

Core Java Syllabus

Introduction

- What is Java
- History and Features of Java
- C++ vs. Java
- Hello World Java Program
- How to set the JDK path?
- JDK, JRE, and JVM (Java Virtual Machine)
- JVM Memory Management
- Internal Details of JVM
- What is class-loader?
 - How class-loader works
 - Different types of class-loaders

Java Fundamentals

- Unicode System
- Comments
- Identifiers
- Literal
- Keywords
- **Operators**
 - Arithmetic Operator
 - Assignment Operator
 - Relational Operator
 - Logical Operator
 - Increment and Decrement Operators
- **Decision Control**
 - If, if-else, if-else-if
 - Switch case
- **Repetition Control**

- For loop
- While loop
- Do while loop
- for-each loop
- Break vs Continue.
- Type casting, Default value

Data Types in Java

- **Primitive data types**
 - int
 - long
 - short
 - byte
 - float
 - double
 - char
 - boolean
- **Non-Primitive data types**
 - Array
 - Class
 - String
- Primitive VS Non-Primitive Memory allocation

Variable

- Variable declaration and initialization
- Proper naming convention
- **Types of variable**
 - Static variable
 - Local variable
 - Instance variable
- And memory allocation

Final Keyword

- Final variable

- Final method
- Final class

Super and this keyword

- this keyword
- Calling of current class constructor, and method
- super keyword
- Super class method
- Super class constructor
- Super key use to call super class object and instance variable

Use of static keyword.

- What is Static keyword
- Static variable
- Static method
- Static block, instance block
- Static nested class

String in Java

- What is String
- What is StringBuilder
- What is StringBuffer
- String vs StringBuffer vs StringBuilder
- Immutable String and StringBuffer, StringBuilder why not?
- And it's methods

Array

- What is Array
- Single Dimensional Array
- Multi Dimensional Array
- **How to declare array in java**
 - `int intArray[]; // declaring array`
 - `intArray = new int[20]; // allocating memory to array`

- `int[] intArray = new int[]{ 1,2,3,4,5,6,7,8,9,10 }; //`
Declaring

Access modifier in java

- Private
- Default
- Public
- Protected

Method in Java

- Use of method in Java
- Method declaration(Naming convention[camel case])
- Main method
- Type of method
 - Predefine methods
 - User define methods
- Calling of method
- Define a method with return type

Constructor in Java

- What is constructor in java
- **Types of constructor in java**
 - Default constructor
 - Parameterized constructor
- Constructor chaining

OOPs in Java

- Class and Objects
- How to declare a class
- How to create an object
- Object declaration and initialization

- **Type of Object**
 - Anonymous object
 - Object with reference
- **Inheritance in OOPs**
 - Single level Inheritance
 - Multilevel, Multiple inheritance
 - Is-A Relationship
 - Aggregation and Composition(HAS-A)
 - Hierarchical inheritance
- **Polymorphism in OOPs**
 - Run-time polymorphism
 - Compile-time polymorphism
 - Rules of method Overloading and method Overriding
 - Covariant Return type
- **Abstraction in OOPs**
 - Abstract class
 - Abstract method
- **Encapsulation in OOPs**
 - How to achieve Encapsulation
 - Data hiding
 - Tightly encapsulated class
 - Getter and setter method in Java
 - Naming convention of getter and setter method
- **Interface in OOPs**
 - Interface in Java
 - Nested interface, rules, and example programs
 - Functional Interfaces
- Abstract class vs interface
- Top-Down vs Bottom-Up Approach

Inner Class

- Member Inner Class

- Static Inner Class
- Local Inner Class
- Anonymous Inner Class

Exception Handling

- When an exception occurs
- Importance of Exception Handling
- Exception Types
- Using try and catch
- Multiple Catch Block
- Throw, throws, finally
- User defined Exceptions
- Throw vs Throws
- Final vs Finally vs Finalize
- Different ways to print Exception messages in Java

MultiThreading in Java

- Creating Thread with various types
- **Thread life cycle**
 - New
 - Runnable
 - Blocked
 - Waiting
 - Timed Waiting
 - Terminated
- Thread Scheduler.
- Sleep method()
- How to call thread and it's types
- Difference between direct call run method and call start method
- Thread priority
- Get name of thread
- Join Thread

- User define name of Thread
- Start a thread twice
- Interrupting Thread
- Daemon Thread
- **Synchronization**
 - Synchronized method
 - Synchronized block
 - Static synchronization
- Thread pool
- Thread group
- Shutdownhook

Wrapper Classes in Java

- Introduction
- Byte
- Short
- Integer
- Long
- Float
- Double
- Character
- Boolean classes

