

VB.NET

Day wise Break up

DAY	MODULE	TOPICS
Day 1	Module 1	Getting Started
	Module 2	Basic Language Constructs
	Module 3	Classes and Objects
Day 2	Module 4	Inheritance
	Module 5	Interfaces
	Module 6	Exception Handling
	Module 7	Some Useful In-Built Classes
Day 3	Module 8	Delegates & Events
	Module 9	Generics & Nullable Types
	Module 10	Win Forms
Day 4	Module 11	More Win Forms Controls
	Module 12	Deployment
	Module 13	Working with Databases
Day 5	Module 14	Data-Bound Controls
	Module 15	New Features in ADO.NET
	Module 16	.NET Framework and Assembly
Day 6	Module 17	Shared Assemblies and Versioning
	Module 18	Reflection
Day 7	Module 19	Attributes
	Module 20	Threads
Day 8	Module 21	The I/O Classes
	Module 22	Introduction to Web Services

Course Content

Module 1 : Getting Started

- Introduction to VB.NET
- Understanding Visual Studio.NET projects
- Understanding the .NET IDE
- Writing the first VB.NET program and running it

Module 2 : Basic Language Constructs

- Data types and variables
- Conditional Statements
- Looping constructs
- Continue statement
- Functions and subroutines
- Working with arrays

Module 3 : Classes and Objects

- Understanding the object oriented concepts
- Defining classes
- Attributes, properties and methods
- Creating objects and methods
- Access control
- Constructors
- Method overloading
- Parameter values
- Shared fields and methods
- The 'Main' method
- Garbage collection
- The Object class

Module 4 : Inheritance

- Understanding class inheritance
- Constructors in derived classes
- Method overriding
- Polymorphism
- Hiding base class fields in derived classes
- Making methods and classes 'NotOverridable/NotInheritable'
- Must Inherit classes and MustOverride methods

Module 5 : Interfaces

- What is an interface?
- Creating, implementing and using interfaces
- Interfaces and properties
- Interfaces versus abstract classes

Module 6 : Exception Handling

- The 'Try-Catch-Finally' block
- Throw keyword
- The in-built Exception Classes
- Creating exception classes

Module 7 : Some Useful In-Built Classes

- The String class
- Common Collection classes : Array list, Hash table, Stack, Queue, Sorted List
- Operator Overloading

Module 8 : Delegates and Events

- Understanding the concept of delegates
- Declaring, instantiating and invoking delegates
- Creating delegate methods
- SingleCast and Multicast delegates
- Delegates : safe function pointers
- Loose coupling using delegates: Strategy Pattern
- Events

Module 9 : Generics & Nullable types

- Understanding Generics
- Type-safe collection classes
- Nullable types

Module 10 : Win Forms

- Introduction to Win Forms
- Windows forms library
- Creating simple GUI by hand
- Creating forms with controls
- Responding to form events
- Event handling
- Using basic controls (e.g., Label, Textbox, Button, etc.)
- Anchor and Dock properties
- MDI application

Module 11 : More Win Forms Controls

- MenuStrip and ContextMenuStrip controls
- Dialog controls
- ToolStrip and Status Strip controls
- Container Controls
- Web Browser Control
- Date time Picker control
- Common Dialog Boxes
- Adding controls at run time

Module 12 : Deployment

- Deploying your application to the end user
- CAB setup
- Microsoft Inraller (MIS) setup

Module 13 : Working with Databases

- Introduction to ADO.NET
- ADO.NET Architecture
- Using Connection and command objects
- Using Dataset, DataReader, Data view objects
- The ADO.NET architecture and its components

Module 14 : Data Bound Controls

- Data Grid view control
- Binding data to textbox controls
- Using Binding Navigator, List controls
- Master- detail transactions
- Stored Procedures

Module 15 : New Features in ADO.NET

- Connected and disconnected environment
- ADO.NET object model
- Data sources, providers and connections
- Commands and data reports
- Data sets and data adaptors
- Data tables, rows and columns
- Constrains and relations

Module 16 : .Net Framework and Assembly

- The .NET framework
- What are assemblies?
- Modules and using modules
- Assemblies versus regular.DDLs
- Working with assemblies
- Viewing the Manifest file
- Assemblies for reuse,versioning, deployment and security
- Assembly contents : Metadata, Resources, code and manifest
- Private and shared assemblies
- Creating and using assembly DLLs
- Assemblies and using hard names

Module 17: Shared Assemblies and versioning

- Shared Assemblies
- Global Assembly data
- Placing assemblies within GAC
- Assigning strong name to assemblies
- Side by side versions and cultures
- Strong Names

Module 18 : Reflection

- What is reflection?
- Metadata, data about data
- Reflection API: Assembly, Module type and member info classes

Module 19 : Attributes

- What are attributes?
- Some intrinsic (build-in) attributes
- Creating custom attributes
- Retrieving attributes value

