DOT.NET : MODULE 1 : C#

UNIT 1 C# and the NET Framework

1. Before NET
2. Windows Programming in the Late 1990s
3. Goals for the Next-Generation Platform
4. Enter Microsoft NET
5. Components of the NET Framework
6. An Improved Programming Environment
7. Compiling to the Common Intermediate Language (CIL)
8. Compiling to Native Code and Execution
9. Overview of Compilation and Execution
10. The Common Language Runtime (CLR)
11. The Common Language Infrastructure (CLI)
12. Important Parts of the CLI

UNIT 2 Overview of C# Programming

1. A Simple C# Program
2. More About Simple Program
3. Identifiers and Keywords
4. Naming Conventions
5. Keywords
6. Main: The Starting Point of a Program
7. Whitespace
8. Statements
9. Simple Statements
10. Blocks
11. Text Output from a Program
12. Write
13. Write Line
14. The Format String
15. Substituting Values
16. Multiple Markers and Values
17. Comments: Annotating the Code
18. More About Comments
19. Documentation Comments
20. Summary of Comment Types

UNIT 3 Types, Storage, and Variables

1. A C# Program Is a Set of Type Declarations
2. A Type Is a Template
3. Instantiating a Type
4. Data Members and Function Members

5. Types of Members
6. Predefined Types
7. More About the Predefined Types
8. User-Defined Types
9. The Stack and the Heap
10. The Stack
11. The Heap
12. Value Types and Reference Types
13. Storing Members of a Reference Type Object
14. Categorizing the C# Types
15. Variables
16. Variable Declarations
17. Multiple-Variable Declarations
18. Using the Value of a Variable

UNIT 4 Classes: The Basics

1. Overview of Classes
2. A Class Is an Active Data Structure
3. Programs and Classes: A Quick Example
4. Declaring a Class
5. Class Members
6. Fields
7. Methods
8. Creating Variables and Instances of a Class
9. Allocating Memory for the Data
10. Combining the Steps
11. Instance Members
12. Access Modifiers
13. Private and Public Access
14. Accessing Members from Inside the Class
15. Accessing Members from Outside the Class
16. Putting It All Together

UNIT 5 Methods

1. The Structure of a Method
2. Code Execution in the Method Body
3. Local Variables
4. Local Variables Inside Nested Blocks
5. Flow-of-Control
6. Method Invocations
7. Return Values
8. The Return Statement and Void Methods
9. Parameters
10. Formal Parameters

11. Actual Parameters
12. Value Parameters
13. Reference Parameters
14. Output Parameters
15. Parameter Arrays
16. Method Invocation
17. Expanded Form
18. Arrays As Actual Parameters
19. Summary of Parameter Types
20. Stack Frames
21. Recursion
22. Method Overloading

UNIT 6 More About Classes

1. Class Members
2. Order of Member Modifiers
3. Instance Class Members
4. Static Fields
5. Accessing Static Members from Outside the Class
6. Example of a Static Field
7. Lifetimes of Static Members
8. Static Function Members
9. Other Static Class Member Types
10. Constants
11. Constants Are Like Statics
12. Local Constants
13. Properties
14. Property Declarations and Accessors
15. A Property Example
16. Using a Property
17. Properties and Associated Fields
18. Performing Other Calculations
19. Read-Only and Write-Only Properties
20. A Computed, Read-Only Property Example
21. Example of Properties and Databases
22. Static Properties
23. Instance Constructors
24. Constructors with Parameters
25. Default Constructors
26. Static Constructors
27. Example of a Static Constructor
28. Accessibility of Constructors
29. Finalizers
30. Calling the Finalizer
31. Comparison of Constructors and Finalizers

32. The read-only Modifier
33. The this Keyword
34. Indexers
35. What Is an Indexer?
36. Indexers and Properties
37. Declaring an Indexer
38. The set Accessor
39. The get Accessor
40. More About Indexers
41. Declaring the Indexer for the Employee Example
42. An Additional Example of an Indexer
43. Indexer Overloading
44. Access Modifiers on Accessors
45. Partial Classes

UNIT 7 Classes and Inheritance

1. Class Inheritance
2. Accessing the Inherited Members
3. All Classes are Derived From Class object
4. Hiding Members of a Base Class
5. Base Access
6. Using References to a Base Class
7. Virtual and Override Methods
8. Overriding a Method Marked override
9. Constructor Execution
10. Constructor Initializers
11. Class Access Modifiers
12. Inheritance Between Assemblies
13. Member Access Modifiers
14. Regions Accessing a Member
15. Public Member Accessibility
16. Private Member Accessibility
17. Protected Member Accessibility
18. Internal Member Accessibility
19. Protected Internal Member Accessibility
20. Summary of Member Access Modifiers
21. Abstract Members
22. Abstract Classes
23. Example of an Abstract Class and an Abstract Method
24. Sealed Classes
25. External Methods

