



IDM TECHPARK
GUIDE'S FOR PERFECT CAREER PATHWAY



C, C++ Course Syllabus



N.S.D.C.
National
Skill Development
Corporation



1

About Us

IDMTECHPARK global retail & corporate training solutions provider in Coimbatore, Erode, Trichy & Salem that offers a comprehensive range of training and placement services for both fresher's and professionals seeking new opportunities. The company commenced its IT training business in 2016. A pioneer in IT education, over the years, we have trained over 50k students. Idmtechpark has a wide range of courses, maintains education standards & provides placement assistance.

www.idmtechparkcoimbatore.com
www.idmtechparkerode.com

+91 9585305700

**2**

About IDMTECHpark Education Quality

IDMTECHPARK is managed and developed by industry specialists with more than 8 years of expertise in the field. IDMTECHPARK offers a staff of highly skilled professional trainers who deliver effective IT training in a friendly setting, concentrating on the needs of each individual to help them succeed in a demanding work world. In the book of career and success, our staff never leaves a page unturned.

www.idmtechparkcoimbatore.com
www.idmtechparkerode.com

+91 9585305700

IDMTECHPARK's versatile instructor-led training class rooms and lower-class sizes enable people to engage more easily and absorb knowledge, resulting in remarkable results for both themselves and the organizations for which they work. Our training programmes are adaptable and customizable to ensure that each participant gets the most out of their time with us. IDMTECHPARK focuses in providing hands-on IT training in over 30 different courses.

- We teach in-demand courses
- We provide impactful learning material
- Our teachers are well-selected & trained
- We follow world-class teaching methods
- Our courses include E-Projects
- We conduct technical workshops
- Exams are held and based on Exams providing Certification
- Certificates are recognized the world over
- Our course timings are flexible



3

Our Recent Placement

Idmtechpark assists students in getting job placements on successful completion of their courses. Idmtechpark also provides recruitment assistance to organizations. Idmtechpark students are shortlisted based on the organization's requirement. To make students job-ready, Idmtechpark conducts workshops e.g. How to do Group Discussions, how to behave in a Personal Interview. From time to time, job fairs & campus recruitments are conducted. Workplace skills such as time management, making effective presentations and communication skills are also provided. All this helps students find appropriate jobs in the IT industry while also helping save companies recruitment costs.