Collection in Java

- **List**
 - ArrayList
 - LinkedList
 - Vector
 - Stack
 - And it's methods
- **Set**
 - HashSet
 - LinkedHashSet

- SortedSet
- TreeSet
- And it's methods
- Queue
 - Priority Queue
 - Deque
 - ArrayDeque
 - And it's methods

Map in Java

- HashMap
- LinkedHashMap
- TreeMap
- And it's methods

Other Usefull Topic

- What is generic
- Creating User-defined Generic classes
- Iterator
- ListIterator
- Enumerator
- Annotations
- Wildcards in Java
- Serialization and Deserialization
- Comparable interface
- Comparator interface
- Regular Expressions

Java Reflection

- Reflection API
- newInstance() method
- javap toolcreating
- javap tool creating

- java.lang.Class class

Java IO

- Getting User Input
- Writing Program Output
- Creating Files
- Reading a Text File
- Writing data to a Text File
- Comparing two text files
- Handling Excel file
- Scanner Class
- Console Class

Java 8 Features

- **New Features in Java 8**
 - Lambdas and Functional Interfaces
 - Interface Default and Static Methods
 - Method References
 - Repeating annotations
 - Better Type Inference
 - Extended Annotations Support
 - Functional Interface
- **Default Methods**
 - Multiple Defaults
 - Static Default Methods
- **New Features in Java libraries**
 - **Optional**
 - Class Declaration
 - Class Methods
 - **Stream**
 - What is Stream
 - Generating Streams
 - forEach

- Map
- Filter
- Limit
- Sorted
- Collectors
- **Date/Time API**
 - Local Date-Time API
 - Zoned Date-Time API
 - Chrono Units Enum
 - Period & Duration
- Nashorn JavaScript engine
- Parallel Arrays
- Concurrency
- `forEach()` method

Java 9 Features

- Platform Module System (Project Jigsaw)
- Interface Private Methods
- Try-With Resources
- Anonymous Classes
- `@SafeVarargs` Annotation
- Collection Factory Methods
- Process API Improvement
- New Version-String Scheme
- Process API Improvement
- Control Panel
- Stream API Improvement

Java 10 Features

- **Unmodifiable Collections**
 - `copyOf()`
 - `toUnmodifiable()`
- Local Variable Type Inference("var")

- Optional
 - `orElseThrow()`
- `List.copyOf()`, `Set.copyOf()`, and `Map.copyOf()`
- APIs for Creating Unmodifiable Collections
 - `Collectors.toUnmodifiableList()`
 - `toUnmodifiableSet()`
 - `toUnmodifiableMap()`

Java 11 Features

- Java String Methods
 - `isBlank()`
 - `lines()`
 - `strip()`, `stripLeading()`, `stripTrailing()`
- Local-Variable Syntax for Lambda Parameters

Package in Java

- How to declare a package.
- Package naming conventions.
- Sub packages.
- Types of packages
 - User-defined packages
 - Built-in packages
- Importing packages in Java
- Role of public, protected, default and private w.r.t package
- Autoboxing and Auto unboxing

Garbage Collection

- Need of Garbage collection
- Unreachable Objects
- How Garbage collection is done
- `Finalize()` Method
- Different types of References

JDBC in Java

- Introduction

- JDBC Architecture
- JDBC Drivers
- Types of Drivers
- Steps to connect to Database
- Connectivity with MySQL
- DriverManager
- **Types of JDBC statements**
 - Statement
 - Prepared statement
 - Callable statement
- Database Metadata, Resultset Metadata
- **ResultSet**
 - Read Only ResultSet
 - Updatable ResultSet
 - Forward Only ResultSet
 - Scrollable ResultSet
- PreparedStatement
- **Image**
 - Sorting
 - Retrieving image
- **File**
 - Storing file
 - Retrieving file
 - Stored procedures
 - and functions
- Prepared Statement
- Connection Modes
- Batch Updations
- CallableStatement
- BLOB & CLOB

Servlets

- Basic of Servlet
 - Introduction
 - Servlet: What and Why?
 - Basics of Web
 - Servlet API
 - Servlet Interface
 - Generic Servlet
 - Http Servlet
 - Servlet Life Cycle
 - Working with Apache Server
 - Steps to create a servlet in
 - How servlet works
 - servlet in Eclipse
- Servlet-Browser Communication
 - sendError
 - setHeader
 - sendRedirect
- Web-Component Communication
 - Forward
 - Include
- Session Tracking Mechanisms
 - Session tracking Mechanisms
 - Http Session
 - Cookies
 - URL-Rewriting
 - Hidden-Form Fields
 - Filters & Wrappers
 - Listeners
 - Web-Security
 - Servlet Request

