Core java

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* *Why Java?*
* OOPs
* Platform Independent
* Automatic memory management
* Compiled/interpreted approach
* Robust
* Secure
* Dynamic linking
* Multithreading
* *Fundamentals of Java*
* Data types
* Operators
* Control statements
* Arrays
* Enhanced for loop
* Enumerations
* Static import
* Auto boxing and auto Unboxing
* Variable arguments
* Essentials of OOPs programming in java
* Object and class definition
* Using encapsulation
* Inheritance and polymorphism
* Writing java classes
* Encapsulation
* Polymorphism
* Inheritance
* Class fundamentals
* Using objects
* Constructors
* Garbage collection
* Method overloading
* Method overriding
* Static members
* Understanding interfaces
* Using interfaces class
* Packages
* Understanding classpath
* Access modifiers and their scope
* Exception handling
* Exception types
* Using try and catch block
* Throw, throws keyword finally block
* Writing user defined exceptions
* I/O Operations in java
* Byte oriented streams
* File handling
* Readers and writers
* Multithreading programming
* Understanding threads and its states
* Thread class and runnable interface
* Thread priorities
* Thread synchronization
* Preventing deadlocks
* Java.util package/Collection Framework
* Collection and iterator interface
* Enumeration
* List and Arraylist
* Comparator
* Set interface and sortedset
* Hashtable
* Properties
* Generics
* Introduction to generics
* Using build-in generics
* Writing simple generic class
* Bounded generics
* Wild card generics
* Inner classes
* Nested top level classes
* Member classes
* Local classes
* Anonymous classes

***Working Professional with Core Java***

* Abstract window toolkit
* Graphics
* Colour and font
* AWT components/controls
* Event handling and layouts
* Swing programming
* Introduction to swing
* Light weight components
* Swing hierarchy
* Atomic components e.g. jbutton,jlist and more
* Intermediate container e.g. jpanel,jsplitpane and more
* Top-level container e.g. jframe and japplet
* Swing related Events
* Developing java apps
* Defining a solution without writing code
* Organizing a concept solution
* Creating a program skeleton
* Defining error checking requirements
* Introduction to application security
* **Major Project in Core Java**
* Creating GUI based Online Exam Application in a jar file without using third party Application (eclipse) with the help of Command Prompt.