



Course Progression Syllabus

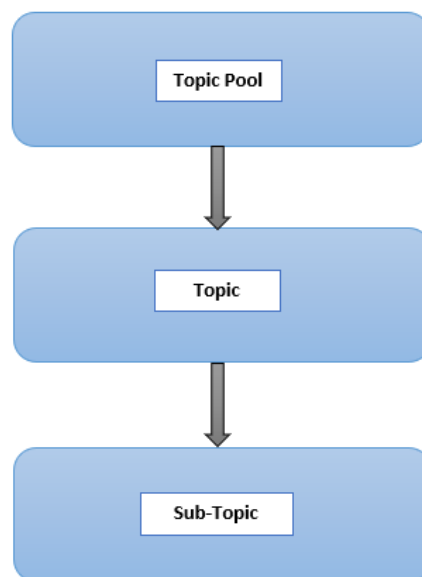
Hey,

Please find attached the course structure & list of topics/stacks that we provide training for in the 6 month rigorous course. The content has been carefully created keeping industry standards in mind and we are confident that proficiency in these will make you a "Rockstar Developer".

Newton school trains you on the following listed tech stacks:

- Data Structures & Algorithms
- HTML
- CSS
- JavaScript
- ReactJS
- NodeJS
- ExpressJS
- MongoDB
- MySQL

Course Structure





Course structure of Newton School is divided into various **Topic Pools**, which is further divided into **Topic**, which is then finally divided into **Sub Topics**.

Course Structure break-up

Data Structures & Algorithms:

1. DSA Basic 1

1.1. Basics of Programming (Java)

1.1.1. Variables & Operators

1.1.2. Control Structures

1.1.2.1. If-Else Ladder

1.1.2.2. Switch Case & Jump Statements

1.1.3. Loops

1.1.3.1. For Loop

1.1.3.2. While Loop & Do-While Loop

1.1.3.3. Pattern Making

1.1.4. Functions

1.1.5. Complexity Analysis

1.1.5.1. Time Complexity

1.1.5.2. Space Complexity

1.1.6. Java Programming Syntax

1.2. Arrays

1.2.1. 1-D Array

1.2.2. 2-D Array (Matrix)

1.2.3. Prefix Sum

1.2.3.1. Kadane Algorithm

1.3. Recursions

1.3.1. Base Case & Mathematical Interpretation

1.3.2. Tail & Non-Tail recursion

1.4. Sorting Algorithms

1.4.1. Simple Sorting

1.4.1.1. Bubble Sort

1.4.1.2. Insertion Sort

1.4.1.3. Selection Sort

1.4.1.4. Merge Sort



- 1.4.1.5. Shell Sort
 - 1.4.1.6. Quick Sort
 - 1.4.2. Merge Sort
 - 1.4.3. Quick Sort
 - 1.4.4. Inplace and Stable Sorting Algorithms
- 1.5. Strings**
 - 1.5.1. String Basics
 - 1.5.2. String & Char Array
 - 1.5.3. String Mutability
- 1.6. Collections**
 - 1.6.1. Hashing
 - 1.6.2. Array List
 - 1.6.3. Comparator Sort
 - 1.6.4. Upper and Lower Bound
- 1.7. Maths**
 - 1.7.1. Modulo Arithmetics
 - 1.7.2. Primality Testing
 - 1.7.2.1. Trial Division Method
 - 1.7.2.2. Sieve of Erathosthenes
 - 1.7.3. GCD
 - 1.7.3.1. Euclidian Basic
 - 1.7.3.2. Euclidian Extended
 - 1.7.3.3. Euler totient method
 - 1.7.4. Bit Manipulation
 - 1.7.5. Combinatorics
 - 1.7.6. Probability
 - 1.7.7. Fibonacci Properties
 - 1.7.8. General Maths
 - 1.7.9. Geometry
 - 1.7.10. Game Theory
- 2. DSA Basic 2**
 - 2.1. Object Oriented Programming**
 - 2.1.1. Class & Objects
 - 2.1.2. Polymorphism
 - 2.1.3. Inheritance
 - 2.1.4. Abstraction
 - 2.1.5. Encapsulation
 - 2.2. Searching**
 - 2.2.1. Linear Search
 - 2.2.2. Binary Search
 - 2.2.3. Two-Pointers



