# Java

CORE JAVA

| Core Java Programing (Course Duration: 40 Hours) | |
|---|---|
| **Introduction to Java**<br>• What is Java?<br>• Why should we use Java?<br>• Java Platform Architecture<br>• Java Virtual Machine<br>• Java Runtime Environment<br>• A Simple Java Program | **Introduction to Eclipse and Net beans IDE**<br>• Starting the Eclipse IDE<br>• Using a workspace<br>• Creating a project<br>• Creating a Java package<br>• Creating a Java Class<br>• Executing a Java program<br>• Debugging code in Eclipse<br>• Updates and Installation of Plug-ins |
| **Expressions and Flow Control**<br>• Distinguish between instance and local variables<br>• Describe how to initialize instance variables<br>• Recognize, describe, and use Java software operators<br>• Casting | **Identifiers, Keywords, and Types**<br>· Use comments in a source program<br>· Distinguish between valid and invalid identifiers<br>· Recognize Java technology keywords<br>· List the eight primitive types<br>Define literal values for numeric and textual types<br>Define the terms primitive variable and reference variable<br>· Declare variables of class type<br>· Construct an object using new<br>· The "this" reference |
| **Object Oriented Concepts**<br>• OO Concepts<br>• Abstraction<br>• Encapsulation<br>• Inheritance<br>• Polymorphism<br>• Classes and Objects<br>• Attributes<br>• Methods<br>• Constructors<br>• Packages | **Class Design**<br>• Define inheritance, polymorphism, overloading, overriding, and virtual method invocation<br>• Use the access modifiers protected and the default(package-friendly)<br>• Describe the concepts of constructor and method overloading<br>• Describe the complete object construction and initialization operation |
| **Object Life Cycle**<br>• Object Creation<br>• Garbage Collection<br>• Finalize method<br>• | **Type of Classes**<br>• Nested Classes<br>• Inner Classes<br>• Local and Anonymous Inner Classes |
| **Interfaces**<br>• Defining an Interface<br>• The Interface Body<br>• Using an Interface as a Type<br>• Extending Interfaces | **String Handling**<br>• Methods of String Class<br>• Comparing Strings<br>• Immutable String<br>• String Buffer Class<br><br>**Exception**<br>• Define exceptions<br>• Use try, catch, and finally statements<br>• Describe exception categories |

| | |
|---|---|
| | • Identify common exceptions |
| **Input/output Streams**<br>• Overview of Streams<br>• Bytes vs. Characters<br>• Converting Byte Streams to Character Streams<br>• File Object<br>• Binary Input and Output<br>• Print Writer Class<br>• Reading and Writing Objects<br>• Basic and Filtered Streams<br>• File Class<br>• Scanner Class | **Internalization and Property class**<br>• Internalizing application<br>• Date Time formatter<br>• Number Formatter<br>• Properties class<br>• Command Line Arguments |
| Multithreading in Java<br>• Non-Threaded Applications<br>• Threaded Applications<br>• Creating Threads<br>• Thread States<br>• Runnable Threads<br>• Coordinating Threads<br>• Interrupting Threads<br>• Runnable Interface<br>• Synchronizing Threads<br>• Interthread Communication<br>• Thread Groups<br>• Advanced Locking Concepts | Collection Framework and Generics<br>• The Collection Framework<br>• The Set Interface<br>• Set Implementation Classes<br>• The List Interface<br>• List Implementation Classes<br>• The Map Interface<br>• Map Implementation Classes<br>• Sorting with Comparator<br>• Sorting Lists and Arrays<br>• Collections Utility Methods |
| Introduction to standalone application and Applet<br>• Introduction to AWT<br>• Introduction to Swing<br>• Garbage Collection in Java | Introduction to JDBC and SQL<br>• The JDBC Connectivity Model<br>• JDBC Driver types<br>• Loading drivers and Connecting to DB<br>• Creating a SQL Query<br>• Getting the Results<br>• Updating Database Data<br>• Database Metadata<br>• Finishing Up |
| **Socket Programming and RMI** | |

