LaunchPad - Data Structures and Algorithms in C++

# Overview

C++ is known to be a very powerful language. C++ allows you to have a lot of control as to how you use computer resources effectively and efficiently better than other languages. Thanks to C++'s performance, it is often used to develop game engines, games, and desktop apps. As a statically typed language, C++ is generally more performant than dynamically typed languages because the code is type-checked before it is executed. Companies like Google and Facebook, also need C++ developers to optimize their apps or work on their products.

With this course your learning will span –

1. Basic programming in C++
2. Recursion
3. Time and Space Complexity
4. Object Oriented Programming
5. Data Structures
6. Algorithms

This course entails more than 100 hours of teaching and hands on coding. Students will be solving more than 200 complex problems and will be evaluated for their problem solving skills and understanding of data structures throughout the course.

Programming is a practical skill and is best learnt by doing, hence we make it a point that all our classes are partitioned in a way to balance the theoretical and practical aspect.

# Highlights

* Lectures – more than 24
* Teaching Hours – 100+
* Problems done in class – 150+
* Problems in assignments – 100+
* Evaluations - Done
* Projects – 1
* Hackathons – 2
* Hands-on learning, Theory to Practical ratio – 1:3
* Student to Mentor Ratio – 15:1

# Curriculum

We have designed this course to provide you with a platform from where you can start your journey in the amazing world of programming and software. A beginners’ course including basic programming fundamentals, object oriented programming and in-depth coverage of data structures and algorithms. During the course, focus will be on developing strong fundamentals, algorithms and data structures skills via practical problem solving. Find the specifics of course below

## Programming Fundamentals and Logic Building (**Lecture 1 to 2**)

1. Flowcharts, pseudo-code and brain teasers
2. Getting Started with C++
3. Debugging

## Strings, Arrays, Recursion (**Lecture 3 to 10**)

1. Functions
2. Arrays, 2D Arrays
3. Character Arrays, Strings
4. Vectors in C++
5. Recursion
6. Introduction
7. Classical Recursion Problems
8. Back-tracking
9. Advanced Recursion problems
10. Optimization

## OOPs & Complexity Analysis (**Lecture 11 to 12**)

1. Space & Time Complexity Analysis
2. Object Oriented Programming
3. Vectors Implementation

## First Hackathon

Based on topics covered so far

## Data Structures and Algorithms (**Lecture 13 to 18**)

1. Stacks and Queues
2. Linked Lists
3. Generic Trees
4. Binary Trees
5. Binary Search Trees

Advanced Data Structures and Algorithms (**Lecture 19 to 25**)

1. Heaps
2. Hash Tables
3. Graphs – Traversals, Minimum Spanning trees, Shortest path algorithms
4. Tries
5. Text Processing – Text Matching, Compression
6. Dynamic Programming
7. Segment Tress

## Second Hackathon

Covering complete course

# Pre-requisite

* Basic computer literacy.
* Number System – Decimal, Binary.
* Some coding experience in any language, though not an essential but it’s a good to have.

# Sessions

* June – July
* August – November
* December - January
* February - April

# FAQ

**Q1. Who should do this course?**

Ans. Anybody who is interested in learning programming can do this course. C++ is a powerful and elegant language, but still is pretty easy to learn.

**Q2. I know C, should I do this course?**

Ans. Yes. Absolutely. Not only will this course teach you C++, but also Object Oriented Programming which forms the base of most, if not all code running live across the globe. Besides it builds your problem solving skills, teaches you some fine data structures and helps you develop the thinking process to optimise your solutions.

**Q3. I have done C++ in Class 12th, should I do this course?**

Ans. Yes, absolutely. There is a lot more to learn in C++ than the school curriculum. School syllabus just scratches the surface of C++ programming and doesn’t cover most of the Data Structures, which will be covered in this course in great detail.

**Q4. I didn’t have C++ in Class 12th, should I do this course?**

Ans. Yes. Our instructors will start the course beginners level. You don’t need any prior programming knowledge to attend this course. What we need is dedication and passion to learn programming from your side.

**Q5. Will this help me with competitive programming?**

Ans. This course will help you become smarter with your solutions. You’ll be able to write solutions that run in lesser space and time. The last time we heard from capable competitive programmers, they said these skills made them capable.

**Q6. Will this help me with interviews?**

Ans. Yes. If you learn well, all that the course intends to teach, you’ll be able to impress quite a few interviewers across quite a few fine companies.