UNIT 8 Expressions and Operators

1. Expressions

2. Literals
3. Integer Literals
4. Real Literals
5. Character Literals
6. String Literals
7. Order of Evaluation
8. Precedence
9. Associativity
10. Parenthesized Expressions
11. Simple Arithmetic Operators
12. The Remainder Operator
13. Relational and Equality Comparison Operators
14. Comparison and Equality Operations
15. Increment and Decrement Operators
16. Conditional Logical Operators
17. Logical Operators
18. Shift Operators
19. Assignment Operators
20. Compound Assignment
21. The Conditional Operator
22. Unary Arithmetic Operators
23. User-Defined Type Conversions
24. Explicit Conversion and the Cast Operator
25. Operator Overloading
26. Restrictions on Operator Overloading
27. Example of Operator Overloading
28. The type of Operator

UNIT 9 Statements

1. What Are Statements?
2. Expression Statements
3. Flow-of-Control Statements
4. The if Statement
5. The if else Statement
6. The switch Statement
7. A Switch Example
8. More on the switch Statement
9. Switch Labels
10. The while Loop
11. The do Loop
12. The for Loop
13. Scope of Variables in a for Statement
14. Multiple Expressions in the Initializer and Iterator
15. Jump Statements
16. The break Statement

17. The continue Statement
18. Labeled Statements
19. Labels
20. The Scope of Labeled Statements
21. The goto Statement
22. The goto Statement Inside a switch Statement
23. The using Statement
24. Packaging Use of the Resource
25. Example of the using Statement
26. Multiple Resources and Nesting
27. Another Form of the using Statement
28. Other Statements

UNIT 10 Namespaces and Assemblies

1. Referencing Other Assemblies
2. The mscorlib Library
3. Namespaces
4. Namespace Names
5. More About Namespaces
6. Namespaces Spread Across Files
7. Nesting Namespaces
8. The using Directives
9. The using Namespace Directive
10. The using Alias Directive
11. The Structure of an Assembly
12. The Identity of an Assembly
13. Strongly Named Assemblies
14. Creating a Strongly Named Assembly
15. Private Deployment of an Assembly
16. Shared Assemblies and the GAC
17. Installing Assemblies into the GAC
18. Side-by-Side Execution in the GAC
19. Configuration Files
20. Delayed Signing

UNIT 11 Exceptions

1. What Are Exceptions?
2. The try Statement
3. Handling the Exception
4. The Exception Classes
5. The catch Clause
6. Examples Using Specific catch Clauses
7. The catch Clauses Section
8. The finally Block

9. Finding a Handler for an Exception
10. Searching Further
11. General Algorithm
12. Example of Searching Down the Call Stack
13. Throwing Exceptions
14. Throwing Without an Exception Object

UNIT 12 Structs

1. What Are Structs?
2. Structs Are Value Types
3. Assigning to a Struct
4. Constructors and Finalizers
5. Instance Constructors
6. Static Constructors
7. Summary of Constructors and Finalizers
8. Field Initializers
9. Inheritance
10. Boxing and Unboxing
11. Structs As Return Values and Parameters
12. Additional Information About Structs

UNIT 13 Enumerations

1. Enumerations
2. Underlying Types and Values
3. Setting the Underlying Type
4. Setting Explicit Values for the Members
5. Implicit Member Numbering
6. Bit Flags
7. The Flags Attribute
8. Example Using Bit Flags
9. More About Enums

UNIT 14 Arrays

1. Arrays
2. Definitions
3. Important Details
4. Types of Arrays
5. An Array As an Object
6. One-Dimensional and Rectangular Arrays
7. Declaring a One-Dimensional Array or a Rectangular Array
8. Instantiating a One-Dimensional or Rectangular Array
9. Accessing Array Elements
10. Initializing an Array

11. Automatic Initialization
12. Explicit Initialization of One-Dimensional Arrays
13. Explicit Initialization of Rectangular Arrays
14. Syntax Points for Initializing Rectangular Arrays
15. Jagged Arrays
16. Declaring a Jagged Array
17. Shortcut Instantiation
18. Instantiating a Jagged Array
19. Sub-Arrays in Jagged Arrays
20. Comparing Rectangular and Jagged Arrays
21. The for each Statement
22. The Iteration Variable Is Read-Only
23. The for each Statement with Multidimensional Arrays
24. Array Covariance
25. Useful Inherited Array Members
26. The Clone Method
27. Comparing Array Types

UNIT 15 Delegates

1. What Is a Delegate?
2. Declaring the Delegate Type
3. Creating the Delegate Object
4. Assigning Delegates
5. Combining Delegates
6. Adding Methods to Delegates
7. Deleting Methods from a Delegate
8. Invoking a Delegate
9. Delegate Example
10. Invoking Delegates with Return Values
11. Invoking Delegates with Reference Parameters
12. Anonymous Methods
13. Using Anonymous
14. Syntax of Anonymous Methods
15. Scope of Variables and Parameters

UNIT 16 Events

1. Events Are Like Delegates
2. An Event Has a Private Delegate
3. Overview of Source Code Components
4. Declaring an Event
5. An Event Is a Member
6. The Delegate Type and Event Handler
7. Raising an Event
8. Subscribing to an Event

