**Basic Java**

1. Introduction
   1. History
   2. The Java language
   3. Comparison with C# and C++
2. Setup
   1. JDK vs JRE
   2. Installing JDK
   3. Installing the development environment
   4. JVM concepts
   5. Garbage collection vs
3. Basics
   1. Data types
   2. Strings
   3. Arrays
   4. Variables
   5. Type checking and conversion
   6. Operators
   7. Control statements
   8. Jump statements
   9. Miscellaneous topics
4. Fundamentals
   1. Classes, objects and methods
      1. Object oriented programming and principles
      2. Class members and data
      3. Revisiting data hiding and encapsulation
      4. Class hierarchies and important keywords
      5. More about classes
      6. Predefined and custom classes
      7. Introduction to Class design and Identifying objects
      8. Environment variables
      9. Packages
   2. Inheritance
      1. Super classes and sub classes
      2. Overriding and overloading
      3. Reflection
      4. Introduction to Is-a and has-a
   3. Polymorphism
      1. Static polymorphism
      2. dynamic polymorphism
5. Intermediate
   1. Callbacks and interfaces
   2. Nested classes
   3. Exception handling
   4. Assertion
   5. Logging
   6. Debugging
   7. Enumerations
   8. Generics
      1. Generics and their need.
      2. Generic lists and arrays
      3. Define your own generic methods
      4. Define your own generic classes
   9. Java Collections
   10. Autoboxing
   11. Lambda expressions
   12. Modules and services
   13. JAR files

**Advanced Java**

1. Advanced
   1. I/O
      1. Streams and files
      2. Text input and output
      3. Binary reading and writing
      4. File management
      5. RegEx
   2. XML
      1. Introduction
      2. Types of parsers
      3. Parsing and streaming out
   3. JSON
      1. Introduction and usage areas
      2. Parsing and streaming
      3. JSON as file DB ?
   4. Internationalization
      1. Encoding and code pages
      2. Unicode and UTF
      3. Java Strings
      4. Windows character sets

**Microservices with Java**

1. Introduction to Microservices using Java
   1. Microservices and API ?
   2. Microservices vs Monolithic architecture
   3. SOAP VS Rest API’s
   4. Introduction to MVC
   5. Introduction to spring and spring boot
   6. Writing and deploying
      1. GET, PUT, POST, DELETE
   7. API response codes
   8. Testing API’s with postman
   9. Deploying on cloud (AWS)
   10. Security, encryption, authentication, and authorization

**Design patterns**

1. Design patterns
   1. Introduction
   2. UML and class diagrams
   3. Is-a relationship (inheritance)
   4. Has-a relationship (association)
      1. Aggregation
      2. Composition
   5. Introduction to common patterns
      1. Singelton
      2. Factory
      3. Abstract factory
      4. Composite
      5. Adaptor
      6. Observer
   6. General concepts when designing an application
   7. Identifying pattern usage

**Algorithms and Data Structures**

1. Algorithms and data structures
   1. Basics
      1. Algorithms vs data structures
      2. Analysis and Big O notation
      3. Design
   2. Basic Data structures
      1. Linked list (Array and node based)
      2. Doubly list
      3. Stack
      4. Queues (Array and node based)
   3. Recursion and back tracking
      1. Solving N-queen problem
   4. Sorting and searching
      1. Divide and conquer
      2. Sorting
         1. Bubble sort
         2. Insertion sort
         3. Merge sort
         4. Quick sort
         5. Heaps and heap sort
         6. Priority Queues (Introduction)
         7. External sorting (Introduction only)
            1. Further reading
      3. Searching
         1. Sequential search
         2. Binary search
         3. Binary search trees
         4. Balanced search trees
         5. Hash table
         6. External searching (Introduction only)
            1. Further reading
   5. Greedy algorithms paradigm (theory only)
   6. Dynamic programming paradigm (theory only)
      1. Introduction.
      2. Example problems.
      3. Further reading.
   7. Graphs
      1. Directed and undirected graphs
      2. Searching and Traversing graphs
         1. DFS and BFS
         2. Solving a Maze
         3. Cycle detection
      3. Topological sorting
      4. Strong connectivity and strongly connected components
         1. Kosaraju’s algorithm
      5. Revisit priority queues
      6. Weighted graphs
         1. Minimum spanning trees
            1. Prims algorithm
            2. Kruskal’s algorithm
         2. Single source shortest paths
            1. Dijkstra’s algorithm
            2. Bellman-Ford algorithm
         3. All source shortest paths
            1. Floyd-Warshall algorithm
            2. Johnson’s algorithm
         4. Further reading.
   8. Strings
      1. String sorting
      2. Tries
      3. Sub string searching
         1. Brute-force
         2. Knuth-Morris-Pratt
         3. Further reading.
   9. NP-Complete problems
   10. Genetic algorithms
       1. Introduction
       2. An alternative to DP paradigm ?
       3. Solving NP-complete problems with genetic algorithms ?
       4. Theory before implementation
       5. Revisiting: Solving N-Queen problem
   11. Advanced topics (theory only)

**Recommended books:**

* **Java**

Java: The Complete Reference, Herbert Schildt

Core Java: Fundamentals, Cay Horstmann

Core Java: Advanced, Cay Horstmann

Algorithms, Robert\_Sedgewick and Kevin\_Wayne

* **C & C++**

The C Programming Language, Ritchie & Kernighan

Data Structures and Algorithms, Aho

Algorithms, S. Dasgupta

Algorithms, Robert Sedgewick

Inside the C++ Object Model, Stanley B. Lippman

The C++ Programming Language, Stroustrup, (4th edition to read modern C++)

Introduction to Algorithms, CRLS

* **C#**

C# in Depth, Jon Skeet

C# in a Nutshell, Joseph Albahari

* **Microservices Basic (C#)**

Building Microservices, Sam Newman

* **Microservices Basics (Java)**

Spring Microservices, John Carnell

Spring boot, Craig Walls

* **Design patterns & UML**

Design Patterns: Elements of Reusable Object-Oriented Software (Gang of four)

UML (only class diagrams, anywhere from internet)