

# Syllabus

## C++

1. **c++ from here** Introduction to computers - Basics of C++ - Number representation, Basic data types - int, float, double, char, bool, void.  
Flow of Control - Conditional statements - If-else, Switch-case constructs, Loops - while, do while, for.
2. Functions - user defined functions, library functions, parameter passing - call by value, call by reference, return values, Recursion.  
Arrays - Single, Multi-Dimensional Arrays, initialization, accessing individual elements, passing arrays as parameters to functions.
3. Pointers and Dynamic Arrays - Multidimensional Dynamic Arrays, creation and deletion of single and multi-dimensional arrays.
4. **Data structures** - Big-oh, Big-omega, Theta, Little-oh, Little-omega notations, Properties of Asymptotic Notations
5. Analysis of Iterative and Recursive Algorithms
6. Implementation and Applications of Stacks
7. Queues
8. Practical stack and queue implementations
9. Practical stack and queue implementations
10. Introduction to Trees and importance like sorts
11. Implementation of Trees
12. Binary trees

13. Binary search trees
14. Query and Update Operations on BSTs
15. BST Traversals
16. BST Traversals
17. Hashing: Implementation of Dictionaries,
18. Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing, Analysis of Search Operations
19. Priority Queues: Priority Queue ADT,
20. Binary Heap Implementation and Applications of Priority Queues
21. Sorting Algorithms intro and complexity
22. Stability and In Place Properties, Insertion Sort, Merge Sort,
23. Quick Sort
24. Heap Sort and other applications - how trees implement heaps
25. Linear Sorting  
Algorithms: Counting Sort, Radix Sort, Bucket Sort
26. graphs intro
27. minimum spanning trees
28. Shortest path dijkstra
29. Basic file structures. How files work and records are stored on disk
30. Only theoretical introduction to btrees - (if students are able to get this else skip)

Basic programs to discuss in the classes. Very simple ones which we will discuss in class

10 hours

1. Write a program to implement stack using arrays.
2. Write a program to evaluate a given postfix expression using stacks.
3. Write a program to convert a given infix expression to postfix form using stacks.
4. Write a program to implement circular queue using arrays.
5. Write a program to implement double ended queue (de queue) using arrays.
6. Write a program to implement a stack using two queues such that the push operation runs in constant time and the pop operation runs in linear time.
7. Write a program to implement a stack using two queues such that the push operation runs in linear time and the pop operation runs in constant time.
8. Write a program to implement a queue using two stacks such that the enqueue operation runs in constant time and the dequeue operation runs in linear time.
9. Write a program to implement a queue using two stacks such that the enqueue operation runs in linear time and the dequeue operation runs in constant time.
10. Write programs to implement the following data structures:
  - (a) Single linked list
  - (b) Double linked list
11. Write a program to implement a stack using a linked list such that the push and pop operations of stack still take  $O(1)$  time.
12. Write a program to implement a queue using a linked list such that the enqueue and dequeue operations of queue take  $O(1)$  time.

If possible

Write a program to create a binary search tree(BST) by considering the keys in given

order and perform the following operations on it.

- (a) Minimum key
- (b) Maximum key
- (c) Search for a given key
- (d) Find predecessor of a node

- (e) Find successor of a node
- (f) delete a node with given key

Implement the following sorting algorithms:

- (a) Insertion sort
- (b) Merge sort
- (c) Quick sort
- (d) Heap sort

Write programs for implementation of graph traversals by applying: (a) BFS (b) DFS

Write programs to find out a minimum spanning tree of a simple connected undirected

graph by applying: (a) Prim's algorithm (b) Kruskal's algorithm

Write a program to implement Dijkstra's algorithm for solving single source shortest path problem using priority queue

## Object oriented programming

1. Introduction to Java or if students not comfortable c++
2. Object Oriented Thinking – Object - Oriented Design Access Modifiers
3. Single Responsibility Principle
4. Solid techniques
5. Interface and its uses - A Paradigm - Program Structure, TheIntuitive Description of Inheritance
6. Subclass, Subtype, and Substitutability
7. Types of Polymorphism
8. Synchronized keyboard and example

2 hours buffer - 50 till here

## DBMS

1. File Versus a DBMS, Advantages of DBMS
2. Describing and storing data in DBMS, Architecture of a DBMS,
3. Entity Relationship model- features of ER model
4. SQL Query introduction
5. Constraints, Form of SQL query, UNION, INTERSECT and EXCEPT - basics of SQL
6. Normalization process and intro to 1NF, 2NF, 3NF and BCNF,
7. Normalization process and intro to 1NF, 2NF, 3NF and BCNF, with examples
8. Benefits of Normalization
9. ACID properties
10. transactions introduction why and when do we use it
11. schedules and concurrent execution of transactions
12. Modern no sql stores and where do we store it
13. Recovery of databases, log based implementation of mysql
14. Buffer
15. Run through basic of SQL programming from

<https://www.khanacademy.org/computing/computer-programming/sql> for 5 hours

Micro processing/ Computer architecture - decide to keep it or skip it - 2-3 days  
what sir sees fit and students level - nor relevant to interview

## Operating systems

1. Introduction: Batch, iterative, time sharing, multiprocessor, and real-time systems, Unix system introduction and commands
2. Operating system structures : Computer system structure, Dual mode operation, System components, System Calls,
3. Processes and Threads : Process Concept, Process Scheduling, Operations on Processes,  
Cooperating Processes, Interprocess Communication,
4. Communication in Client – Server Systems
5. Multithreading Models, Threading Issues, threads Basic Concepts  
CPU Scheduling : Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling,  
Real-Time Scheduling
6. Process Synchronization : Synchronization Background, The Critical-Section Problem,  
Synchronization , Semaphores, Classic Problems of Synchronization, Critical Regions, OS Synchronization
7. Deadlocks : System Model, Deadlock Characterisation, Handling Deadlocks, Prevention, Avoidance, Detection, Recovery from Deadlock
8. Memory Management : Memory Management Background, Swapping, Segmentation with Paging, Virtual Memory, Demand Paging
9. File System : File Concept, Access Methods, Directory Structure, File-System Mounting, File

