

Lesson Plan: Basic ObjectOriented Programming Concepts in C

Session 1: Introduction to OOP

1. Introduction to OOP (30 minutes)

- What is OOP?
 - Definition and importance
- Four Pillars of OOP
 - Encapsulation
 - Inheritance
 - Polymorphism
 - Abstraction

Session 2: Classes and Objects

1. Classes and Objects (45 minutes)

- Defining Classes
 - Class syntax
 - Fields and methods
- Creating Objects
 - Instantiation
 - Accessing class members
- Constructors
 - Default and parameterized constructors

2. HandsOn Practice (45 minutes)

- Exercise 1: Create a `Car` class with fields like `Make`, `Model`, and `Year`. Add a method `DisplayInfo` to print these details.
- Exercise 2: Instantiate multiple `Car` objects and call the `DisplayInfo` method for each.
- Exercise 3: Add a parameterized constructor to the `Car` class and create objects using this constructor.

Session 3: Encapsulation and Properties

1. Encapsulation (30 minutes)

- Access Modifiers
 - public, private, protected, internal
- Properties
 - get and set accessors
 - Auto Implemented properties

2. HandsOn Practice (1 hour)

- Exercise 1: Modify the `Car` class to use private fields and public properties for `Make`, `Model`, and `Year`.
- Exercise 2: Add validation in the set accessor of the `Year` property to ensure it is a valid year.
- Exercise 3: Create a `BankAccount` class with private fields for `balance` and public properties to deposit and withdraw money.

Session 4: Inheritance

1. Inheritance (45 minutes)

- Base and Derived Classes
 - Syntax and usage
- Method Overriding
 - virtual and override keywords
- Base Keyword
 - Calling base class constructors and methods

2. HandsOn Practice

- Exercise 1: Create a base class `Vehicle` with fields like `Speed` and `Capacity`. Derive a `Car` class from `Vehicle` and add additional fields.
- Exercise 2: Override a method `DisplayInfo` in the `Car` class to include additional details.
- Exercise 3: Use the `base` keyword to call the base class constructor from the derived class constructor.

Session 5: Polymorphism

1. Polymorphism (45 minutes)

- Method Overloading
 - Definition and examples
- Method Overriding
 - Recap and deeper dive

- Abstract Classes and Methods
 - Definition and usage
- Interfaces
 - Definition and implementation

2. HandsOn Practice

- Exercise 1: Create a class `MathOperations` with overloaded methods `Add` to handle different data types (int, double).
- Exercise 2: Create an abstract class `Shape` with an abstract method `CalculateArea`. Derive classes `Circle` and `Rectangle` from `Shape` and implement the method.
- Exercise 3: Define an interface `IDrawable` with a method `Draw`. Implement this interface in classes `Circle` and `Rectangle`.

Session 6: Abstraction and Advanced Topics

1. Abstraction (30 minutes)

- Abstract Classes vs. Interfaces
 - Comparison and use cases
- Sealed Classes and Methods
 - Definition and usage

2. Advanced Topics (30 minutes)

- Static Classes and Members
 - Definition and usage
- Namespaces
 - Organizing code with namespaces

3. HandsOn Practice (30 minutes)

- Exercise 1: Create a static class `Utility` with static methods like `PrintMessage` and `AddNumbers`.
- Exercise 2: Organize your classes into different namespaces and use them in a console application.
- Exercise 3: Create a sealed class `FinalClass` and attempt to derive from it to understand the restriction.

Project Description for assignment

Build a simple console school management application demonstrating OOP concepts