

COMPANY PROFILE

Warm Welcome to Join Hands with Our **REDO** Chain!!!

Founded in 2020, **REDO Skill Maturing Workshop**, the name itself has its solution which was started to recycle your skills which you left as it was some days ago. You cannot get your wasted time back, but you can get another chance to redo your process of implementing your dream work from now and with us to grow logically even more than better and stronger. We **never try to educate** our candidate's mind **by commanding, grading or by triggering their mind by promising for fake placements**. Rather we create passion on the work, **we make them understand "why they came to us, what they learn from and how can they make money out of it" if they spend their valuable time with us.**

Our Secret of Success and Our Strength Is,

"We Never Work for Feedbacks..., We Work for Results..."

Important Things About Us:

- ✓ **Origin(al):** As We represent us as **"A Proud Training Unit of Embuzz Technologies Private Limited"**, We Don't Give Fake Certificates or Support Indiscipline Peoples
- ✓ **Real Coaching / Training Content:** We Don't have Time / Money / Energy Wasting Fake Syllabus
- ✓ **Ability After Coaching / Training:** We Don't Blabber About Placement Assurance, Rather Definitely Make Them Perform Logically and Make Candidates to Decide Their Career in Their Own Way
- ✓ **Our Approach:** We'll be Kind As Well As Aggressive Occasionally At Sessions, Depends On The Candidate's Learning Speed and Capability. **We Focus on Best Results Rather Than Best Feedbacks.** Our Best / Old Book Methods (Punctuality / Neatly Dressed / Notes Making / Communication / Discipline / etc...) Will Look Worst for Inexperienced Candidates but We Definitely Want Them to Follow Our Guidelines Without Any Excuses and Come Out Of Their Insecure Zone Technically As Well As Behaviourally Too To Grow Them Immensely.
- ✓ **We Are Unique and Very Content Specific In Delivering Contents, Definitely Candidates Will Feel While Sessions For Sure.**





Embedded System Design And Development

Training Topics

Duration: 30 Hours to 45 Hours

This syllabus can be done for with slight modification Arduino, ESP32, AVR, ARM, PIC, STM32.

8 - BIT OR 32 - BIT TRAINING (including Automation / Robotics / Wireless Communication, IoT, Data Science, AI):

<u>S.No</u>	<u>Topic</u>
1	Embedded System Basics
2	C-Language
3	EEPROM Programming
4	Input & Output Operations
5	Serial Communication Programming
6	LCD Interfacing
7	Analog to Digital Converter
8	Timing & Counting Operations
9	SPI Protocol
10	I ² C Protocol
11	Interfacing RFID
12	ZIGBEE Communication
13	GSM Communication
14	Robotics
15	Sensors Interfaces (5 Sensors Minimum)
16	Hardware Interfacing Program
17	Automation and Artificial Intelligence
18	SCADA Basics
19	Synthesis tools
20	Trouble shooting processes



Corporate C Programming

Week 1: Introduction to C and Basic Syntax (8 hours)

Day 1: C Programming Basics (4 hours)

1. Introduction to C Language
 - History and evolution of C
 - Applications of C
 - Setting up a development environment (IDE, compiler setup)
2. Basic Structure of a C Program
 - Writing and compiling a simple C program
 - main() function
 - Header files and standard libraries
 - Comments in C
3. Basic Input and Output
 - printf() and scanf() functions
 - Format specifiers
4. Variables and Data Types
 - Primitive data types (int, char, float, double)
 - Constants and literals
 - Type casting and conversions

Day 2: Operators and Control Structures (4 hours)

1. Operators in C
 - Arithmetic, relational, logical, and bitwise operators
 - Unary and assignment operators
2. Control Flow
 - Conditional statements: if, else, switch-case
 - Looping constructs: for, while, do-while



- Nested loops and conditional statements
-

Week 2: Functions, Arrays, and Pointers (8 hours)

Day 3: Functions in C (4 hours)

1. Function Basics

- Definition and declaration of functions
- Function arguments and return types

2. Function Scope and Lifetime

- Local vs Global variables
- Static variables

3. Recursion

- Concept of recursion
- Writing recursive functions (factorial, Fibonacci)

Day 4: Arrays and Pointers (4 hours)

1. Arrays in C

- Declaring and initializing arrays
- Accessing array elements
- Multi-dimensional arrays

2. Pointers in C

- Introduction to pointers and memory addresses
 - Pointer arithmetic
 - Pointers and arrays relationship
 - Dynamic memory allocation using malloc(), calloc(), and free()
-



Week 3: Advanced Concepts (8 hours)

Day 5: Strings and Structures (4 hours)

1. Strings in C

- Declaring and initializing strings
- String manipulation functions (strcpy(), strlen(), strcat())
- Input and output of strings

2. Structures

- Defining and initializing structures
- Nested structures
- Array of structures
- Structures and functions

Day 6: File Handling and Advanced Pointers (4 hours)

1. File Handling in C

- Opening and closing files (fopen(), fclose())
- File reading and writing (fscanf(), fprintf(), fgets(), fputs())
- File pointers and binary files

