DotNet Technology (CACS302) - Complete Notes

Unit 1: Introducing C# and .NET Framework

Object-Oriented Programming

C# is a fully object-oriented language. Key concepts include Encapsulation, Inheritance, Polymorphism, and Abstraction.

Type Safety

Type safety ensures that variables are accessed in a valid way. C# prevents invalid casts and memory errors.

Memory Management

Memory is automatically managed by Garbage Collector (GC). It removes unused objects from memory.

CLR (Common Language Runtime)

CLR is the execution engine of .NET. It provides services like memory management, exception handling, and security.

.NET Framework Architecture

It consists of CLR and Base Class Library (BCL). CLR executes code, BCL provides reusable classes.

.NET Standard

.NET Standard defines a set of APIs that all .NET implementations must follow.

Unit 2: C# Language Basics

Data Types

Value types: int, float, char. Reference types: string, object.

Variables

Variables store data. Example: int x = 10;

Operators

Arithmetic (+, -, *), Relational (==, !=), Logical (&&, ||).

Control Statements

if, else, switch for decision making. Loops include for, while, do-while.

Arrays

Collection of same data types. Example: int[] arr = {1,2,3};

Namespaces

Used to organize code. Example: using System;

Unit 3: Creating Types in C#

Classes and Objects

Class is a blueprint. Object is an instance.

Constructors

Used to initialize objects. Types: Default, Parameterized.

Inheritance

Allows one class to inherit another. Syntax: class B : A

Polymorphism

Method overloading and overriding.

Interfaces

Defines contract. A class must implement all methods.

Abstract Classes

Cannot be instantiated. Used as base classes.

Generics

Allow type-safe data structures. Example: List