side Programming Servlets

- ◆ Server
- ◆ Installation And Basic Web Application
- ◆ JEE Container
- ◆ JEE Application
- ◆ Structure Of JEE App
- ◆ Static And Dynamic Application
- ◆ Presentation, Persistence & Business Logic
- ◆ Web.xml / Deployment Descriptor Stub
- ◆ Servlet
- ◆ Generic Servlet
- ◆ Http Servlet
- ◆ Configuration Of Servlet In Web.xml
- ◆ Data
- ◆ Ui/Form Datacode For Ui Data &



- ◆ Enumeration
- ◆ Servlet Life Cycle
- ◆ Load On Startup
- ◆ Servlet Config
- ◆ Servlet Context
- ◆ Code For Servlet Context
- ◆ Programmatic Data & Declarative Data
- ◆ Http Servlet
- ◆ Servlet Chaining
- ◆ Request Dispatcher
- ◆ Cookie
- ◆ Session

Java Server Pages

- ◆ Jsp Architecture
- ◆ Specifications Of Jsp
- ◆ Life Cycle Of Jsp
- ◆ Criteria For Jsp
- ◆ Scopes Of Jsp
- ◆ Directives Of Jsp
- ◆ Taglib Directive
- ◆ Code For Jsp
- ◆ Code For Include Directive
- ◆ Code For Include Standard Action

Hibernate

- ◆ Introduction.
- ◆ Data persistency.
- ◆ ORM [Object Relational Mapping].
- ◆ Hibernate Architecture.
- ◆ Steps to prepare Hibernate application.
- ◆ CRUD operations using ORM approach.
- ◆ Servlet-Hibernate Application.
- ◆ JSP-Hibernate Application.
- ◆ Persistence Object lifecycle, Schema Generation Tools.
- ◆ Primary Key Generation Algorithms in Hibernate.
- ◆ Hibernate Mapping.
- ◆ Bulk Operation.
- ◆ Native SQL.
- ◆ Criteria API.
- ◆ Hibernate Filter.
- ◆ Hibernate Transaction Management.
- ◆ Hibernate Connection pooling.

JSP Standard Tag Library (JSTL)

- ◆ Introduction to JSTL tag libraries.
- ◆ Downloading external lib jar to access inbuilt functions.
- ◆ Core tags.
- ◆ SQL tags.
- ◆ Function tags.
- ◆ MVC Architecture type-1
- ◆ MVC Architecture type-2

Spring

- ◆ Introduction to Spring Framework.
- ◆ Steps to prepare Spring Application.
- ◆ Spring Core Module.
- ◆ Spring JDBC-DAO Module.
- ◆ Spring AOP Module.
- ◆ Spring Transaction module.
- ◆ Spring ORM Module.
- ◆ Spring Web and MVC Module.
- ◆ Spring Security Module.
- ◆ Maven tool.
- ◆ Spring Boot.

Database

- ◆ Introduction -what is Database - RDBMS Terminology -MySQL Database
- ◆ DATA
- ◆ Attributes
- ◆ Entity
- ◆ Data base
- ◆ Crud operation
- ◆ DATA BASE MANAGEMENT SYSTEM (DBMS)
- ◆ Types of DBMS
- ◆ RDBMS (Relational data base management)



- ◆ RATIONAL MODEL
- ◆ RDBMS /DBMS
- ◆ RDBMS/ Excel sheet

OBJECTS

- ◆ Large object
- ◆ Binary large object
- ◆ Date
- ◆ Number
- ◆ Precision

CONSTRAINTS

- ◆ Types of constraints
- ◆ Unique constraints
- ◆ Not null constraints
- ◆ Check constraints
- ◆ Primary key constraints
- ◆ Foreign key constraints

STATEMENTS

- ◆ Types of SQL statement
- ◆ DDL
- ◆ DML
- ◆ TCL
- ◆ DCL
- ◆ DQL

DATA QUERY LANGUAGE (DQL)

