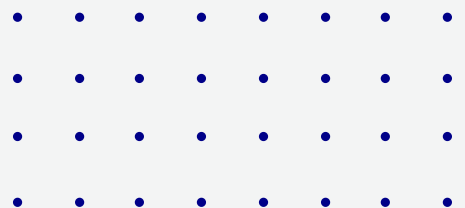
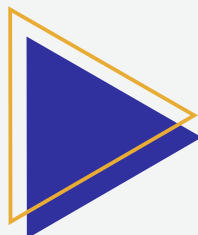


JAVA CERTIFICATION COURSE



Lesson 01**Introduction to Java EE**

- 1.1 Introduction to Java EE

Lesson 02**Java Servlet I**

- 2.1 Servlets API, Interfaces, and Methods
- 2.2 Servlet Lifecycle
- 2.3 Configure and Deploy Servlet
- 2.4 ServletRequest, ServletResponse
- 2.5 ServletConfig, ServletContext
- 2.6 Servlet Scopes, Attributes, and Collaboration

Lesson 03**Java Servlet II**

- 3.1 Session Management
- 3.2 Listeners in Java EE
- 3.3 Filters in Java EE

Lesson 04**Java Server Pages**

- 4.1 JSP Lifecycle
- 4.2 Creating and Working With JSP Elements
- 4.3 Working With JSP Standard Action
- 4.4 JSTL and Custom Tag Libraries

Lesson 05**Introduction to Hibernate**

- 5.1 Introduction to Hibernate
- 5.2 Hibernate CRUD Operation

Lesson 06**Hibernate Queries and Relationship**

- 6.1 Hibernate Queries and Relationships
- 6.2 Mapping Relationship with Hibernate

Lesson 07**Introduction to Spring**

- 7.1 Introduction to Spring
- 7.2 Dependency Injection, SpringBean Lifecycle, Wiring and Scope

Lesson 08**Spring AOP**

- 8.1 Introduction to Spring AOP (Aspect-Oriented Programming)
- 8.2 Configuring AOP in Java Application using AspectJ Approach

Lesson 09**Spring JDBC and Spring Hibernate**

- 9.1 Spring JDBC Implementation in an Application
- 9.2 Spring Hibernate Template
- 9.3 Spring JDBC Transaction Management

Lesson 10**Spring MVC**

- 10.1 Spring MVC Architecture, Components, and Framework
- 10.2 Spring MVC Program

Lesson 05**SOA and Web Services**

- 11.1 Basics of SOA Architecture and Web Services
- 11.2 Creating SOAP based and RESTful Web Services

Core Java**Lesson 01****Introduction to Java 11 and OOPs Concepts**

- 1.01 Course Introduction
- 1.02 Learning Objectives
- 1.03 Introduction
- 1.04 Working of Java program
- 1.05 Object Oriented Programming
- 1.06 Install and Work with Eclipse
- 1.07 Demo - Basic Java Program
- 1.08 Demo - Displaying Content
- 1.09 Basic Elements of Java
- 1.10 Unicode Characters
- 1.11 Variables
- 1.12 Data Types
- 1.13 Operators
- 1.14 Operator (Logical Operator)
- 1.15 Operators Precedence
- 1.16 Type Casting or Type Conversion
- 1.17 Conditional Statements
- 1.18 Conditional Statement (Nested if)
- 1.19 Loops
- 1.20 for vs while vs do while
- 1.21 Access Specifiers
- 1.22 Java Eleven
- 1.23 Null, this, and instanceof Operators
- 1.24 Destructors
- 1.25 Code Refactoring



- 1.26 Garbage Collector
- 1.27 Static Code Analysis
- 1.28 String
- 1.29 Arrays Part One
- 1.30 Arrays Part Two
- 1.31 For – Each Loop
- 1.32 Method Overloading
- 1.33 Command Line Arguments
- 1.34 Parameter Passing Techniques
- 1.35 Types of Parameters
- 1.36 Variable Arguments
- 1.37 Initializer
- 1.38 Demo – String Functions Program
- 1.39 Demo – Quiz Program
- 1.40 Demo – Student Record and Displaying by Registration Number Program
- 1.41 Summary

Lesson 02

Utility Packages and Inheritance

- 2.01 Learning Objectives
- 2.02 Packages in Java
- 2.04 Inheritance in Java
- 2.05 Object Type Casting in Java
- 2.06 Method Overriding in Java
- 2.07 Lambda Expression in Java
- 2.08 Static Variables and Methods
- 2.09 Abstract Classes
- 2.10 Interface in Java
- 2.11 Java Set Interface
- 2.12 Marker Interfaces in Java
- 2.13 Inner Class
- 2.14 Exception Handling in Java
- 2.15 Java Memory Management
- 2.03 Demo – Utility Packages Program
- 2.17 Demo – Bank Account Statement using Inheritance
- 2.18 Demo – House Architecture using Polymorphism Program
- 2.16 Demo – Creating Errors and Catching the Exception Program
- 2.19 Summary

Project 3

Patient Management Application

- 3.01 Learning Objectives
- 3.02 Multithreading
- 3.03 Introduction to Threads
- 3.04 Thread Life Cycle