- Servlet Request methods
- Registration example with DB
- **Servlet Collaboration , Servlet Config and Servlet Context**
 - Request Dispatcher
 - Send Redirect
 - Servlet Config
 - Servlet Config
 - Servlet Context
 - Servlet Context
- **Attribute**
 - How to set
 - How to get
 - How to remove
- **Session Tracking**
 - Cookies
 - Hidden Form Field
 - URL Rewriting
 - Http Session
- **Filter**
 - Filter
 - Authentication Filter
 - Filter Config
 - Useful examples

JSP

- Introduction
 - JSP LifeCycle
 - JSP Implicit Objects & Scopes
 - JSP Directives
 - page
 - include
 - taglib
- JSP Scripting Elements
 - Declaratives
 - Scriptlets
 - Expression
- What is JSP
 - Advantage of JSP over Servlet
 - Life cycle of a JSP Page
 - Creating a simple JSP Page
 - The JSP API
 - javax.servlet.JSP
 - javax.servlet.JSP.tagext
 - JSPPage interface
 - HttpJSPPage interface
 - JSP Scriptlet tag
 - Example of JSP scriptlet tag
 - JSP expression tag
 - Example of JSP expression tag
 - JSP Custom Tag API
 - JSPTag interface
 - Tag interface
 - IterationTag interface
 - TagSupport class
 - Example of JSP Custom Tag
 - Create the Tag handler class

- Create the Tag Library Descriptor(TLD) file
- JSP file that uses the Custom tag defined in the TLD file
- Attributes in JSP Custom Tag
- Iteration using JSP Custom Tag
- Looping using Iteration Tag (creating tag for loop)
- Custom URI in JSP Custom Tag
- Pagination in JSP
- JSP CRUD Example
- Development in JSP
- Registration Form in JSP
- Uploading file to the server using JSP
- MultipartRequest class
- Downloading file from the server using JSP
- Scripting elements
 - Scriptlet tag
 - Expression tag
 - Declaration tag
- **Implicit Objects**
 - Out
 - Request
 - Response
 - Config
 - Application
 - Session
 - Page Context
 - Page
 - Exception
- **Directive Elements**
 - Page directive
 - Include directive
 - Taglib directive
- **Action Elements**
 - JSP:forward

- JSP:include
- Bean class
- JSP:use Bean
- JSP:set Property & JSP:get Property
- Displaying applet in JSP
- **Custom tags**
 - Custom Tag: What and Why?
 - Custom Tag API?
 - Custom Tag Example
 - Attributes
 - Iteration
 - Custom URI
- **JSP Action(Standard Actions)**
 - useBean tag
 - setProperty tag
 - getProperty tag
 - include tag
 - forward tag
 - param tag
 - plug-in tag
 - params tag
 - fallback tag
 - directives tag
 - scriptlet tag
 - expression tag
 - Custom Actions:-Classic Tags,Simple Tags.

Spring Boot in Java

➤ Introduction

- Spring Boot Introduction
- Spring Boot Basics
- Why Spring Boot
- Spring Boot Features

- Main Goal of Spring Boot
- Creating Project- Spring Initializer
- Spring Boot Hello World Application
- Spring Boot Annotations
- Inversion of Control (IOC)
- Dependency Injection
- **Maven**
 - Pom.xml
 - Dependencies
 - scope of the dependency
 - version of the dependency
 - Understanding of every tag in pom.xml like groupid, artifactid
 - **Different stages of maven**
 - clean
 - install
 - test
 - run
 - debug
- Spring Boot Actuator
- Spring Boot DevTools
- Spring Boot Build Systems
- Spring Boot Code Structure
- Spring Boot with RESTful Service
- Spring Boot with Logger
- Spring Boot MVC Pattern
- Spring Boot Common Application Properties (application.properties)
- **Spring Boot, Spring Data with RESTful Web Service and How to create REST APIs**
 - Rest API
 - Rules to create rest api

- Different methods GET, POST, DELETE etc.
- QueryParam
- Path Variable
- RequestBody
- Validation on RequestBody
- Spring Boot, Spring Data JPA CRUD Application
- Spring Boot, Spring- JdbcTemplate CRUD
- Spring Boot, MySQL, Project Lombok CRUD Application
- Spring Boot with XML Payload using PostmanAPI
- Spring Boot with JSON Payload using PostmanAPI
- Spring Boot Global Custom Validations
- Spring Boot Global Custom Exception Handling
- Global Exception Handled @ControllerAdvice
- Spring Boot Code Debug Practices
- Spring Boot Application connectivity with Multiple Databases
- Spring Boot Project Lombok, Swagger UI
- Spring Boot Local Network Chat Application
- Spring Boot with Spring Security- JWT
- Spring Boot Scheduler
- Spring Boot Custom Port Number
- Deploy Spring Boot Applications on External Server and deploy war file
- Spring Boot QR Code Generator Hands on Session
- Spring Boot QR Code Reader Hands on Session
- Spring Boot Log4j

