

J2EE (2008)

(JAVA 2 ENTERPRISE EDITION) New

(includes JSP, SERVLETS, JDBC, JAVA BEANS, EJB, STRUTS, RMI, SPRING, DESIGN PATTERNS, ECLIPSE, HIBERNATE, HQL, JBOSS, JAVA MAIL)

BLOCK 1 : J2EE

DURATION : 2 MONTHS

Core Java	<p>Course Outline</p> <ul style="list-style-type: none"> • Overview • User-defined Data Types • Methods • Applications • Applications vs. Applets • Comment Syntax • Variable Definition and Assignment • Creating Class Instances • Data Types • Method Overloading • Instance Variables • Access Methods • Instance Methods • Conditional Execution • Methods that Return Values • Access Methods Revisited • Iterative Execution • Multifunction Operators • Strings • Reference Variable Usage • Default Variable Initializations • Arrays • Equality • Expressiveness • Garbage Collection • Run-time Environments and Class Path Settings • Decompiling • Applets <p>MULTITHREADED PROGRAMMING The Java Thread Model Creating a Thread: Extending Thread, Implementing Runnable Creating Multiple Threads and Context Switching</p>
-----------	---

	<p>Synchronization: Methods and Statement Interthread Communication Introduction to I/O streams File Handling Binary Streams Character Streams Data Streams Serialization</p> <ul style="list-style-type: none"> • Externalization
<p>JDBC</p>	<ul style="list-style-type: none"> • Introduction to JDBC <ul style="list-style-type: none"> ○ SQL ○ ODBC ○ The Java Programming Language and JDBC • A Complete Example <ul style="list-style-type: none"> ○ Describing the Scenario ○ Creating a Database ○ Connecting to the Database ○ Creating a Table ○ Inserting Information into a Database ○ Step by Step ○ Retrieving Information from a Database <ul style="list-style-type: none"> ▪ Data Navigation ▪ Data Extraction • Connecting a Java Program to a Database <ul style="list-style-type: none"> ○ Areas Controlled by the Connection Interface ○ Generalizing Connection Information • Statements, ResultSets, and Interacting with a Database <ul style="list-style-type: none"> ○ Modifying Data ○ Database Queries • Prepared Statements • Java-SQL Type Equivalence • Metadata <ul style="list-style-type: none"> ○ Database Metadata ○ ResultSet Metadata • Stored Procedures <ul style="list-style-type: none"> ○ MetaData Support ○ Parameter INs and OUTs ○ Escape Syntax ○ CallableStatement ○ Setup, Invocation, and Value Retrieval • Transactions <ul style="list-style-type: none"> ○ Commit ○ Rollback

	<ul style="list-style-type: none"> ○ Concurrency ○ Typical Transaction Code <p>Batch Update Facility</p>
JSP	<p>Course Outline</p> <ul style="list-style-type: none"> • Introduction <ul style="list-style-type: none"> ○ JSP Advantages ○ Comparing JSP with ASP ○ JSP or Servlets? • JSP Architecture • JSP Access Models • JSP Syntax Basics <ul style="list-style-type: none"> ○ Directives <ul style="list-style-type: none"> ▪ Page Directive ▪ Include Directive ○ Declarations ○ Expressions ○ Scriptlets ○ Comments • Object Scopes • JSP Implicit Objects • Synchronization Issues • Exception Handling • Session Management • Standard Actions <ul style="list-style-type: none"> ○ Using JavaBean Components ○ Forwarding Requests <ul style="list-style-type: none"> ▪ Request Chaining ○ Including Requests
SERVLET	<p>Course Outline</p> <p>SERVLET BASICS</p> <p>Browsers, Servers and Servlets The Basic Servlet The Servlet Life Cycle The HttpServlet Approach More do Methods Threading in Servlets Debugging</p>

	<p>REQUEST AND RESPONSE</p> <p>Request and Response Basics The HttpServletRequest Object Request Headers Status Codes Response Headers Ensuring Valid Characters</p> <p>SESSION TRACKING</p> <p>Understanding Cookies The Cookie Class Cookies in JSP Cookie Properties Session Tracking The HttpSession Class Sessions in JSP Encoding URLs Terminating Sessions</p>
<p>SWING</p>	<p>Building Java GUIs</p> <ul style="list-style-type: none"> • Abstract Window Toolkit • The javaawt Package • Containers • Positioning Components • Frames • The FrameExample Class • Example Frame • Panels • The FrameWithPanel Class • Layout Managers • Default Layout Managers • A Simple FlowLayout Example • The FlowLayout Manager • The FlowExample Class • The BorderLayout Manager • Organization of the Border Layout Components • The BorderExample Class • Example of BorderLayout • The GridLayout Manager • The GridExample Class • Example of GridLayout

