

J2EE Syllabus

Overview:

Java is a language and J2EE is a platform which implements java language. J2EE standard for Java 2 Enterprise Edition. Core Java and advanced java are the standard editions of java whereas J2EE is the enterprise edition which is a combination of both core & advanced java. It is used for creating enterprise web applications. J2EE makes use of Servlets and JSPs to provide enterprise applications like web pages and portals.

Course Objectives:

- ❖ To understand the importance of extension JDBC package in Enterprise Java applications.
- ❖ To understand and use the Java Persistence Architecture API for ORM activities (JPA).
- ❖ To implement asynchronous applications and MessageDriven Beans using JMS.
- ❖ To apply Security in Java EE Applications.
- ❖ To learn send/receive mails using Internet protocols SMTP, IMAP and POP3 (JavaMail).
- ❖ To master the whole process of designing, implementing and deploying J2EE Applications.
- ❖ To Understand SOAP, Web Services and Service Oriented Architecture (SOA)
- ❖ .To implement and access Web Service components using EJB in a Java EE application.

Pre-requisite / Target Audience:

This course is designed to meet the needs of Java programmers who want to be professional in building mission-critical enterprise software using EJB. Students should be familiar with programming techniques and have substantial Java programming experience including JDBC, RMI, Servlets and JSP.

Module 1: Remote Method Invocation (RMI)

RMI is mostly used in distributed applications. In this module you will learn how client and server communicate through remote objects.

- ❖ Object Persistence and Serialization
- ❖ Introduction to Distributed Computing
- ❖ RMI Architecture
- ❖ Importance of RMI Registry
- ❖ Developing Simple RMI application
- ❖ Callback Implementation in RMI

Module 2: CORBA

In this module you will learn, how corba is used to establish the communication between distributed objects in

different platforms either by remotely or locally.

- ❖ Introduction to CORBA
- ❖ CORBA for Distributed computing
- ❖ ORB & IIOP
- ❖ Interface Definition Language (IDL)

Module 3: Java EE Application

In this module you will learn how to create a multi-tier architecture and how to distribute each tier in their particular server or machine.

- ❖ Java EE Architecture
- ❖ Introduction to Java EE Components, Containers and Connectors
- ❖ Java EE Modules (Web App, EJB JAR, App Client)
- ❖ Structure of Java EE Application (Enterprise Archive)
- ❖ Packaging and Deploying Java EE Applications

Module 4: JNDI(Java Naming and Directory Interface)

JNDI is an API that provides naming and directory functionality to the application. In this module you will learn how to use JNDI Services in your application.

- ❖ Introduction to Naming Services
- ❖ JNDI as Java API to Naming Services
- ❖ Using JNDI

Module 5: JDBC Extension

In this module you will learn the new features of jdbc and how to send multiple connection objects to resource pool and how it is shared among different clients.

- ❖ javax.sql package (Extension to JDBC)
- ❖ DataSource and Connection Pool
- ❖ Using JDBC and JNDI

Module 6: Enterprise JavaBeans [EJB 3]

In this module you will learn how ejb become a part of the multi layered architecture. And how it will interact with the database.

- ❖ Introduction to Server-Side Components
- ❖ EJB Design Goals and Roles

- ❖ EJB Architecture
- ❖ Simplified EJB 3 API
- ❖ Metadata Annotations in place of XML
- ❖ RMI over IIOP

Types of EJB

- ❖ Session Bean
- ❖ MessageDriven Bean
- ❖ Entity Bean

EJB Container Services

- ❖ Transactions
- ❖ Security
- ❖ Life Cycle Management
- ❖ State and Persistence of EJB

Session Beans

- ❖ Role of Business Interface
- ❖ Remote and Local Interfaces
- ❖ Session Bean Lifetime
- ❖ Developing Stateless Beans
- ❖ Developing Stateful Beans
- ❖ Standalone and Web Clients

Module 7: Java Persistence API (JPA)

Jpa provides a framework which helps us to store a vast amount of data into a database there is no need to write relational models.

- ❖ Designing Persistent Class
- ❖ Entity Fields and Properties
- ❖ Entity Instance Creation
- ❖ Primary Keys and Entity Identity
- ❖ Entity Relationships
- ❖ Entity Operations
- ❖ Entity Manager
- ❖ Entity Instance Life Cycle
- ❖ Persistence Context
- ❖ Query API

- ❖ Query Language

Module 8: Java Transaction Management (JTA)

In this module you will learn how to perform series of operations on different data sources simultaneously.

- ❖ The ACID Test for Transactions
- ❖ Introduction to JTS & JTA
- ❖ Container-Managed Transactions
- ❖ Bean-Managed Transactions
- ❖ Transaction Attributes
- ❖ Using JTA

Module 9: Security Model

In this module you will learn how to provide security for different layers present in the web server.

- ❖ Role-Driven Access Control
- ❖ Security Identity
- ❖ Security and the Deployment Descriptor
- ❖ Using Security Roles
- ❖ Accessing Security Information via EJBContext
- ❖ Using JAAS to access secure EJB

Module 10: Java Message Service (JMS)

In this module you will learn how to create/send/read messages from one application to another.

Introduction to Messaging Systems.

- ❖ Benefits of using JMS
- ❖ Pub/Sub Model
- ❖ Point-to-Point Model
- ❖ Message Formats, Headers & Properties
- ❖ How JMS fits into EJB system
- ❖ Developing Message Driven Beans (MDB)

Module 11: Web Services

In this module you will learn how web services will make different applications to communicate over the internet/to convert standalone application to web application.

- ❖ Introduction to Web Services

- ❖ Service Oriented Architecture
- ❖ Architecture and Advantages
- ❖ SOAP Significance
- ❖ WSDL Importance
- ❖ Web Service Annotations
- ❖ Implementing a Web Service
- ❖ Java API for XML Web Services (JAX-WS)
- ❖ Writing a Web Service Client

Real-time Project involving most of the above concepts with following will be provided

- Product Abstract Document
- Requirement Specification Document
- **Step-by-Step procedure for building the project from ground up by using IDE.**
- Complete Source Code

At the end of the course participants will get the knowledge of:

Creating your own distributed web application and you can able to know how to utilize the various available resources without need of other's help.