- 2.3. Linked List**
 - 2.3.1. Linked Lists vs Array
 - 2.3.2. Single Linked List
 - 2.3.3. Doubly Linked List
 - 2.3.4. Circular Linked List
- 2.4. Stack**
 - 2.4.1. Implementing with Arrays
 - 2.4.2. Implementing with Linked List
 - 2.4.3. Stack as Library
- 2.5. Queues**
 - 2.5.1. Simple Queue
 - 2.5.2. De-Queue
 - 2.5.3. Circular Queue
 - 2.5.4. Implementing with Arrays
 - 2.5.5. Implementing with Linked List
 - 2.5.6. Queue as Library
- 3. DSA Basic 3**
 - 3.1. Heaps**
 - 3.1.1. Introduction
 - 3.1.2. Build Heap
 - 3.1.3. Heap Sort
- 4. DSA Basic 4**
 - 4.1. Dynamic Programming**
 - 4.2. Pattern Matching**
 - 4.2.1. Z Algorithm
 - 4.2.2. KMP
 - 4.3. Backtracking**
 - 4.4. Divide & Conquer Algorithms**
 - 4.5. Greedy Algorithms**
 - 4.6. Trees**
 - 4.6.1. Binary Tree
 - 4.6.1.1. Traversals (BFS, DFS)
 - 4.6.2. Binary Search Tree
 - 4.6.3. Red-Black Tree
 - 4.6.4. AVL Tree
 - 4.7. Graphs**
 - 4.7.1. Directed Graph
 - 4.7.2. Undirected Graph



Web Development Fundamentals:

1. HTML

- 1.1. HTML & HTML5 Tags
- 1.2. Elements, Attributes and Meta Tags
- 1.3. Formatting
- 1.4. Tables and List
- 1.5. HTML Links
- 1.6. Frames
- 1.7. Background, Colors and Fonts
- 1.8. Forms
- 1.9. Marquees, Header and Layouts

2. CSS

- 2.1. Color, Background and Fonts
- 2.2. Text, Images and Links
- 2.3. Tables
- 2.4. Borders and Margins
- 2.5. List, Padding and Cursors
- 2.6. Outlines and Dimensions
- 2.7. Positioning
- 2.8. Layers
- 2.9. Validations
- 2.10. Grids
- 2.11. Flex
- 2.12. CSS Frameworks

3. JavaScript

- 3.1. Variables and Operators
- 3.2. If-Else and Switch Case
- 3.3. Loops and Loop Control
- 3.4. Functions
- 3.5. Events
- 3.6. Cookies
- 3.7. Objects
 - 3.7.1. Number
 - 3.7.2. Boolean
 - 3.7.3. Strings
 - 3.7.4. Arrays
 - 3.7.5. Date



- 3.7.6. Math
- 3.7.7. RegExp
- 3.7.8. HTML DOM
- 3.8. Error Handling
- 3.9. Validations
- 3.10. Animations
- 3.11. Multimedia
- 3.12. Debugging

Frontend Libraries and Framework:

1. Bootstrap

- 1.1. Grid System
- 1.2. Code
- 1.3. Tables, Forms and Buttons
- 1.4. Helper Classes
- 1.5. Bootstrap Layout and Component

2. ReactJS

- 2.1. Introduction to ReactJS
- 2.2. JSX
- 2.3. Component
- 2.4. State
- 2.5. Props
- 2.6. Component Lifecycle
- 2.7. Component API
- 2.8. Events
- 2.9. Flux
- 2.10. Higher Order Component

3. Redux

- 3.1. Introduction to Redux
- 3.2. Data Flow
- 3.3. Store, and Actions
- 3.4. Pure Functions
- 3.5. Reducers
- 3.6. Middleware
- 3.7. Devtools and Testing



Backend Technologies:

1. NodeJS

- 1.1. Introduction to NodeJS
- 1.2. REPL terminals
- 1.3. Callbacks
- 1.4. Event Loop
- 1.5. Buffers and Streams
- 1.6. File System
- 1.7. Global Objects
- 1.8. Utility and Web Modules
- 1.9. REST API
- 1.10. Packaging

2. ExpressJS

- 2.1. Introduction to ExpressJS
- 2.2. Routing
- 2.3. HTTP Methods
- 2.4. Form Data
- 2.5. Cookies and Sessions
- 2.6. Authentication
- 2.7. REST API
- 2.8. Scaffolding
- 2.9. Error Handling
- 2.10. Debugging

Database Management System:

1. MongoDB

- 1.1. Introduction to NoSQL
- 1.2. Data Modelling
- 1.3. Create and Drop Database
- 1.4. Create and Drop Collection
- 1.5. Data Types
- 1.6. Insert, Query, Update, Delete Document
- 1.7. Limiting and Sorting Records
- 1.8. Indexing
- 1.9. Deployment



2. MySQL

- 2.1. Introduction to MySQL
- 2.2. Create, Drop, Select Database
- 2.3. Data Types
- 2.4. Create and Drop Tables
- 2.5. Insert, Select, Update, Delete Query
- 2.6. Where and Like Clause
- 2.7. Order
- 2.8. Join
- 2.9. Union
- 2.10. Transaction
- 2.11. Temporary Tables
- 2.12. Sub Queries