9. Removing Event Handlers
10. Adding Anonymous Method Event Handlers
11. Standard Event Usage
12. Using the EventArgs Class
13. Passing Data by Extending EventArgs
14. Using the Custom Delegate
15. The MyTimerClass Code
16. Event Accessors

UNIT 17 Interfaces

1. What Is an Interface?
2. Example Using the IComparable Interface
3. Declaring an Interface
4. Implementing an Interface
5. Example with a Simple Interface
6. An Interface Is a Reference Type
7. Implementing Multiple Interfaces
8. Implementing Interfaces with Duplicate Members
9. References to Multiple Interfaces
10. An Inherited Member As an Implementation
11. Explicit Interface Member Implementations
12. Accessing Explicit Interface Member Implementations
13. Interfaces Can Inherit Interfaces
14. Using the as Operator with Interfaces
15. Example of Different Classes Implementing an Interface

UNIT 18 Conversions

1. What Are Conversions?
2. Implicit Conversions
3. Explicit Conversions and Casting
4. Casting
5. Types of Conversions
6. Numeric Conversions
7. Implicit Numeric Conversions
8. Overflow Checking Context
9. Explicit Numeric Conversions
10. Reference Conversions
11. Implicit Reference Conversions
12. Explicit Reference Conversions
13. Valid Explicit Reference Conversions
14. Boxing Conversions
15. Boxing Creates a Copy
16. Unboxing Conversions

17. The Unboxing Conversions
18. User-Defined Conversions
19. Constraints on User-Defined Conversions
20. Example of a User-Defined Conversion
21. Evaluating User-Defined Conversions
22. Example of a Multi-Step User-Defined Conversion
23. The is Operator
24. The as Operator

UNIT 19 Generics

1. What Are Generics?
2. A Stack Example
3. Generics in C#
4. Continuing with the Example
5. Generic Classes
6. Declaring a Generic Class
7. Creating a Constructed Type
8. Creating Variables and Instances
9. The Stack Example Using Generics
10. Comparing the Generic and Non-Generic Stack
11. Constraints on Type Parameters
12. Where Clauses
13. Constraint Types and Order
14. Generic Structs
15. Generic Interfaces
16. An Example Using Generic Interfaces
17. Generic Interface Implementations Must Be Unique
18. Generic Delegates
19. Generic Methods
20. Declaring a Generic Method
21. Invoking a Generic Method
22. Example of a Generic Method

UNIT 20 Enumerators and Iterators

1. Enumerators and Enumerable Types
2. Using the foreach Statement
3. Types of Enumerators
4. Using the IEnumerator Interface
5. Declaring an IEnumerator Enumerator
6. The IEnumerable Interface
7. Example Using IEnumerable and IEnumerator
8. The Non-Interface Enumerator
9. The Generic Enumeration Interfaces
10. The IEnumerator<T> Interface

11. The IEnumerable<T> Interface
12. Iterators
13. Iterator Blocks
14. More about Iterators
15. Producing Enumerables and Enumerators
16. Using an Iterator to Produce an Enumerable
17. Using an Iterator to Produce an Enumerator

UNIT 21 Attributes

1. What Is an Attribute?
2. A Quick Preview
3. The Stages of an Attribute
4. Applying an Attribute
5. Multiple Attributes
6. Other Targets
7. Global Attributes
8. Predefined Attributes
9. Using Custom Attributes
10. Declaring a Custom Attribute
11. Using Attribute Constructors
12. Specifying the Constructor
13. Using the Constructor
14. Positional and Named Parameters in Constructors
15. Restricting the Usage of an Attribute
16. Using Suggested Practices for Custom Attributes
17. Accessing an Attribute
18. Using the IsDefined Method
19. Using the GetCustomAttributes Method
20. Using Reserved Attributes
21. The Obsolete Attribute
22. The Conditional Attribute

UNIT 22 Preprocessor Directives

1. What Are Preprocessor Directives?
2. General Rules
3. The #define and #undef Directives
4. Conditional Compilation
5. The Conditional Compilation Constructs
6. Diagnostic Directives
7. Line Number Directives
8. Region Directives

UNIT 23 Other Topics

1. Miscellaneous Topics
2. Strings
3. Using Class StringBuilder
4. Formatting Numeric Strings
5. Parsing Strings to Data Values
6. Nullable Types
7. Creating a Nullable Type
8. Using Nullable User-Defined Types
9. Method Main
10. Documentation Comments
11. Inserting Documentation Comments
12. Nested Types
13. Example of a Nested Class
14. Visibility and Nested Types

UNIT 24 WinForms

1. Introduction
2. Controls
3. Control events
4. Menus and Conext Menus
5. MenuStrip, ToolbarStrip
6. SDI and MDI Applications
7. Dialogbox Modal and Modeless
8. Form Inheritance
9. Developing Custom, Composite controls
10. Other Misc topics
11. Working with Resource Files
12. Working with Settings

UNIT 25: Q/A Session and If anything student wants to know will be discussed here