Sharing, Protection File-System Structure, File-System Implementation

10. User Authentication , Program Threats, System Threats, Securing Systems and Facilities, Intrusion Detection, Cryptography, Computer-Security Classifications

## Computer Networks

1. Introduction – network architecture - Quantitative performance metrics - network design.
2. OSI Reference Model
3. TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models Low
4. Virtual circuits Switching technologies-Switched Ethernet and ATM The design of hardware based switches
5. Network layer network layer design issues Congestion control
6. The network layer in the internet-Internet Protocol (IP).- Unicast, multicast, and inter domain routing
7. Transport layer-Elements of transport protocol-Congestion control
8. Network applications and the protocols- File transfer protocol - email and the Web, multimedia

2 hours buffer

## Algorithms Analysis

1. Introduction to Algorithm Analysis
2. Asymptotic Notations, Divide and Conquer Method,
3. Quick sort analysis
4. Master Theorem

5. Elements of Greedy Method, Activity Selection Problem,
6. Knapsack Problem, Prim's and Krukal's Algorithms for finding Minimum Spanning Tree
7. Elements of Dynamic Programming
8. Matrix Chain Multiplication
9. Other examples from the theorem discussed already

1 hour buffer

Web Development - 140 hours

Marking frontend and backend together/ web - any one can take it

1. How does Internet work What is http and browsers
2. What is DNS and how do we host it, How to use developer tools
3. SSH and command line basic
4. Git
5. Frontend Html tags
6. Html Dom Manipulations. JQuery introduction
7. JQuery and selectors



8. HTML Semantics. forms etc
9. Cover the html handbook for 3 hours
10. Cover the html handbook for 3 hours
11. Cover the html handbook for 3 hours
12. Simple web form
13. CSS introduction
14. CSS Floats
15. CSS selectors
16. CSS Specificity
17. Cover the CSS Handbook for 2 hours
18. Cover the CSS Handbook for 3 hours
19. Cover the CSS Handbook for 3 hours
20. Buffer

1. Introduction to js
2. Don't allow the people to explain from any other source except ydkjs
3. Intro to js programming setup vscode
4. Intro to js programming - syntax and running
5. Types and values
6. Natives and Coercion(type casting)
7. JS Lexical scope
8. Hoisting closures
9. classes and scopes
10. This keywords and ES6 prototypes
11. Asynchronous programming intro
12. Asynchronous programming intro with examples

13. Callbacks and events
14. Timeouts
15. Promises
16. Generators
17. Performance and Tuning - from ydkjs
18. Async patterns from ydkjs
19. Buffer
20. Buffer → buffer to be filled with examples and practical examples of the speaker

## Backend Language Introduction

1. Setup node and editor (already from js class) npm,
2. running a basic app, npm install, node repl
3. input output to cmdline, a basic app introducing
  1. timeouts
  2. callbacks
  3. events
4. packages and exports, package json following up on the example app
5. continue on node dependencies, resolution and packaging
6. Node and browser differences - intro to v8
7. Event loop
8. Event loop examples
9. Event loop questions and more example , start a basic app here
10. Example including timeout, promises and types of promise operations
11. Introduce Http and https with axios

12. File handling, environment options handling, how to pass configs
13. Express and middleware
14. Express and middleware
15. famous modules - fs
16. modules crypto
17. modules path
18. modules os
19. Streams
20. Buffer

#### Backend app with Nodejs

1. Introduction to MVC
2. Introduction to json
3. Request parameters
4. express modules like cookie and body parsers
5. Response and response headers
6. logging with morgan, 404 and error handlers
7. Templating with ejs/pug
8. implement a login page, Redirect page
9. Basic app - read from file and power an api
10. helmet for security, intro to web security
11. CSRF, login security
12. Use in memory express session and intro to sessions
13. setup authorization to routes with users access
14. Validating and sanitizing input, guard functions in controller

15. File uploads in express
16. Introduction to Mongodb, sql vs nosql
17. Introduction to Mongodb document model
18. Mongodb operations and installation in atlas
19. Mongodb mongoose
20. Use express session or mongodb to create a user model and set in session

### React frontend learning

1. Create react app and setup
2. functional vs class components
3. props vs state
4. component life cycle
5. Conditional rendering
6. List of items, importance of rendering by id
7. Basic hooks
8. Higher order components, error boundaries
9. React Router
10. useCallback and useRef
11. useMemo useReducer
12. useContext
13. Api calls
14. Redux
15. Redux libraries
16. Ant design and a form

17. Validations and better routing
18. styling in react
19. debugging react
20. testing and other frameworks

## System design

1. Measuring latencies - how do we measure application performance
2. Caching techniques - redis
3. Web server - high Availability
4. Need of nginx and ha proxy
5. Different types of auth like oauth, jwt openid
6. Rsa and crypto, hashing, encrypting, bcrypt for passwords
7. CDN and media delivery
8. Development strategies like Solid, dry and SRE
9. Elasticsearch and inverted index
10. Elasticsearch and inverted index
11. Docker
12. Kubernetes
13. Load mitigation by throttling and back pressure
14. Kafka
15. micro services vs SOA
16. OWASP and cors
17. CAP theorem
18. Dbs and their tradeoffs

19. APM like new relic

20. Buffer

<https://github.com/donnemartin/system-design-primer>