# NEW EMPLOYEE TRAINING PROGRAM: PART I

## 🎯 GENERAL OBJECTIVES

- Master modern C++ (C++11/14/17)  
- Understand and apply Object-Oriented Programming (OOP)  
- Use and write CMake projects  
- Solid knowledge of Data Structures and Algorithms (basic to intermediate)

## 📅 DURATION: 12 WEEKS

### Week 1: Introduction to Modern C++ (Basic C++11) & Environment Setup

\*\*Theory:\*\*

- Environment setup: Vscode, Visual Studio(Community)

- Variables, data types, operators, control flow

- Functions, scope, pass by value/reference

- `auto`, `nullptr`, `enum class`, `range-based for`

- Smart Pointers: `unique\_ptr`, `shared\_ptr`

\*\*Practice:\*\*

- Exercises with numbers, strings, functions, smart pointers

### Week 2: OOP – Overview & Basic Classes

\*\*Theory:\*\*

- Object-Oriented Programming (OOP) concepts: class, object, encapsulation

- Creating and using `class`, `struct`

- Constructors, destructors, member functions

- Static members, friend functions

- `this` pointer

\*\*Practice:\*\*

- Create classes: `Student`, `Rectangle`, `BankAccount`

- Manipulate object lists

### Week 3: Advanced OOP – Inheritance, Polymorphism, Abstraction

\*\*Theory:\*\*

- Inheritance, Overriding, `virtual` and `override`

- Abstract classes & pure virtual functions

- Polymorphism via base class pointers

- Access specifiers: `private`, `protected`, `public`

- Composition vs Inheritance

\*\*Practice:\*\*

- Zoo animal management using inheritance

- Vehicle management system (Vehicle, Car, Truck,...)

- Polymorphism with base class pointers

### Week 4: C++14/17 & Advanced Templates

\*\*Theory:\*\*

- `constexpr`, `std::move`, `std::optional`, `std::variant`

- Lambda functions, template function/class

- `if constexpr`, fold expressions

\*\*Practice:\*\*

- Create generic `Math<T>` template

- Manage optional config and variant type

### Week 5: CMake & Project Organization

\*\*Theory:\*\*

- CMake basics: writing CMakeLists.txt

- Managing multi-file projects

- Building static/shared libraries

\*\*Practice:\*\*

- Create a small project with CMake

- Split project into core, utils, app modules

### Week 6–9: Data Structures & Algorithms

\*\*Theory:\*\*

- Basic structures: Array, Linked List, Stack, Queue

- Sorting: Bubble, Merge, Quick

- Binary Tree, Binary Search Tree

- Hash table

- Graph: BFS, DFS

\*\*Practice:\*\*

- Implement data structures in C++

### Week 10: STL & Combining OOP with Data Structures

\*\*Theory:\*\*

- `vector`, `map`, `set`, `unordered\_map`, `priority\_queue`

- Iterators, Lambda with std::sort, std::find

- Custom comparator in STL

- Combining OOP and STL

\*\*Practice:\*\*

- Ticket booking system using `map` + OOP

### Week 11: Mini Project

\*\*Suggestions:\*\*

- Library management system

- Simulated flight booking system

- Smart file classification manager

### Week 12: Evaluation & Development Planning

- Knowledge review

- Team/individual project demo

- Feedback and personal development planning

## 🔧 TOOLS & REFERENCES

- IDEs: VSCode, CLion or Visual Studio- Books: C++ Primer (5th Edition), Effective Modern C++, Data Structures and Algorithms in C++  
- Reference: cppreference.com