- ◆ Selection
- ◆ Projection
- ◆ Selection
- ◆ Joins

PROJECTION

- ◆ From
- ◆ Select
- ◆ Order of execution
- ◆ Asteric wild-char
- ◆ Expression
- ◆ Alias
- ◆ Distinct

SELECTION

- ◆ Where
- ◆ Order of execution

OPERATORS

- ◆ Types of operators
- ◆ Arithmetic operators
- ◆ Comparison operators
- ◆ Relational operators
- ◆ Logical operators
- ◆ Special operators
- ◆ Concatenation operators
- ◆ Sub -query operators

FUNCTIONS

- ◆ Types of functions
- ◆ Single row functions
- ◆ Multi-row functions
- ◆ Upper
- ◆ Dual
- ◆ INIT CAP
- ◆ Reverse
- ◆ Length
- ◆ MOD
- ◆ Concat
- ◆ Substr
- ◆ Replace
- ◆ Instr
- ◆ NVL

CLAUSES

- ◆ GROUP BY CLAUSE
- ◆ Group- by expression
- ◆ Order of execution
- ◆ Nature of execution
- ◆ HAVING CLAUSE
- ◆ Order of execution
- ◆ Nature of execution
- ◆ ORDER BY CLAUSE
- ◆ Order of execution
- ◆ Nature of execution

SUB QUERY

- ◆ Working principle of sub -query
- ◆ Rules of sub-query



TYPES OF SUB-QUERY

- ◆ Single row sub-query
- ◆ Multi-row sub-query
- ◆ All operator
- ◆ Any operator
- ◆ Nth MAX SAL & Nth MIN SAL
- ◆ Employee manager relationship
- ◆ To find manager
- ◆ To find Reporter

PSEUDO COLUMN

- ◆ Row Id
- ◆ Row num

JOINS

- ◆ Types of joins
- ◆ Cartesian join/Cross join
- ◆ Inner join/EQUI join
- ◆ Natural join
- ◆ Outer join
- ◆ Left outer join
- ◆ Right outer join
- ◆ Full outer join
- ◆ Self join

DDL (DATA DEFINITION LANGUAGE)

- ◆ Create
- ◆ Rename
- ◆ Alter
- ◆ Truncate
- ◆ Drop

DML (DATA MANIPULATING LANGUAGE)

- ◆ Insert
- ◆ Update
- ◆ Delete

TCL (TRANSACTION CONTROL LANGUAGE)

- ◆ Commit
- ◆ Save point
- ◆ Roll back

- ◆ DCL (DATA CONTROL LANGUAGE)
- ◆ Grant
- ◆ Revoke

SET OPERATION

- ◆ Union
- ◆ Union all
- ◆ Intersection
- ◆ Minus
- ◆ VENN DIAGRAM FOR JOINS

DATABASE PROPERTIES

- ◆ ACID properties
- ◆ Key attributes
- ◆ Non key attributes
- ◆ Primary key attributes
- ◆ Non primary key attributes
- ◆ Composite key attribute
- ◆ Foreign key attribute
- ◆ Anomalies

DEPENDENCY

- ◆ Types of dependency
- ◆ Total functional dependency
- ◆ Partial function dependency
- ◆ Transitive functional dependency

NORMALIZATION

- ◆ Levels of normal form
- ◆ First normal form(1NF)
- ◆ Second normal form(2NF)
- ◆ Third normal form (3NF)
- ◆ BCNF (Boyce code normal form)

CO-RELATED SUB-QUERY

- ◆ Working principle
- ◆ Exist operator
- ◆ Not-exist operator
- ◆ ER diagram
- ◆ ER diagram Model
- ◆ Cardinality Number
- ◆ Interview grooming



HTML5

- ◆ HTML Basics.
- ◆ HTML5 Introduction.
- ◆ HTML5 Semantic Elements.
- ◆ HTML5 Canvas.
- ◆ HTML5 forms.
- ◆ HTML5 MathML.

