TECHNOCATION FREELANCING TRAINING INSTITUTE & SOFTWARE HOUSE

Professional Object-Oriented Programming (OOP) Course Outline

Module 1: Introduction to Object-Oriented Programming (OOP)

- What is OOP? Principles & Benefits
- Procedural vs. Object-Oriented Programming
- Popular OOP Languages (Java, Python, C++, PHP)
- Setting Up the Development Environment

Module 2: Core OOP Concepts

- Classes & Objects
- Attributes (Fields, Properties) & Methods
- Encapsulation & Data Hiding
- Constructors & Destructors

Module 3: Inheritance & Polymorphism

- Understanding Inheritance & Superclasses
- Method Overriding & Overloading
- The "super" Keyword & Multiple Inheritance
- Abstract Classes & Interfaces

Module 4: Encapsulation & Data Protection

- Private, Public, and Protected Access Modifiers
- Getters & Setters for Data Encapsulation
- Encapsulation Best Practices

Module 5: Abstraction & Interfaces

- What is Abstraction?
- Abstract Classes vs. Interfaces
- Implementing Interfaces in Java, Python, C++
- Real-World Applications of Abstraction

Module 6: Polymorphism & Dynamic Method Binding

- Static vs. Dynamic Polymorphism
- Function Overloading & Operator Overloading
- Virtual Functions & Late Binding (C++)
- Duck Typing & Polymorphism in Python

Module 7: Object Relationships & Associations

- One-to-One, One-to-Many, Many-to-Many Relationships
- Composition vs. Aggregation
- Dependency Injection & Loose Coupling
- Design Patterns for Object Relationships

Module 8: Error Handling & Exception Management

- Types of Errors & Exceptions
- Exception Handling Mechanisms (try, catch, finally)
- Custom Exception Classes
- Best Practices for Exception Handling

Module 9: File Handling & Serialization

- Working with Files in OOP (Reading/Writing)
- Serialization & Deserialization
- Storing Objects in Files & Databases
- Object Persistence & Data Storage

Module 10: Design Patterns in OOP

- Introduction to Design Patterns
- Creational Patterns (Singleton, Factory, Builder)
- Structural Patterns (Adapter, Decorator, Proxy)
- Behavioral Patterns (Observer, Strategy, Command)

Module 11: OOP in Real-World Applications

- OOP in Web Development (PHP, JavaScript)
- OOP in Game Development (Unity C#, Unreal C++)
- OOP in Mobile Apps (Java/Kotlin for Android, Swift for iOS)
- OOP in Machine Learning & Data Science (Python)

Module 12: Advanced OOP & Best Practices

- Object-Oriented Database Management
- Metaprogramming & Reflection
- Unit Testing & Test-Driven Development (TDD)
- Best Practices for Writing Clean & Maintainable OOP Code

Final Project & Certification Preparation

- Hands-on Project: Developing an OOP-Based Application
- Debugging & Refactoring OOP Code
- Preparing for OOP-Related Certification Exams
- Industry Best Practices & Career Guidance