Krishnaveni M

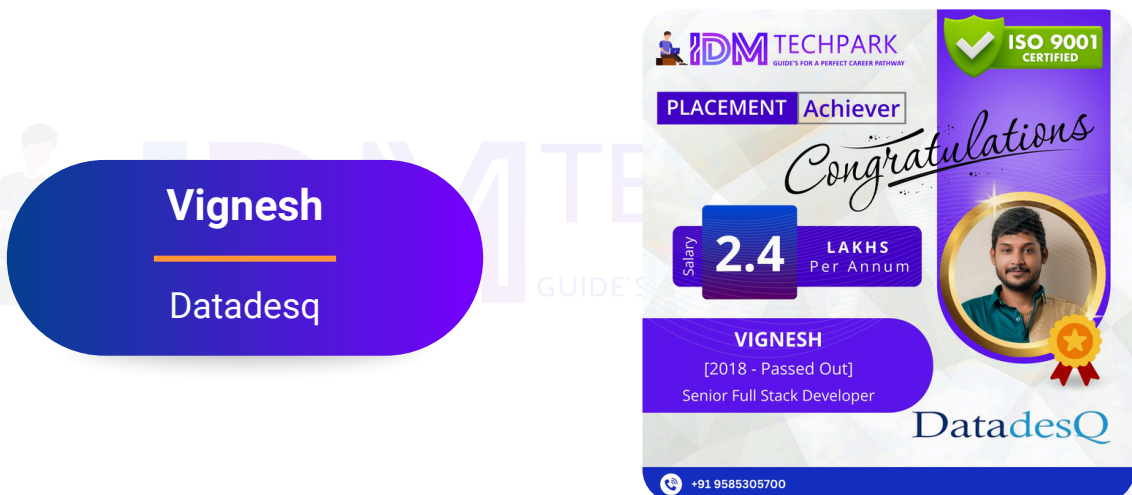
Frutterlabs





Joshwa

Xccessor



Vignesh

Datadesq



Madhumitha

Accenture

Keerthana

AJS



Brindha Boopathi

AJS



Kavinkumar

AJS

Ajithkumar

AJS



Gowthami

Genpact



Surya

ZUCI

Pavithra

Vel Info Tech



Poovitha

Gray Matter



Ramesh

TDT

Siva Sankar
 ST Cloudspark tech



Nabeel Hisham
 VTail



Kalayarasan
 Violet Infotech

4

Our Alumni Working At



MODULE 1

Introduction to C Programming

- Overview of C language
- Basic structure of a C program
- Compiling and executing C programs
- Understanding the role of the C compiler and linker

MODULE 2

Data Types and Variables

- Primitive data types: int, char, float, double, etc.
- Constants and enumerations
- Typecasting in C
- Variable declaration, initialization, and scope
- Type modifiers (short, long, unsigned, etc.)

MODULE 3

Operators in C

- Arithmetic operators
- Relational operators
- Logical operators
- Bitwise operators
- Assignment operators
- Increment/decrement operators
- Conditional (ternary) operator

MODULE 4

Control Statements

- Decision making: if, if-else, switch
- Nested control structures
- The goto statement
- Conditional loops: for, while, do-while
- break, continue control flow statements

MODULE 5

Functions

- Defining and calling functions
- Function arguments and return types
- Function overloading (concepts)
- Recursion in functions
- Scope and lifetime of variables (local vs global variables)
- The static keyword for local variables

MODULE 6

Arrays

- Introduction to arrays and array indexing
- One-dimensional and two-dimensional arrays
- Arrays of pointers
- Dynamic memory allocation for arrays
- Array manipulation using pointers

MODULE 7

Strings in C

- Declaring and initializing strings
- Standard string handling functions (strlen(), strcpy(), strcmp(), etc.)
- Pointers and strings
- String manipulation techniques
- Memory management for strings

MODULE 8

Pointers

- Introduction to pointers and memory addresses
- Pointer arithmetic
- Pointers to arrays and arrays of pointers
- Pointer to functions
- Pointers to structures
- Void pointers

MODULE 9

Dynamic Memory Allocation

- Memory management using malloc(), calloc(), realloc(), and free()
- Understanding memory leaks and avoiding them
- Using sizeof() operator
- Memory fragmentation
- Creating custom memory allocators

MODULE 10

Structures

- Defining structures
- Accessing structure members
- Nested structures
- Arrays of structures
- Pointers to structures
- Passing structures to functions

MODULE 11

Unions

- Defining unions and understanding the difference from structures
- Union initialization and usage
- Memory optimization using unions
- Unions with structures

MODULE 12

File Handling

- File operations: opening, reading, writing, and closing files
- File modes (r, w, a, rb, wb)
- Random access files using fseek() and ftell()
- File error handling and errno
- Binary files vs text files

MODULE 13

Preprocessor Directives

- Using #define, #include, and #if
- Conditional compilation
- Macros and macro functions
- The #pragma directive
- Inline functions and their advantages over macros

MODULE 14

Error Handling in C

- Using errno, perror(), and strerror()
- Handling runtime errors using assert()
- Writing custom error handling mechanisms
- Exception handling concepts in C (error codes, return values)

MODULE 15

Command-Line Arguments

- Introduction to command-line arguments
- argc and argv usage
- Parsing command-line inputs
- Writing programs that use command-line arguments effectively

MODULE 16

Advanced Data Structures

- Linked lists (singly and doubly)
- Stacks and queues using linked lists
- Trees (binary trees, binary search trees)
- Hash tables and hash maps
- Implementing and using heaps and priority queues