Spring Data- JPA

- Spring Data JPA Basics
- Spring Data JPA Introduction
- Spring Data JPA Architecture

- Spring Data JPA application.propertiesfile
- JPA Repository
- CRUD Repository
- **Repository, CrudRepository, JpaRepository**
 - EntityManager
 - Repository methods like findByIdAndName, countByIdOrName
 - Exception handling in repository
 - Database configurations in properties file
 - Database and JPA dependency in pom.xml
 - **Different ways of writing JPA methods**
 - Inbuilt methods
 - Custom methods
 - Custom methods with native query
 - JPQL Java Query Language
- Validations and Constraints on columns
- Understanding of schema and tables
- **Mapping Relationship**
 - OneToOne Mapping
 - OneToMany Mapping
 - ManyToOne Mapping
 - ManyToMany Mapping
- Understanding of keys
 - Primary key,
 - Foreign key,
 - Composite key

Units

➤ Introduction

- Unit Testing
- Terminology
- Unit Testing with JUnit
- Installation of JUnit

➤ JUnit

- Introduction to JUnit API

- Test
- Test Case
- Assert
- TestRunner
- TestSuite

- Preparation
- Create a Java class
- Create a JUnit test
- Run your test cases
- Run your test via code

➤ JUnit4

- Static imports with Eclipse
- Annotations
- Assert statements

- Unit testing in Java
- Test coverage
- JUnit assertions
- Initial Steps
- Using @Ignore annotation
- Creating suite tests
- Creating parameterized tests

MicroServices

- Microservices Introduction
- Microservices Architecture
- Microservices Project Structure
- Microservices Frameworks
- Spring Boot with Microservices Basics
- Spring Boot with Microservices Coding Standards
- Spring Boot with Microservices Builder Design Pattern
- Spring Boot with Microservices QR Code Generator

- Application.properties vs application.yml file
- Spring Boot, Hibernate, Maven, MySQL CRUD Application- POSTMAN API
- Spring Boot, Spring Data- JPA, Maven, MySQL CRUD Application- POSTMAN API
- Spring Boot, Spring Data- JdbcTemplate, Maven, MySQL CRUD Application
- Spring Boot, ORM, Microservices with Project Lombok
- Spring Boot, CORS, Spring Data[JdbcTemplate], Microservices CRUD with Project Lombok using React JS
- Creating Multiple Services using Spring Boot
- Logger with Microservices
- Rest Template Implementation
- VO package creation for Multi Model Application
- Set port number and application name for each service
- Service Registry- Eureka Server
- API Gateway for traverse centralize request
- Hystrix DashBoard
- Zipkin
- Sleuth
- Trace Application- System Health
- Spring cloud API Gateway
- Trace Fallback controller
- Trace Services are up/down using controller service
- Trace Dependent Services
- Trace Controller | Methods and Services how many times used/hit
- Advanced Level Research and Development for Microservices
- Microservices HRM/Payroll Application
- Microservices Customer | Product | Payment Order Application

- Spring Boot with Microservices all Advanced Interview Questions & Answers

MySQL

- What is MySQL
 - MySQL History
 - MySQL Features
 - MySQL Data Types
 - Install MySQL
 - Create Database
 - Select Database
 - Drop Database
 - Table & Views
 - CREATE Table
 - ALTER Table
 - TRUNCATE Table
 - DROP Table
 - MySQL Views
- Understanding of keys
 - Primary key
 - Foreign key
 - Composite key
- MySQL Queries
 - INSERT Record
 - UPDATE Record
 - DELETE Record
 - SELECT Record
 - MySQL Clauses
 - MySQL WHERE
 - MySQL DISTINCT
 - MySQL FROM

- MySQL ORDER BY
- MySQL GROUP BY
- MySQL HAVING
- MySQL Conditions
- MySQL AND
- MySQL OR
- MySQL AND OR
- MySQL LIKE
- MySQL IN
- MySQL NOT
- MySQL IS NULL
- MySQL IS NOT NULL
- MySQL BETWEEN
- **MySQL JOINS**
 - MySQL INNER JOIN
 - MySQL LEFT JOIN
 - MySQL RIGHT JOIN
 - MySQL FULL JOIN
- **Aggregate Functions**
 - MySQL count()
 - MySQL sum()
 - MySQL avg()
 - MySQL min()
 - MySQL max()
 - MySQL first()
 - MySQL last()