Core Java

Class 1:

- Java & JVM and It's futures
- Programming fundamental (Data Types, Arrays, Operators, Arrays and Flow Control statements if, else, switch, while, do while, for and for each.)

Class 2:

- Variables
- Methods
- Class
- Object
- Coding standards
- Main methods
- packages
- imports
- Access specifies (Public, Private, protected and default)

Class 3:

- Inheritance
- Abstraction
- Encapsulation
- Polymorphism (Method Over loading and Method Overriding)
- Constructors

Class 4:

- Abstract class
- Interfaces
- var-args
- Inner classes
- super and this key words
- final key word

Class 5:

- Object class
- Garbage Collector
- String

- String buffer and String builder
- enum
- Wrapper classes,
- Auto boxing, Auto unboxing
- Object Type casting

Class 6:

- Exception overview
- Default Exception Handler
- Handling Exception using try and catch
- finally
- throws
- throw
- Custom Exceptions

Class 7:

- Threads overview
- Thread creation
- Thread Life cycle
- Threads class methods (priority, yeild, join)
- Synchronization
- Intra thread communication (Wait, notify methods)

Class 8:

- Collection framework interfaces List and set overview
- List implemented classes (ArrayList,Vector...)
- Set implemented Classes (HashSet,TreeSet...)
- Iterator , ListIterator and for each

Class 9:

- Comparator and Comparable interface
- Map implemented classes (HashMap,Hashtable....)
- Collections class
- Generics

Class 10:

- Create File , Directory..
- Read and Write file data
- I/O Streams
- Serialization

JDBC

Class 11:

- JDBC API
- JDBC architecture
- Driver classes
- Connection object
- Statement, PreparedStatemnt.. class

Class 12:

- Call the procedure and functions using Callable Statement
- Result Set
- setAutoCommit, commit and rollback methods in jdbc
- Connection pooling

Servlets & JSP

Class 13:

- Web Application
- HTTP protocol
- Servelt API
- Tomcat, Web Logic Servers
- XML
- Servlet application folder structure
- Servlet architecture
- Server request and response processing
- Servlet Lifecycle

Class 14:

- ServeItConfig and ServeItContext interface
- RequestDispatcher interface
- Different tags in Web.xml
- Listeners
- Filters

Class 15:

- JSP architecture
- JSP lifecycle
- Implicit objects in jsp
- JSP Elements

Class 16:

- Session Management
- Cookies
- JSP Expression Language

Class 17:

- JSP Scopes
- JSTL tags
- Design JSP using HTML & JS

Struts

Class 18:

- MVC
- Struts architecture
- Actions
- Form Beans
- Struts flow

Class 19:

- Form Data Binding(Action Form, ValidatorForm...)
- Validations using validation framework
- Client side validations
- Server side validations

Class 20:

- Internalization
- Global exceptions
- Global Forwards

Hibernate

Class 21:

- ORM
- Hibernate architecture
- Hibernate overview
- Cfg file
- Mapping file

Class 22:

- Relations (one to one, one to many....)
- CRUD operation using Hibernate
- First level cache and Second level cache

Class 23:

- Id generators
- HQL
- Criteria API
- Annotations

Spring

Class 24:

- What is Spring Framework, Spring Introduction
- Quick Steps To Developing Spring Applications
- Dependency Injection In Spring Framework
- Setter Injection
- Constructor Injection
- Different tags in Spring configuration file with examples

Class 25:

- Spring JDBC Introduction
- Importance of the Spring JdbcTemplate
- Execute the select queries using spring jdbctemplate
- Execute the non select queries using spring jdbctemplate

Class 26:

- Spring ORM Introduction
- Importance of the Spring Hibernate Template
- Execute the select queries using spring hibernate Template
- Execute the non select queries using spring hibernate Template

Class 27:

- Spring AOP Introduction
- Spring aop terminology
- Spring advices (Before advice, After advice, around advice and throw advice) with examples
- Spring Pointcuts(NameMatchMethodPointcut,RegularExpressionMethodPointcut)

Class 28:

- Spring AOP example with xml tag based
- Spring AOP example with annotation based
- Data Base transaction perform using the Spring AOP

Class 29:

- Spring MVC Introduction
- Dispatcher Servlet
- Spring handler mapping
- Spring Controllers

Class 30:

- How to perform form databinding using Spring MVC
- How to perform the validations using Spring MVC
- Views in Spring MVC
- View resolvers in Spring MVC

Class 31:

- Singleton
- DAO
- DTO
- Front controller design pattern
- Factory pattern
- Abstract factory pattern
- Adapter pattern

JMS

Class 32:

- Weblogic

Class 33:

- JMS Introduction
- Queue

Class 34:

- Topics

WebServices

Class 35:

- Web Services Overview
- Develop web Service
- WSDL
- SOAP
- Develop a Web Service Client

Class 36:

- REST Web Services Overview
- Develop REST web Service
- Develop a REST Web Service Client

Real Time Tools

Class 37:

- Real-time Tools
- Sample Project
- ANT

Class 38:

- Meven
- Log4j
- SVN
- Tomcat & web Logic server

Class 39:

- Sample Project explanation

Key Features

- In every class write the example and show it practically with well explanation.
- We are explaining and develop the each topic with taking the some real time project scenarios. (Just like develop a mini project with those topics)
- Total Course - 40 to 45 classes
- We will share java all softwares.
- Each class has taken minimum of 1 hours to 1:30 hours.
- We will send the examples to the Students after completion of each class and ask them to practice the same.
- We will give the Assignments and Interview Questions to the students after completion of each topic.
- We will monitor the each Students assignment.

Example Assignment Programs look likes:

1. Create the Student class with following properties:

Student Id, Student name,age, Student class-name. Write a constructor to initialize the properties with default values.

Define a new main class to do the following operations

1. Initialize list of Students with different values.
2. Split the list of students into Map with class-name as key and list of students is value.
3. Define a method to take class-name as argument and display the list of students belongs to the class-name.

2. Create Employee Class and add few properties to it, like employee Id, employee name, salary and date of birth, implements getters and setters. Write a constructor to initialize instance variable employee Id, employee name, salary and date of birth.

1. Write a one main method class which contains list of Employee objects. Some of the employee ids are same in that list.
2. Write logic to eliminate the duplicate employees from that list and print it as some sorting order based on the Employee number.