	<ul style="list-style-type: none"> • The ComplexLayoutExample Class • Drawing in SWING • Other important SWING Components
RMI	<p>Course Outline</p> <ul style="list-style-type: none"> • Introduction to Distributed Computing with RMI <ul style="list-style-type: none"> ○ Goals ○ Comparison of Distributed and Nondistributed Java Programs • Java RMI Architecture <ul style="list-style-type: none"> ○ Interfaces: The Heart of RMI ○ RMI Architecture Layers <ul style="list-style-type: none"> ▪ Stub and Skeleton Layer ▪ Remote Reference Layer ▪ Transport Layer • Naming Remote Objects • Using RMI <ul style="list-style-type: none"> ○ Interfaces ○ Implementation ○ Stubs and Skeletons ○ ○ Running the RMI System
JAVA BEANS	<p>FORMS AND JAVABEANS</p> <p>HTML Forms JavaBeans JavaBeans and JSP JavaBeans and SERVLET Bean Properties Property Types Properties and Forms Bean Scopes</p>
EJB	<p>EJB AND THE J2EE ARCHITECTURE</p> <p>Evolution of Distributed Computing on the Web The J2EE Solution The Enterprise JavaBean Roles in Enterprise JavaBeans Development EJB Container and Application Server Web Services and J2EE</p>

	<p>GETTING STARTED WITH EJB</p> <ul style="list-style-type: none"> Defining the Bean Class Remote Interface Writing Business Methods Home Interface Deployment Descriptors and Deployment The Client Locating the Bean Create an Enterprise Bean Instance Invoking the Bean's Methods Compiling and Running the Client <p>THREE TYPES OF EJB</p> <ul style="list-style-type: none"> A Session Bean A Message-Driven Bean An Entity Bean What About State? Stateless Session Beans Stateful Session Beans MDB Code Entity Bean Persistence Models Entity Bean Code Deployment Descriptor
WEBLOGIC 8.1	<p>WEBLOGIC OVERVIEW</p> <ul style="list-style-type: none"> • Architecture • Start & Stopping WebLogic • Administration Console • Application Deployment
STRUTS	<p>Developing Web Applications Using Struts</p> <ul style="list-style-type: none"> • Model-View-Controller Pattern • Struts MVC Framework • Front Controller Pattern • Struts MVC Framework • Struts Activity Diagram • Struts Action Class • Struts Formbean • Configuration file • Configuring the Struts Action Mappings • Configuring the Infrastructure Controller

	<ul style="list-style-type: none"> • Front Controller Servlet Mapping • Configuring Action Mappings • Action Mapping Object Representation • Physical Web Hierarchy
JAVA MAIL	<p>Java Mail</p> <ul style="list-style-type: none"> • Mail Systems and JavaMail • The javax.mail Packages • Establishing a Session • The Message Interface • Sending a Message • Message Stores • Mail Folders • Multipart Messages
JNDI	<p>JNDI</p> <ul style="list-style-type: none"> • Naming and Directory Services • Namespaces and Contexts • Naming Operations • Bindings • Attributes • Directory Operations • DNS Lookups with JNDI • JNDI in J2EE
JBOSS	<p>JBoss</p> <ul style="list-style-type: none"> • JBoss Introduction <ul style="list-style-type: none"> ○ JBoss Family ○ JBoss Application Server ○ Hibernate ○ JBoss Portal ○ JBoss jBPM ○ JBoss Eclipse IDE • Installing JBoss <ul style="list-style-type: none"> ○ Installation Prerequisites ○ Other Common Components ○ JBoss Installation ○ JBoss Installer ○ Installing in Production ○ Starting JBoss ○ JBoss Admin Consoles ○ JBoss Configurations • Managing JBoss Configurations <ul style="list-style-type: none"> ○ JBoss Configurations ○ Controlling JBoss Servers

<p>USING GANG OF FOUR CREATIONAL PATTERNS</p>	<p>Module 3 - Using Gang of Four Creational Patterns</p> <ul style="list-style-type: none"> • Describe the basic characteristics of the Creational patterns • Apply the Factory Method pattern • Apply the Abstract Factory pattern • Apply the Singleton pattern •
<p>USING GANG OF FOUR STRUCTURAL PATTERNS</p>	<p>Module 4 - Using Gang of Four Structural Patterns</p> <ul style="list-style-type: none"> • Describe the basic characteristics of the Structural patterns • Apply the Facade pattern • Apply the Proxy pattern • Apply the Adapter pattern • Apply the Composite pattern • Apply the Decorator pattern