Module 20 : Threads

- Understanding threads
- Creating, Executing and terminating threads
- Thread states and priorities
- Synchronization among threads

Module 21 : The I/O Classes

- The Stream classes
- Reading and writing to files
- The Directory and File classes

Module 22 : Introduction to Web Services

- Overview of Web Services
- Creating Web Services
- Calling web services from Win forms and Web forms

ASP.NET

Day wise Break up

DAY	MODULE	TOPICS
Day 1	Module 1	Introduction to ASP.NET, ASP.NET 2005 & C#
	Module 2	Web Form Controls
Day 2	Module 3	ADO.NET architecture
	Module 4	Accessing data in the connected environment
Day 3	Module 5	Disconnected data
	Module 6	Understanding the XML Web Services
Day 4	Module 7	Web Applications and XML
	Module 8	Data Source Control
Day 5	Module 9	Data Bound Control
	Module 10	Navigation Control
Day 6	Module 11	Themes and Master Pages
	Module 12	State Management
Day 7	Module 13	Caching
	Module 14	Security
	Module 15	Login Controls and Membership API

Course Contents

Module 1 : Introduction to Asp.net 2005 and C#

- What is Web Application
- What is Web Server – IIS
- Visual Studio IDE 2005
- Response/Request Model
- Understanding C# as Object oriented language.

Module 2 : Web Form Controls

- Using HTML server controls and code behind files
- Web pages and Browsers
- Static and dynamic web pages
- ASP.NET Server controls
- IsPostBack Control
- PostBack Control
- Auto Post Back Property
- Validation Controls
- New Controls
- Session state
- Data binding

Module 3 : ADO.NET Architecture

- Use of Ado.Net
- Disconnected Model
- Connected Model
- Example on connected and disconnected model

Module 4 : Accessing data in the connected environment

- Commands
- Creating and executing commands
- Reading data using a data reader
- Single result queries
- Parameterised queries (input & output parameters)
- Adding, editing and deleting data

Module 5 : Disconnected Data

- Why using a disconnected environment?
- Dataset and Data adapter features
- Filling sets using data adaptors
- Read data using data tables, rows and columns

Module 6 : Understanding the XML Web Services

- Markup Languages
- What is XML?
- Creating XML Documents
- Validating XML Data using DTD
- Validation XML Data using XML Schema
- Structure of XML Schema
- Element Declarations
- Creating and reading DOM trees
- Working with Elements and Attributes

Module 7 : Web Applications and XML

- What is a Web application, really?
- Web services and XML
- Simple object access protocol(SOAP)
- Web service Description Language(WSDL)
- Universal Discovery, Description and integration(UDDI)
- Asynchronous web services

Module 8 : Data Source Control

- SQL Data source Control
- Object Data Source Control
- Example – Retrieve, Insert, Update, Delete data using Data Source

Module 9 : Data Bound Control

- Grid View
- Detail view
- Data List
- Paging
- Sorting
- Bound Field
- Command Button Field
- Template Field

Module 10 : Navigation Control

- XML Data Source
- Site Map Data Source
- Tree view
- Menu
- Site Map Control

Module 11 : Themes and Master pages

- Creating themes in skin files
- Apply themes for the pages
- Creating Master page
- Create Content page

Module 12 : State management

- Client side state management techniques
- View State
- Query String
- Cookie
- Server side state management techniques
- Application Variable
- Session Variable

Module 13 : Caching

- What is Caching
- Page Caching
- Data Caching
- Parameter Caching
- Expiration policy Sliding and Absolute expiration
- File Dependency
- SQL Data Cache invalidation
- Fragment Caching

Module 14 : Security

- What is Security
- Authentication
- Authorization
- Types of Authentication Windows, Forms, Passport
- Example using Windows security mode
- Example using Forms Security Mode

Module 15 : Login controls and Membership API

- Creating security form using Membership API function
- User/Role Based security
- Login Controls
- Login, Create user wizard, Login View, Login Name, Login status
- Change Password, Password Recovery

C# .NET

Day wise Break up

DAY	MODULE	TOPICS
Day 1	Module 1	Overview of the Microsoft .NET Platform
	Module 2	Overview of C# .NET
Day 2	Module 3	Using Value-Type Variables
	Module 4	Statements and Exceptions
Day 3	Module 5	Methods and Parameters
	Module 6	Arrays
Day 4	Module 7	Essentials of Object-Oriented Programming
	Module 8	Using Reference-Type Variables
Day 5	Module 9	Creating and Destroying Objects
	Module 10	Inheritance in C#
Day 6	Module 11	Aggregation, Namespaces, and Advanced Scope
	Module 12	Operators and Events
Day 7	Module 13	Properties and Indexers
	Module 14	Attributes

Course Contents

Module 1: Overview of the Microsoft .NET Platform

The following topics are covered in this module:

- Introduction to the .NET Platform
- Overview of the .NET Framework
- Benefits of the .NET Framework
- The .NET Framework Components
- Languages in the .NET Framework

After completing this module, you will be able to list the major elements of the .NET Framework and explain how the C# language fits into the .NET Platform. This includes:

- Describing the .NET Platform.
- Listing the main elements of the .NET Platform.
- Explaining the language support in the .NET Framework.
- Describing the .NET Framework and its components.