2. Advanced Pointer Concepts

- Function pointers
- Pointer to structure

Week 4: Data Structures, Algorithms, and Industry Practices (6 hours)

Day 7: Data Structures (4 hours)

1. Linked Lists

- Singly linked list: creating, inserting, and deleting nodes
- Doubly linked list: structure and operations
- Circular linked list



2. Stacks and Queues

- Stack operations (Push, Pop, Peek)
- Queue operations (Enqueue, Dequeue, Front)

Day 8: Algorithms and Industry Best Practices (2 hours)

1. Basic Algorithms

- Sorting algorithms: Bubble sort, Selection sort, Insertion sort
- Searching algorithms: Linear search, Binary search

2. Industry Best Practices

- Code documentation and commenting
 - Writing efficient and maintainable code
 - Debugging techniques (using gdb or IDE debugger)
 - Version control basics (using Git)
-

Week 5: Real-world Applications and Project Work (8 hours)

Day 9: Project Implementation (4 hours)

1. Mini Project Development

- Select a simple project based on course concepts (e.g., Student management system, Library management system)
- Step-by-step development using functions, file handling, and data structures

2. Code Review and Debugging

- Troubleshooting and debugging the project
- Performance optimization

Day 10: Final Project and Industry Insights (4 hours)

1. Final Project Presentation

- Finalize the mini project and present it to peers or instructors
- Discuss challenges faced and how to overcome them



2. Industry Insights

- Common C programming challenges in the industry
- Best practices for code testing and optimization
- Interview preparation: common C programming questions

Additional Learning Topics (optional depending on time)

- Multithreading (2-3 hours)
 - Introduction to threads in C
 - Creating and managing threads
 - Synchronization and mutexes
 - Networking in C (2-3 hours)
 - Basics of networking concepts (TCP/IP)
 - Socket programming in C (client-server model)
-



Corporate C++ Programming

Week 1: Introduction to C++ and Basic Syntax (8 hours)

Day 1: Introduction to C++ and Basic Concepts (4 hours)

1. Introduction to C++
 - History and evolution of C++
 - Differences between C and C++
 - Applications of C++ in the industry
2. Setting up Development Environment
 - Installing IDEs (e.g., Visual Studio, CLion, Code::Blocks)
 - Compiling and running a simple C++ program
 - Understanding the basic structure of a C++ program
3. Basic Input and Output
 - cout and cin for console output and input
 - Using format specifiers
 - Input validation and handling

Day 2: Data Types, Operators, and Control Flow (4 hours)

1. C++ Data Types and Variables
 - Primitive data types: int, float, double, char, bool
 - Constants and const keyword
 - Type casting and conversions
2. Operators in C++
 - Arithmetic, relational, logical, and bitwise operators
 - Unary and assignment operators
 - Precedence and associativity
3. Control Flow Statements
 - if, else, else if, switch statements
 - Loops: for, while, do-while



- Nested loops and conditions
-

Week 2: Functions, Object-Oriented Programming Basics (8 hours)

Day 3: Functions and Recursion (4 hours)

1. Functions in C++

- Function definition, declaration, and calling
- Parameters: pass-by-value vs. pass-by-reference
- Return types and overloading functions

2. Recursion

- Understanding recursion with examples (factorial, Fibonacci)
- Base case and recursive case
- Tail recursion and its benefits

Day 4: Introduction to Object-Oriented Programming (OOP) (4 hours)

1. Classes and Objects

- Defining classes and objects
- Constructors and destructors
- Member functions and access specifiers (public, private, protected)

2. Encapsulation

- Access control using getters and setters
- Private, protected, and public members

3. Inheritance

- Inheriting from a base class
 - Overriding member functions
 - Constructors and destructors in inheritance
-



Week 3: Advanced OOP, Templates, and STL (8 hours)

Day 5: Advanced Object-Oriented Programming (4 hours)

1. Polymorphism

- Function overloading and operator overloading
- Virtual functions and dynamic polymorphism
- Abstract classes and pure virtual functions
- Interfaces in C++

2. Multiple Inheritance and Inheritance Hierarchy

- Handling multiple inheritance and ambiguities
- Virtual inheritance
- Inheriting from multiple base classes

Day 6: C++ Templates and Standard Template Library (STL) (4 hours)

1. C++ Templates

- Function templates
- Class templates
- Template specialization

2. Introduction to STL

- Overview of containers: vector, list, deque, map, set, stack, queue
- Iterators in STL
- Using algorithms with STL containers

Week 4: Exception Handling, File I/O, and Industry Practices (6 hours)

Day 7: Exception Handling and File I/O (4 hours)

1. Exception Handling in C++

- Basics of try, catch, throw
- Handling multiple exceptions



- Custom exception classes

2. File Handling in C++

- Working with text and binary files using fstream
- Reading and writing data to files (ifstream, ofstream, fstream)
- Handling file errors and exceptions

Day 8: Industry Best Practices and Debugging (2 hours)

1. Code Optimization and Best Practices

- Efficient use of memory (dynamic memory management)
- Writing clean, readable, and maintainable code
- Commenting and documenting code