BLOCK 2 : HIBERNATE & SPRINGS

DURATION : 1 MONTH

<p>Spring</p> <ul style="list-style-type: none"> • Overview • Core Techniques 	<p>Introduction</p> <ul style="list-style-type: none"> • History of Spring • How Spring Works <p>The BeanFactory & How it</p> <ul style="list-style-type: none"> • About the Bean Factory • A Bean's LifeCycle in The BeanFactory • IOC / Dependency Injection • The Bean Definition Exposed • Configuring Properties & Dependencies • Pre-Initializing Your Beans • Ready State • Destroying Beans • The ApplicationContext • Internationalization & Message Source • Practical of Related Concepts
<p>SPRING</p>	<p>AOP</p>

	<ul style="list-style-type: none"> ○ AOP Concepts ○ Pointcuts in Spring ○ Advice types in Spring ○ Practical of Related Concepts <p>Getting started with Spring MVC</p> <ul style="list-style-type: none"> ○ The DispatcherServlet ○ Controllers ○ Handler Mappings ○ ViewResolver ○ HandlingException ○ Practical of Related Concepts
SPRING	<p>Spring in Business Layer</p> <ul style="list-style-type: none"> ○ Spring DAO ○ Using JDBC with Spring ○ Using JdbcTemplate ○ Developing CRUD application with JdbcDao
SPRING	<p>Data Access using OR Mappers</p> <ul style="list-style-type: none"> ○ Integrating Hibernate with Spring ○ Hibernate overview ○ Managing Hibernate Resource ○ Inversion of Control : Template ○ Accessing Hibernate through HibernateTemplate <p>Practical of Related Concepts</p>
<ul style="list-style-type: none"> • Introduction to Hibernate • Configuring Hibernate 	<ul style="list-style-type: none"> • Introduction to Hibernate <ul style="list-style-type: none"> ○ JDBC persistence model ○ Plain Old Java Objects (POJOs) ○ Object-Relational Mapping issues ○ The Object/Relational Mapping Problem ○ The Hibernate Alternative ○ Hibernate Architecture and API • Configuring Hibernate <ul style="list-style-type: none"> ○ Hibernate Distribution ○ Required Libraries ○ Hibernate configuration files

	<ul style="list-style-type: none"> ○ Using the Hibernate properties file ○ Using the XML configuration file <p>Selecting a SQL dialect</p>
<ul style="list-style-type: none"> • Hibernate Persistence • Fundamentals of Object Relational Mapping 	<ul style="list-style-type: none"> • Hibernate Persistence <ul style="list-style-type: none"> ○ ID Generators ○ Building the SessionFactory ○ Creating a Session ○ Transactions ○ Implementing CRUD methods ○ The Data Access Object (DAO) deign pattern ○ Hibernate and the Data Transfer Object (Anti-)Pattern • Fundamentals of Object Relational Mapping <ul style="list-style-type: none"> ○ Basic O/R Mapping ○ Modeling Composition with Relationships ○ Modeling Composition using Components ○ One-to-one Relationships ○ Many-to-one Relationships ○ Many-to-many Relationships ○ Unidirectional and Bidirectional Associations <p>Modeling Inheritance</p>
<ul style="list-style-type: none"> • The Criteria Query API • Hibernate Query Language (HQL) 	<ul style="list-style-type: none"> • Hibernate Query Language (HQL) <ul style="list-style-type: none"> ○ Building simple queries ○ Using named parameters ○ Defining named queries ○ Sorting ○ Associations and Aggregates ○ Using native SQL • Caching Support <ul style="list-style-type: none"> ○ Caching overview ○ Caching Architecture ○ API to control cache ○ Working with EHCACHE ○ Using EHCACHE to enhance performance ○ Setting up a distributed cache cluster for enterprise applications

	<ul style="list-style-type: none">• Transaction support<ul style="list-style-type: none">○ Local transactions, global transaction○ Transaction API○ Isolation levels○ Using Transaction API○ Optimistic and Pessimistic locking (versioning)○ Implementing optimistic locking with versioning
--	---

Block : 1

Program Duration : 2 Months

Program Fee : Rs.14000/-

Block : 2

Program Duration : 1 Month

Program Fee : Rs.8000/-

Weekdays or Weekends option available.

Program duration can be extended or reduced.