- 3.05 Thread Priority
- 3.06 Daemon Thread in Java
- 3.07 Thread Scheduling and Sleeping
- 3.08 Thread Synchronization
- 3.09 Wrapper Classes
- 3.10 Autoboxing and Unboxing
- 3.11 java.util and java.lang Classes
- 3.12 java.lang – String Class
- 3.13 java.util – StringBuilder and StringTokenizer Class
- 3.14 java.lang – Math Class
- 3.15 java.util – Locale Class
- 3.16 Java Generics
- 3.17 Collections Framework in Java
- 3.18 Set Interface in Collection
- 3.19 Hashcode() in Collection
- 3.20 List in Collections
- 3.21 Queue in Collections
- 3.22 Comparator Interface in Collections
- 3.23 Deque in Collections
- 3.24 Map in Collections
- 3.25 For – Each Method in Java
- 3.26 Differentiate Collections and Array Class
- 3.27 Input or Output Stream
- 3.28 Java.io.file Class
- 3.29 Byte Stream Hierarchy
- 3.30 CharacterStream Classes
- 3.31 Serialization
- 3.32 JUnit
- 3.33 Logger – log4j
- 3.34 Demo – Creating and Sorting Students Regno using Arrays
- 3.35 Demo – Stack Queue and Linked List Programs
- 3.36 Demo – Multithreading Program
- 3.37 Summary

Project 4

Debugging Concepts

- 4.01 Learning Objectives
- 4.02 Java Debugging Techniques
- 4.03 Tracing and Logging Analysis
- 4.04 Log Levels and Log Analysis
- 4.05 Stack Trace
- 4.06 Logging using log4j
- 4.07 Best Practices of log4j Part – One
- 4.08 Best Practices of log4j Part – Two
- 4.09 log4j Levels
- 4.10 Eclipse Debugging Support





- 4.11 Setting Breakpoints
- 4.12 Stepping Through or Variable Inspection
- 4.13 Demo - Analysis of Reports with Logging
- 4.14 Summary

Lesson 05

Junit

- 5.01 Learning Objectives
- 5.02 Introduction
- 5.03 Unit Testing
- 5.04 JUnit Test Framework
- 5.05 JUnit Test Framework - Annotations
- 5.06 JUnit Test Framework - Assert Class
- 5.07 JUnit Test Framework - Test Suite
- 5.08 JUnit Test Framework - Exceptions Test
- 5.10 Demo - Generating Report using JUnit
- 5.09 Demo - Testing Student Mark System with JUnit
- 5.11 Summary

Lesson 06

Java Cryptographic Extensions

- 6.01 Learning Objectives
- 6.02 Cryptography
- 6.03 Two Types of Authenticators
- 6.04 CHACHA20 Stream Cipher and Poly1305 Authenticator
- 6.05 Example Program
- 6.06 Demo - Cryptographic Program
- 6.07 Summary

Project 07

Design Pattern

- 7.01 Learning Objectives
- 7.02 Introduction of Design Pattern
- 7.03 Types of Design Patterns
- 7.04 Creational Patterns
- 7.05 Factory Method Pattern
- 7.07 Singleton Design Pattern
- 7.08 Builder Pattern
- 7.09 Structural Patterns
- 7.10 Adapter Pattern
- 7.11 Bridge Pattern
- 7.12 Facade Pattern
- 7.13 Flyweight Design Pattern
- 7.14 Behavioral Design Patterns
- 7.15 Strategy Design Pattern
- 7.15 Chain of Responsibility Pattern
- 7.16 Command Design Pattern

- 7.17 Interpreter Design Pattern
- 7.17 Interpreter Design Pattern
- 7.18 Iterator Design Pattern
- 7.19 Mediator Design Pattern
- 7.20 Memento Design Pattern
- 7.21 Null Object Design Pattern
- 7.22 Observer Design Pattern
- 7.23 State Design Pattern
- 7.24 Template Method Design Pattern
- 7.25 Visitor Design Pattern
- 7.26 JEE or J2EE Design Patterns
- 7.27 Demo - Loan Approval Process using One of Behavioural Design Pattern
- 7.06 Demo - Creating Family of Objects using Factory Design Pattern
- 7.28 Demo - State Design Pattern Program
- 7.29 Summary

Industry Project

Project 1

Employee Management System

Create a new system to automate the regulation creation and closure process.

Project 2

Home Insurance

Build a system that helps individuals buy insurance for a home through a web application.



About Edu Eyrie India

Mission:

Edu Eyrie India is driven by a mission to provide quality education and skill development to empower individuals for success in their professional journeys.

Expert Faculty:

Our institution boasts a team of expert instructors who bring industry experience and knowledge to the forefront of our educational programs.

Holistic Learning:

We believe in providing a holistic learning experience, combining technical expertise with soft skill development for well-rounded professionals.

Cutting-edge Curriculum:

Our courses feature cutting-edge curriculum, keeping pace with industry trends and technological advancements.

Innovative Teaching Methods:

Edu Eyrie India employs innovative teaching methods, incorporating practical exercises, projects, and real-world applications to enhance learning outcomes.



+91 731 115 5738
info@edueyrieindia.com
www.edueyrieindia.com
Gurugram, India 122002

eeyrie Edu Eyrie
India
Educate | Elevate | Innovate