

C Syllabus	C++ Syllabus	Data Structures Syllabus
<p>Introduction</p> <ul style="list-style-type: none"> History of C Characteristics of C C Program Structure Constants Data types Variables Keywords Console Input/Output Statements Compilation and Execution <p>Operators</p> <ul style="list-style-type: none"> Arithmetic Unary Assignment Relational & Logical Conditional <p>Branching & Looping Statements</p> <ul style="list-style-type: none"> if Statement Nested if Statement else-if Ladder switch Statement Looping Concepts for loop while loop do-while loop Jump Statements <p>Arrays</p> <ul style="list-style-type: none"> Array Concepts Rules & Restrictions Single & Multi Dimensional Arrays <p>Functions</p> <ul style="list-style-type: none"> Types of Functions Functions and Arrays Function Prototyping Scope of Variables Built-in Functions 	<p>Introduction</p> <ul style="list-style-type: none"> History Advantage Need for C++ Introduction to OOPS First C++ Program Introduction to OOPS Input and Output Statements Defining classes <p>Classes and objects</p> <ul style="list-style-type: none"> Instantiation Creating Objects Constructors and Destructors Access Specifiers Inline Functions Introduction to Function Overloading Default Function Arguments Static Members <p>Pointers Vs Class Objects</p> <ul style="list-style-type: none"> Pointers and Instances Dynamic Memory Allocation Memory Management Creating Object arrays Manipulating Object arrays <p>Inheritance</p> <ul style="list-style-type: none"> Concepts & Syntax Protected Keyword Derivation Types Types of Inheritance <p>Polymorphism</p> <ul style="list-style-type: none"> Overloading (method & operator) Virtual methods Compile time Polymorphism Run time Polymorphism Built-in streams <p>Streams and File I/O</p> <ul style="list-style-type: none"> File streams String streams <p>Advanced Topics</p> <ul style="list-style-type: none"> Friend Functions 	<p>General concepts</p> <ul style="list-style-type: none"> Abstract data structure as an organization of data with specified properties and operations Time and space analysis of algorithms Big oh and theta notations Average, best and worst case analysis Simple recurrence relations and use in algorithm analysis <p>Linear data structures</p> <ul style="list-style-type: none"> Arrays,lists, stacks, queues Array and linked structure implementations of lists, stacks, queues Array of nodes and dynamic pointer implementations of linked structures <p>Trees</p> <ul style="list-style-type: none"> General and binary trees Representations and traversals General trees as binary trees Binary search trees Applications The concept of balancing and its advantages Some balanced tree mechanism, eg. AVL trees, 2-3 trees, red-black trees, self-adjusting trees, <p>Algorithm design techniques</p>

<p>Strings</p> <ul style="list-style-type: none"> • String Functions • String Manipulation <p>Pointers</p> <ul style="list-style-type: none"> • Pointer Concepts • Pointers and Functions • Pointers and Arrays • Array of Pointers • Static Initialization • Pointers and Structures • Illegal indirection <p>Structures</p> <ul style="list-style-type: none"> • Defining New Data types • Unions • Type Casting • Enumerated Data types • Static Variables • Type Definition <p>Dynamic Memory Allocation and Data Structures</p> <ul style="list-style-type: none"> • malloc(),sizeof() and free() • calloc() and realloc() • Stacks and Queues • Linked Lists • Trees 	<ul style="list-style-type: none"> • Friend Classes • Copy Constructors • Templates • Adding Graphics <p>Projects</p> <ul style="list-style-type: none"> • Creating Projects • Making Files <p>Macros</p> <ul style="list-style-type: none"> • About Macros • Usability of Macros <p>Assertions / Exceptions</p> <ul style="list-style-type: none"> • About Assertions and Exceptions • throw () and catch () 	<ul style="list-style-type: none"> • Greedy methods • Priority queue search • Exhaustive search • Divide and conquer • Dynamic programming • Recursion • Influence of data structure on algorithm performance <p>Hashing</p> <ul style="list-style-type: none"> • Hash functions • Collision resolution • Expected behavior <p>Graphs and digraphs Representations</p> <ul style="list-style-type: none"> • Breadth and depth first searches • Connectivity algorithms • Shortest path • Minimal spanning tree • The union find problem • Hamiltonian path and travelling salesperson problems • Network flow Matchings <p>Sorting</p> <ul style="list-style-type: none"> • Elementary sorts: selection,insertion, bubblesort. Quicksort, mergesort, heapsort • Bucket sorting • External sorting • Worst case and average behavior • Lower bound for sorting using comparisons • Order statistics <p>NP vs. P</p> <ul style="list-style-type: none"> • The spaces P and NP • Polynomial reduction
--	--	---

		<ul style="list-style-type: none">• NP complete problems• Boolean satisfiability and Cook's theorem• Binpacking, knapsack, Hamiltonian path, TSP, independent set, max clique, integer linear programming, graph coloring• Approximation algorithms
--	--	--

STYLUS