2. Debugging Techniques

- Using debugging tools (GDB or IDE debuggers)
- Identifying and fixing logical errors
- Memory management issues (e.g., memory leaks, dangling pointers)

Week 5: Advanced Topics and Final Project (8 hours)

Day 9: Advanced C++ Concepts (4 hours)

1. Smart Pointers

- Understanding raw pointers vs. smart pointers
- Using `std::unique_ptr`, `std::shared_ptr`, and `std::weak_ptr`
- Memory management with smart pointers

2. Lambda Functions

- Basic syntax and usage of lambda functions
- Capturing variables in lambdas
- Practical applications of lambdas in STL algorithms



3. Multithreading in C++

- Introduction to multithreading and concurrency
- Using `std::thread` to create threads
- Synchronization: Mutexes, locks, and condition variables

Day 10: Real-World Project Implementation (4 hours)

1. Project Planning and Design

- Select a real-world application project (e.g., a simple banking system, inventory management, or a mini game)
- Identify required features (OOP concepts, file I/O, STL usage)
- Design the system and break down into manageable tasks

2. Project Development

- Implement the project step by step
- Apply concepts like classes, inheritance, polymorphism, STL, and exception handling
- Final debugging and optimization

Additional Optional Topics (if time permits)

- Networking in C++ (2-3 hours)
 - Basics of networking (TCP/IP)
 - Socket programming in C++: Client-server model
 - Using `boost::asio` for network programming
 - Design Patterns in C++ (2-3 hours)
 - Common design patterns (Singleton, Factory, Observer, Strategy)
 - Practical implementation of design patterns in C++
-



Final Project (Optional Extension)

The final project serves as a comprehensive test of the learner's understanding and application of the concepts covered. It involves:

1. **Project Selection:** A real-world problem like a simple banking system, library management system, or game.
 2. **Project Development:** Designing the system with C++ features such as OOP, file handling, exception handling, and multithreading.
 3. **Code Review and Presentation:** Reviewing the project code, discussing design decisions, and presenting the project to peers or mentors.
-



SYLLABUS FOR JAVA

MODULE-1 (5 Hrs)

INTRODUCTION TO JAVA

Features of Java

JDK and JRE

Program structure and Syntax

Setting up Development Environment (Eclipse/VS code/IntelliJ).

JAVA FUNDAMENTALS

Data Types, Variables and Literals

Operators and its Expressions

Type Casting/Type Conversion

Control Statements: if-else, Switch, else-if ladder

Looping Statements: for, While, do-while

Arrays and Multidimensional arrays.

MODULE-2 (10 Hrs)

STRING HANDLING

String Class and Methods

StringBuffer and StringBuilder

String Manipulation and Formatting

EXCEPTION HANDLING

Types of Exceptions: Checked Vs Unchecked

try, catch, finally, throw and throws

Custom Exceptions

MODULE-3 (5 Hrs)

OOP IN JAVA

Classes and Objects: Constructors, Methods, 'this' keyword

Encapsulation

Inheritance: 'super' keyword, Method overloading and overriding

Polymorphism: Compile time Vs Runtime

Abstraction: Abstract classes and Interfaces.

MODULE-4 (5 Hrs)

MULTITHREADING AND CONCURRENCY

Thread class and Runnable Interface

Thread Lifecycle and Methods

Synchronization and Inter-thread Communication

Executors and Concurrency Utilities.



MODULE-5 (5 Hrs)

FILE HANDLING

File I/O using 'File', 'FileReader', 'FileWriter'

Buffered Streams and Serialization

New I/O: Channels and Buffers

Stream API.



SYLLABUS FOR PYTHON

COURSE CONTENT

MODULE-1 5Hrs

INTRODUCTION TO PYTHON

Python role in programming and industries

Installing Python and setting up the environment(IDLE, VS Code, Jupyter)

Variables, Datatypes, and Type casting

Input and Output in Python.

MODULE-2 5Hrs

CONTROL STRUCTURES

Conditional Statements: if, if-else and nested if-else statements

Operators and its Expressions

Looping statements: for, while and Nested loops.

Iterating over sequences(strings, lists, etc.,)

MODULE-3 10Hrs

OOP WITH PYTHON

Classes and objects

Methods, attributes and the 'self' parameter

Constructor and Destructor

Inheritance, Polymorphism and Encapsulation

Method Overloading and Overriding

Abstract classes and Interfaces.

MODULE-4 5Hrs

FUNCTIONS

Defining and Calling functions

Arguments and return values

Lambda functions.

MODULES AND PACKAGES

Built-in modules(os, math, random)

Creating and importing User defined modules

Introduction to pip and External packages.

MODULE-5 5Hrs

DATA STRUCTURES IN PYTHON

Strings, Lists, Tuples.

Sets and Dictionaries.

EXCEPTION HANDLING

try, except and finally blocks



REDO Skill Maturing Workshop



Raising exceptions.

FILE HANDLING

Working with CSV and JSON files.



A Prestigious Training Unit From **Team Embuzz!!!**



M +91 70107 84 824
E redoskillindia@gmail.com
W www.redoindia.in