+

**Web Programming using Servlets and JSP**

| **Web Fundamentals** | **Servlets** |
|---|---|
| • HTTP Protocol<br>• HTTP Headers<br>• GET & Post Methods<br>• Web Client<br>• Web Server<br>• Web Container<br>• Web Component<br>• Servlet & JSP Engine | • Servlets Architecture<br>• Servlet lifecycle<br>• Servlet interface<br>• Generic Servlet & HttpServlet<br>• Writing your own Servlet<br>• Packaging and Deploying a Servlet<br>• Servlet & DB connectivity |
| **Servlet Collaboration** | **JSP** |
| • Forwarding & Redirecting from a Servlet<br>• Servlet Context & ServletConfig<br>• Maintaining Sessions<br>• Hidden Form Fields, Cookies, URL rewriting, HttpSession<br>• Filters and Listeners | • Need of JSP<br>• Directives<br>• Scripting Elements<br>• Standard Actions<br>• Implicit Objects<br>• Scopes |

| **JSTL** | **Custom Tags** |
|---|---|
| • Introduction to JSTL<br>• Core tags | • taglib directive<br>• Tag Handler, Tag Support<br>• Iterator Tag |
| **Struts Framework 2.0** | |

| Getting Started with Struts 2.0 | Validation and Interceptors |
|---|---|
| • MVC and Struts<br>Review Filters, Servlets, JSP and Web Applications<br>Struts Architecture<br>Struts Components: Filter Dispatcher, Interceptors, Actions, and Results | • Implementing validation<br>Type conversion<br>Interceptor role<br>Using interceptors |
| Mapping the Model to the View | Working with the Tiles Custom Library (Optional) |

| | |
|---|---|
| • Introducing Struts Tags<br>Generic Tags<br>UI Tags<br>Themes and Templates<br>Tag Reference<br>Ajax Tags<br>Value Stack technology<br>OGNL for type conversion | • Defining templates<br>Using put and insert<br>Advanced tile features (optional lesson) |
| **Hibernate** | |

| **The Object-Relational Divide** | **Mapping Persistent Classes** |
|---|---|
| • Relationship between SQL, JDBC and objects<br>Bridging the chasm: issues with OO and databases | • **From POJO to database**<br>**XML-style mapping**<br>**Annotation-style mapping** |
| **The Hibernate Environment** | **Advanced Mapping** |
| • The session manager<br>• Transitive persistence<br>• Dealing with long-running transactions<br>• Locking implications | • Mapping to collection classes<br>Managing multiplicity: one-to-many, many-to-many<br>Custom mapping types |

**JSF 2.0**

| **JSF Overview** | **GUI Components 1** |
|---|---|
| • Model View Controller<br>• JSF elements - web.xml and faces-config.xml<br>• The power of Facelets<br>  ○ tile-like composition<br>  ○ debugging/error handling<br>  ○ custom logic tags and expressions<br>  ○ XHTML<br>• JSF 2.1 - quick summary of features | • Component Families<br>• UIViewRoot and the FacesContext<br>• The JSF lifecycle<br>  ○ GET vs. POST(JSF 2.1)<br>• Facelets pages<br>• Output components - text and labels<br>• Forms<br>• Formatting output<br>• Inputting text and passwords<br>• Messages and severity |

| | |
|---|---|
| **JSF Events** | **GUI Components 2** |
| • The phases of JSF<br>  ○ Restore View<br>  ○ Apply Request Values<br>  ○ Process Validations<br>  ○ Update Model Values<br>  ○ Invoke Application<br>  ○ Render Response<br>• GET vs. POST<br>  ○ creating the view tree<br>  ○ rendering the existing view tree<br>• | • Selecting items from lists<br>• Checkboxes and radio buttons<br>• Command Buttons and Command links<br>• Panel Group and Panel Grid<br>• Data Tables<br>  ○ display-only data tables<br>  ○ "interactive" data tables<br>• Writing Custom Components<br>  ○ The makeup of a Custom Component<br>  ▪ JSF 1.1 and 1.2<br>  ▪ JSF 2.1<br>  ○ Using the component in a JSP<br>  ○ Using attributes with a custom tag<br>  ○ Custom components that provide input |

**EJB 2.0 and 3.0**

1. Introduction of EJB .
2. Creating Application using Session Bean(Stateless Session Bean and Stateful Session Bean).
3. Creating Application using Entity Bean.(Bean Managed Persistence and Container Managed Persistance).
4. MDB Message Driven Bean Using JMS(Java Messaging Services)