MODULE 17

Sorting Algorithms

- Introduction to sorting algorithms
- Bubble sort, selection sort, and insertion sort
- Quick sort and merge sort
- Heap sort and radix sort
- Complexity analysis (Big O, Big Ω , Big Θ)

MODULE 18

Searching Algorithms

- Linear search and binary search
- Hashing techniques
- Searching in trees and graphs
- Search algorithms' complexity analysis

MODULE 19

Bit Manipulation

- Bitwise operators and their usage
- Bit manipulation tricks: setting, clearing, toggling bits
- Bit masking and shifting operations
- Applications of bit manipulation in algorithms

MODULE 20

Advanced Pointer Concepts

- Pointer to pointer (multi-level pointers)
- Pointers to functions
- Pointer arithmetic in multi-dimensional arrays
- Pointer and structure relations
- Pointer to a pointer and memory models

MODULE 21

Multithreading and Concurrency

- Introduction to POSIX threads (pthread)
- Thread creation, joining, and synchronization
- Mutexes, semaphores, and condition variables
- Thread safety and deadlock prevention
- Creating multithreaded applications

MODULE 22

Memory Management and Optimization

- Memory optimization techniques
- Using valgrind and gdb for memory debugging
- Avoiding memory leaks and fragmentation
- Efficient memory allocation strategies
- Buffer overflows and their mitigation

MODULE 23

Networking in C

- Introduction to socket programming (TCP/IP and UDP)
- Creating client-server applications
- Using select() for multiplexing and handling multiple clients
- Raw socket programming
- Network security basics in C (SSL/TLS)

MODULE 24

System-Level Programming

- Interfacing with the operating system using system calls
- Low-level file operations and process management
- Writing device drivers and interacting with hardware
- Shared memory, pipes, and inter-process communication (IPC)
- Managing processes and signals

MODULE 25

Final Project and Best Practices

- Writing modular, reusable, and maintainable C code
- Code optimization techniques (loop unrolling, inlining, etc.)
- Performance profiling and benchmarking
- Documenting and commenting code
- Final C project: Implement a real-world application using all concepts learned

MODULE 26

Introduction to C++

- History of C++ and its evolution
- Basic structure of a C++ program
- Compiling and running a C++ program
- Introduction to cout, cin, and basic I/O operations

MODULE 27

Data Types and Variables

- Primitive data types in C++ (int, float, double, char, etc.)
- Variable declaration and initialization
- Typecasting in C++
- Constants and const keyword
- References and reference variables

MODULE 28

Operators in C++

- Arithmetic operators (+, -, *, /, %)
- Relational operators (==, !=, >, <, >=, <=)
- Logical operators (&&, ||, !)
- Bitwise operators (&, |, ^, ~, <<, >>)
- Assignment operators (=, +=, -=, *=, /=)
- Increment and Decrement operators (++ , --)

MODULE 29

Control Flow in C++

- Conditional statements: if, else, else if, switch
- Nested conditional statements
- Looping constructs: for, while, do-while
- break and continue statements
- The goto statement (use with caution)

MODULE 30

Functions in C++

- Function definition and declaration
- Function parameters and return types
- Pass-by-value vs. pass-by-reference
- Function overloading
- Default arguments
- Inline functions
- Recursion in C++

MODULE 31

Arrays and Strings in C++

- Introduction to arrays: single and multi-dimensional
- Dynamic arrays and memory management
- Introduction to strings in C++ (std::string)
- String operations: concatenation, comparison, length, and searching
- Array of strings and multi-dimensional arrays

MODULE 32

Pointers and Memory Management

- Introduction to pointers in C++
- Pointer arithmetic
- Pointers to functions and arrays
- Memory allocation: new and delete
- Dynamic memory management with new[] and delete[]
- Understanding memory leaks and preventing them

MODULE 33

Object-Oriented Programming (OOP) Fundamentals

- Basic concepts of OOP: classes and objects
- Member functions and data members
- Constructors and destructors
- Access modifiers: public, private, protected
- Introduction to the this pointer
- Defining and creating objects

