CORE JAVA

Saying “Hello” to Java: A primer on Java Programming language

- *Intro to Java & its features*
- *Why Java very famous?*
- *Types of applications that can be developed using Java*
- *Writing my first Java program*
- *Java Programming structure*
- *Compiling & running a program*
- *Java Internals (JRE, JVM & JDK + environmental setup)*

Constructing the nuts & bolts of Java

- *Class & objects*
- *Variables & its types*
- *Defining methods – main & other*
- *Access specifiers*
- *Operators & statements*
- *Constructors*
- *Calling methods with/without arguments*
- *Java keywords*

- *Java naming conventions*
- *Wrapper classes*
- *Typecasting*
- *Autoboxing & Unboxing*

Getting used to OOPS Way of doing things

- *What is Object oriented paradigm*
- *Everything is objects!*
- *Objects are instances of class*
- *5 OOP Principles*
    - *Encapsulation/Data Hiding*
    - *Inheritance (Remodeled bicycle)*
    - *Polymorphism (E.g. finding area of shapes)*
    - *Abstraction (If you don't know something, then don't do it!)*
    - *Interface (choosing which mode of travel based on situation)*
- *Advantage of being object-oriented*
- *Other object-oriented programming languages*

Delving deeper into Java developer's guide – The Java API

- *What developers need?*
- *Java documentation*

- *Types of Java docs*
- *Java Platform, Standard Edition (Java SE)*
- *Java Platform, Enterprise Edition (Java EE)*
- *How the Java API structured?*

Mastering Array & String Handling techniques

- *What is an Array and why I need an array?*
- *Declaring and initializing an array*
- *Array of Primitives | Objects*
- *Facts about Arrays*
- *Performing operations in arrays*
- *Arrays class*
- *Array of Objects*
- *What is String and why I need it?*
- *Creating a String*
- *String escape sequence & concatenating a String*
- *Efficient concatenation*
- *String functions*
- *When to use a String and when not?*

Exception Handling & Writing your own exceptions

- *What is an Exception?*
- *When and Where an Exception can occur?*
- *Types of Exceptions & its hierarchy*
- *Handling/Throwing checked & Unchecked exceptions*
- *Writing your own exception types*
- *Best practices*

Processing input and output through I/O Programming

- *What is an I/O?*
- *Getting an input/returning an output*
- *Everything is through Streams*
- *Types of Streams*
- *Reading i/p from console*
    - *Scanner*
    - *InputStreamReader*
    - *Console*

- *Beauty of buffers*
- *Writing o/p to Console*
- *File Manipulation*
  - *Reading i/p from a file*
  - *Writing o/p to a file*
  - *Make a new directory*
  - *List the contents of the directory*
  - *Get the absolute path of a file or directory*
  - *Delete a file or directory*
  - *Compression*
- *java.io at a glance*
- *Implementing a Stream and writing your own I/O class*

Nailing data Handling using Collections and generics

- *Introduction to collections*
- *Collection Interfaces*
- *List Interface & its Operations*
- *Set Interface & its Operations*
- *Iterating collections using Iterator*
- *Collection of objects*
- *Map Interface & its Operations*
- *Sorting collections using comparable and comparator*

- *Collections class*
- *Generics*
- *Best practices*

Storing Java objects through Serialization

- *What is Serialization & Why I need it?*
- *How to serialize objects?*
- *How to De-serialize objects?*
- *Serializable – A marker interface*
- *Using Transient & static keyword*
- *Using SerialVersionUID*
- *Best Practices*

Coding in an Multi-threaded environment

- *What is Multithreading & its terminologies*
- *Lifecycle of a thread and its states*
- *Understanding Thread class & Runnable interface*
- *ThreadScheduler – The master of all threads*
- *Handling Thread priorities*

- *Multithreading benefits & costs*
- *Using synchronized keyword & other best practices*

J2EE

Introduction to Web App development & Understanding J2EE Architecture

- *What is Web Application and its other terminologies*
- *What is J2EE & its architecture*
- *J2EE components and web flow*

Designing Servlets & Filters

- *Intro to servlets & its lifecycle*
- *Configuring a servlet*
- *Servlet API, Deployment descriptor & configuration*
- *Session handling*
- *Servlet filters*

Designing JSPs

- *JSP Introduction, lifecycle & tags*
- *JSP Scriptlets*
- *JSP directives*
- *JSP implicit objects*
- *Beans & Forms processing*
- *JSTL*

DB Programming using JDBC & MySQL

- *Intro to DB Programming*
- *MySQL Installation and setup*
- *Designing DB and tables in MySQL*
- *Writing queries for performing DB I-CRUD (Insert, Create, Retrieve, Update & Delete) operations*
- *Connecting to DB and executing queries from Java app using JDBC*
    - *JDBC API Overview and its Driver Types*
    - *How JDBC works*
    - *Steps Involved*
    - *JDBC process details*
    - *Queries*
    - *Writing Prepared& Callable Statements*

Pushing Beyond the limit

Struts 2 – A Frond-end web Framework

- *Struts 2 introduction & its architecture*
- *Struts 2 Configuration*
- *Writing Action classes*
- *Adding workflow with interceptors*
- *Working with OGNL & ValueStack*
- *Building the view: Struts2 tags & results*
- *Struts 2 Validation framework & AJAX validation*
- *Struts 2 type conversions*

Spring Framework – The Beast

- *Spring Introduction & its architecture*
- *Defining and wiring beans using Dependency Injection (DI)*
- *SpEL – Wiring with expressions*

- *Using Auto-wiring & Auto-discovery feature*
- *Aspect-oriented Spring (AOP)*
- *Hitting the DB using Spring JdbcTemplate*
- *Building web apps with Spring MVC – An Insight*
- *Integrating Hibernate & Struts2 with Spring*

Hibernate – An ORM Tool

- *Introduction to Hibernate*
- *Configuring DB and tables in Hibernate*
- *Mapping & Associations*
- *Using Hibernate Query Language (HQL)*
- *Writing Hibernate Criteria*
- *Hibernate Caching*
- *Hibernate get vs. load, save vs. persist, openSession vs. getCurrentSession*

Add-Ons

- *Insight into Eclipse IDE*
- *Working with Maven build Tool*
- *SDLC process in a real time s/w development*

Deliverables & Commitments

- *Clean and crispy regular class notes and other references (Word doc, PPTs, Websites, Books etc.)*
- *Coding snippets with proper comments*
- *Tailor-made assignments with its solutions*
- *Periodic assessments to test student's knowledge on the concepts mastered*
- *Tips on General: Best practices on programming, Interview facing skills, Soft skills, Resume preparation, etc.*
- *Interactive sessions*
- *On-time commencement of classes with no delays*
- *24 x 7 support*