CSS

- ◆ CSS Basics(1.0 and 2.0).
- ◆ CSS Transitions.
- ◆ Advanced Selectors in CSS.
- ◆ CSS Gradients.
- ◆ CSS Web Fonts.
- ◆ CSS @media Rule.
- ◆ CSS Multiple Backgrounds.
- ◆ CSS Multiple Columns.
- ◆ CSS 3D Transforms.
- ◆ CSS Website Layout.

Bootstrap

- ◆ Bootstrap Buttons.
- ◆ Bootstrap Forms.
- ◆ Bootstrap Navbars.
- ◆ Bootstrap Grid System.
- ◆ Bootstrap images.
- ◆ Bootstrap Tables.
- ◆ Bootstrap – Jumbotron.
- ◆ Bootstrap - Button Groups

JavaScript

- ◆ Introduction.
- ◆ Data Types.
- ◆ JavaScript Functions.
- ◆ JavaScript Arrays.
- ◆ JavaScript Forms.
- ◆ DOM Elements.

React

Introduction to ReactJs

- ◆ Importance of nodeJS and Installing Node

JS Starting and Stopping a React App
Javascript Module Systems.

JSX Usage

- ◆ Converting HTML to JSX.
- ◆ Inline Styling with JSX.
- ◆ Converting Styling to JSX Format.
- ◆ Class vs ClassName.
- ◆ Referencing JS Variables in JSX.

ReactJS – Props

- ◆ Default props.
- ◆ Using props.
- ◆ State and Props.
- ◆ Passing and Receiving Props.
- ◆ Passing Multiple Props.
- ◆ Validating Props.

ReactJS - Component API

- ◆ setState().
- ◆ forceUpdate.
- ◆ ReactDOM.findDOMNode().
- ◆ Lifecycle Methods.
- ◆ Invoking Callbacks in Children.

ReactJS – Forms

- ◆ Simple Example.
- ◆ Showing Forms to the User.
- ◆ Creating Event Handlers.
- ◆ Child Events.
- ◆ Receiving Values From Controlled Elements.

ReactJS - Using Flux

- ◆ Installing Redux.
- ◆ Create Files and Folders.
- ◆ Actions.
- ◆ Reducers.
- ◆ Store.
- ◆ Root Components.
- ◆ Other Components.
- ◆ How To Use Immutable.js in a React Redux Application.



Building Record List

- ◆ Rendering Lists.
- ◆ Review of Map Statements.
- ◆ Rendering Lists of Components.
- ◆ The Purpose of Keys in Lists.
- ◆ Implementing Keys in Lists.
- ◆ Usage of Refs and Keys and Routers.

Managing App State with Redux Reducers

- ◆ Containers - Connecting Redux to React .
- ◆ Implementation of a Container Class.
- ◆ Containers and Reducers Review.
- ◆ Actions and Action Creators.
- ◆ Binding Action Creators.
- ◆ Creating an Action.
- ◆ Consuming Actions in Reducers.
- ◆ Consuming Actions in Reducers Continued.
- ◆ Conditional Rendering.

Code Review Deployment

- ◆ Tools for code review
- ◆ Standard coding conventions

Junit

- ◆ Annotations for Junit testing
- ◆ Assert class

Log4j

- ◆ Introduction & Installation
- ◆ Architecture
- ◆ Example
- ◆ Maven
- ◆ Properties
- ◆ XML
- ◆ Logging Levels
- ◆ Logging Methods
- ◆ HTML Layout
- ◆ Log Formatting
- ◆ Logging in Files
- ◆ Pattern Layout
- ◆ Appenders

Git

- ◆ Git Introduction
- ◆ Installation & Environment Setup
- ◆ Git Tools
- ◆ Git Terminology
- ◆ Git Commands
- ◆ Git Flow
- ◆ Cheat Sheet
- ◆ Staging & commits
- ◆ Undoing changes
- ◆ Inspecting changes
- ◆ Branching & Merging
- ◆ Collaborating

Projects

1. Console based Hands On projects.
2. Stand Alone Gaming projects.
3. Enterprise Edition projects.
4. Framework projects.
5. Web development projects.

Note: All projects will be implemented and deployed in live environment.