MODULE 34

Classes and Objects in Depth

- Member functions and function definition outside the class
- Static members in a class
- Access control with getters and setters
- Object-oriented principles: Encapsulation
- Friend functions and friend classes

MODULE 35

Inheritance in C++

- Introduction to inheritance
- Types of inheritance: single, multiple, and multi-level
- The protected keyword in inheritance
- Constructor and destructor in inheritance
- Base class and derived class methods

MODULE 36

Polymorphism in C++

- Introduction to polymorphism
- Function overriding and function overloading
- Virtual functions and dynamic dispatch
- Pure virtual functions and abstract classes
- Interface implementation in C++

MODULE 37

Operator Overloading

- Basics of operator overloading in C++
- Overloading arithmetic, relational, and logical operators
- Overloading the assignment operator
- Overloading input/output stream operators (<<, >>)
- Use cases and best practices for operator overloading

MODULE 38

Friend Functions and Classes

- Understanding friend functions
- Friend classes and the use cases
- Access control and friendships
- Benefits and limitations of friend functions in C++

MODULE 39

Templates in C++

- Introduction to templates
- Function templates and class templates
- Template specialization
- Template argument deduction
- Variadic templates

MODULE 40

Exception Handling in C++

- Introduction to exceptions and error handling
- try, catch, and throw keywords
- Handling multiple exceptions
- Creating custom exception classes
- Exception safety and RAI (Resource Acquisition Is Initialization)

MODULE 41

The Standard Template Library (STL) Basics

- Introduction to STL and its components
- Containers in STL: vector, list, deque, set, map, stack, queue
- Iterators in STL
- STL algorithms: sort(), find(), reverse(), etc.
- STL allocators

MODULE 42

STL - Advanced Containers

- Associative containers in STL: set, map, multiset, multimap
- Implementing custom comparators with STL containers
- Using unordered containers: unordered_set, unordered_map
- Set operations and set theory in C++

MODULE 43

STL Algorithms and Function Objects

- Introduction to function objects (functors)
- Advanced STL algorithms: `for_each()`, `transform()`, `accumulate()`, `count()`
- Custom function objects in algorithms
- Sorting algorithms in STL (`sort()`, `stable_sort()`, etc.)

MODULE 44

Memory Management and Smart Pointers

- Introduction to smart pointers: `unique_ptr`, `shared_ptr`, and `weak_ptr`
- Memory management in modern C++ with smart pointers
- `std::allocator` and custom memory allocators
- `std::make_unique()` and `std::make_shared()`
- Preventing memory leaks with smart pointers

MODULE 45

Multithreading in C++

- Introduction to multithreading with the C++11 Thread library
- Creating threads with `std::thread`
- Thread synchronization using `mutex` and `lock_guard`
- Condition variables and producer-consumer problems
- Thread safety and race conditions

MODULE 46

Advanced C++ Features

- Move semantics and `std::move()`
- Rvalue references and lvalue references
- Perfect forwarding with `std::forward()`
- Copy and move constructors
- `std::unique_ptr` and `std::shared_ptr` for ownership semantics

MODULE 47

File Handling in C++

- Introduction to file I/O in C++
- File streams: fstream, ifstream, ofstream
- Reading and writing binary files
- File operations: opening, closing, and seeking
- Error handling in file I/O

MODULE 48

Design Patterns in C++

- Introduction to design patterns and their importance
- Singleton pattern in C++
- Factory pattern
- Observer pattern
- Strategy and Command patterns

MODULE 49

C++11, C++14, and C++17 Features

- New features in C++11: auto, nullptr, range-based for, lambda functions
- C++14 improvements and features
- C++17 features: std::optional, std::filesystem, structured bindings
- Usage of new standard library features

MODULE 50

Final Project and Best Practices

- Implementing a real-world C++ application
- Code structure and organization
- Performance optimization techniques
- Code review, debugging, and testing best practices
- Writing efficient, maintainable, and reusable C++ code

Thank You

+91 9585305700

www.idmtechparkcoimbatore.com

www.idmtechparkerode.com