

JAVA-SE (Java Standard Edition)

- All about Networks
- Introduction to Java
- First Step Towards Java Programming
- Naming Conventions and Data Types
- Operators in java
- Control Statements in Java
- Input and Output
- Arrays
- Strings
- String Buffer and String Builder
- Introduction to OOPs
- Classes and Objects
- Inheritance
- Polymorphism
- Type Casting
- Abstract Classes
- Interfaces
- Packages
- Exception Handling
- Wrapper Classes
- The Collection framework
- Streams and Files
- Threads
- Generic Types
- Garbage Collector
- I18N-Internationalization
- Inner Classes

JDBC

- Introduction to JDBC
- JDBC architecture
- Java.sql package
- Connection ,Statement , ResultSet
- Prepared Statement
- Callable Statement
- Scrollable and Updatable ResultSet
- Batch Updates
- Result Set Metadata
- Simple Transaction Management
- Four Levels of JDBC drivers, their pros & cons

JAVA-EE SERVLETS (2.x)

- Need of Server Side Programming
- Introduction to Servlets

- Servlet Life Cycle
- Javax.servlet package
- ServletConfig, ServletContext, ServletResponse
- Supplying Initialization parameters to Servlets
- Performing database operations in Servlets
- Include and forward mechanisms
- Applying filters to Servlets
- Javax.servlet.http Package
- HttpServlet Life Cycle
- Http request methods GET Vs POST
- HttpServletRequest, HttpServletResponse
- Dealing with Http headers & error codes
- Session Tracking, purpose
- Hidden form fields, Cookies
- Http Session, URL rewriting
- Event listeners
- Web application security

JSP

- Disadvantage of Servlets
- Introduction to JSP
- JSP Life Cycle
- Creating dynamic Web content with JSP
- Scripting elements
- Scriplet
- Declaration
- Expression
- XML syntax for JSP elements
- JSP directives page, include and taglib
- JSP implicit objects
- JSP scopes
- Include and forward mechanism
- Using a java bean in a jsp
- JSP Model 1 architecture
- JSP Model 2 (MVC) architecture
- Custom Tag Development
- Error Handling in a JSP
- JSTL

FRAMEWORKS

Future Q Technologies

Struts Framework 1.3/2.0(Web Application Framework)

- Difference between Servlets and Struts
- MVC introduction
- MVC2 introduction
- Struts Architecture
- Simple application
- Internationalization and Message support in struts
- Struts Controller components (ActionServlet , RequestDispatcher,ModuleConfig, ActionMapping,ActionForward)
- Struts user defined components(Action Form, Action Class)
- **User defined extension classes**
 - DynaActionForm
 - DynaValidatorForm
 - Validator-rules.xml
 - Validation.xml
- **Dispatch Action EventDispatchAction o LookupDispatchAction o appingDispatchAction**
- Forward Action
- Include Action
- Switch Action
- Download Action
- **Tiles(Layout management)**
 - Simple Layout
 - Tiles definitions
 - Tile-defs.xml
- Tiles Inheritance
- Security in struts

Hibernate 3.2 (ORM Framework)

- Difference between JDBC and ORM
- Evolution of Hibernate
- Hibernate Architecture
- Basic application on hibernate
- Hibernate-cfg.xml
- Hibernate-config.xml file
- Connection pool configuration in Hibernate
- Performing INSERT/UPDATE/FIND/DELETE operations in Hibernate

- Primary Key generation techniques in Hibernate
- **Storing Inheritance objects into DB using Hibernate**
 - Table per class
 - Table per sub class
 - Table per concrete class
 - Storing association objects into DB
 - One-One relationship
 - One-Many relationship
 - With List
 - With Bag
 - With Set
- Many-One relationship
- Many-Many relationship
- Caching in Hibernate
- Writing queries in

Hibernate using HQL

- Writing Hibernate Queries using Criteria and Criterion Execution predefined queries in Hibernate
- ECaching in Hibernate
- Writing queries in Hibernate using HQL
- Writing Hibernate Queries using Criteria and Criterion
- Execution predefined queries in Hibernate
- Execution of Stored procedures from Hibernate
- Executing native SQL operations using Hibernate
- Transaction management in Hibernate

Spring 2.x

- How to write light weight business components in spring
- How to write reusable business logic components in spring
- How to achieve them using spring IOC concept
- How spring Container is light weight
- How to add middleware services to business components in spring using AOP
- First example in spring using IOC

- Spring Core Module(1 st module)
- Spring Context Module(6 th module)Bean Factory
- Application Context
- Spring JDBC & DAO Module (2 nd module) jdbc Template, jdbc DaoSupport, Callback interfaces.
- Spring ORM Module(3 rd module)
- HibernateTemplate,Hibern ate DaoSupport,Callback interfaces
- Spring Web Module(4th module)
- Spring Web MVC Module(5 th module)
- Integration with Struts
- Spring AOP Module(7 th module)
- Target object
- roxyFactoryBean
- AOP Proxy
- **Advice**
- MethodBeforeAdvice
- AfterReturningAdvice
- Around Advice(Meth odInterc emptor)
- Throws Advice
- **Pointcuts**

StaticMethodPointcut

Dynamic Method Point cut

RegExpr Method Matcher Point cut

- **Advisors**

Résumé Preparation & Placement Assistance