Module 2: Overview of C# .NET

The following topics are covered in this module:

- Structure of a C# Program
- Basic Input/Output Operations
- Recommended Practices
- Compiling, Running, and Debugging

After completing this module, you will be able to analyze the basic structure of a C# application and be able to document, debug, compile, and run a simple application. This includes:

- Explaining the structure of a simple C# program.
- Using the Console class of the System namespace to perform basic input/output operations.
- Handling exceptions in a C# program.
- Generating Extensible Markup Language (XML) documentation for a C# application.
- Compiling and executing a C# program.
- Using the Microsoft Visual Studio Debugger to trace program execution.

Module 3: Using Value-Type Variables

The following topics are covered in this module:

- Common Type System
- Naming Variables
- Using Built-In Data Types
- Creating User-Defined Data Types
- Converting Data Types

After completing this module, you will be able to create, name, and assign values to variables. This includes:

- Describing the types of variables that you can use in C# applications.
- Naming your variables according to standard C# naming conventions.
- Declaring variables by using built-in data types.
- Assigning values to variables.
- Converting existing variables from one data type to another.
- Creating and using your own data types

Module 4: Statements and Exceptions

The following topics are covered in this module:

- Introduction to Statements
- Using Selection Statements
- Using Iteration Statements
- Using Jump Statements
- Handling Basic Exceptions
- Raising Exceptions

After completing this module, you will be able to use common statements to implement flow control, looping, and exception handling. This includes:

- Describing the different types of control statements.
- Using jump statements.
- Using selection statements.
- Using iteration statements.
- Handling and raising exceptions.

Module 5: Methods and Parameters

The following topics are covered in this module:

- Using Methods
- Using Parameters
- Using Overloaded Methods

After completing this module, you will be able to create methods (functions and subroutines) that can return values and take parameters. This includes:

- Creating static methods that accept parameters and return values.
- Passing parameters to methods in different ways.
- Declaring and using overloaded methods.

Module 6: Arrays

The following topics are covered in this module:

- Overview of Arrays
- Creating Arrays
- Using Arrays

After completing this module, you will be able to create, initialize, and use arrays. This includes:

- Creating, initializing, and using arrays of varying rank.
- Using command-line arguments in a C# program.
- Describing the relationship between an array variable and an array instance.
- Using arrays as parameters for methods.
- Returning arrays from methods.

Module 7: Essentials of Object-Oriented Programming

The following topics are covered in this module:

- Classes and Objects
- Using Encapsulation
- C# and Object Orientation
- Defining Object-Oriented Systems

After completing this module, you will be able to explain the basic concepts and terminology of object-oriented programming. This includes:

- Defining the terms object and class in the context of object-oriented programming.
- Describing the three core aspects of an object: identity, state, and behavior.
- Describing abstraction and how it helps you to create reusable classes that are easy to maintain.
- Using encapsulation to combine methods and data in a single class and enforce abstraction.
- Explaining the concepts of inheritance and polymorphism.
- Creating and using classes in C#.

Module 8: Using Reference-Type Variables

The following topics are covered in this module:

- Using Reference-Type Variables
- Using Common Reference Types
- The Object Hierarchy
- Namespaces in the .NET Framework
- Data Conversions

After completing this module, you will be able to use common objects and reference types. This includes:

- Describing the key differences between reference types and value types.
- Using common reference types such as string.
- Explaining how the object type works and becoming familiar with the methods it supplies.
- Describing common namespaces in the .NET Framework.
- Determining whether different types and objects are compatible.
- Explicitly and implicitly converting data types between reference types.
- Performing boxing and unboxing conversions between reference and value data.

Module 9: Creating and Destroying Objects

The following topics are covered in this module:

- Using Constructors
- Initializing Data
- Objects and Memory
- Resource Managements

After completing this module, you will be able to create, initialize, and destroy objects in a C# application. This includes:

- Using constructors to initialize objects.
- Creating overloaded constructors that can accept varying parameters.
- Describing the lifetime of an object and what happens when it is destroyed.
- Creating destructors and using Finalize methods.

Module 10: Inheritance in C#

The following topics are covered in this module:

- Deriving Classes
- Implementing Methods
- Using Sealed Classes
- Using Interfaces
- Using Abstract Classes

After completing this module, you will be able to build new C# classes from existing classes. This includes:

- Deriving a new class from a base class and calling members and constructors of the base class from the derived class.
- Declaring methods as virtual and override or hiding them as required.
- Sealing a class so that it cannot be derived from.
- Implementing interfaces by using both the implicit and explicit methods.
- Describing the use of abstract classes and their implementation of interfaces

Module 11: Aggregation, Namespaces, and Advanced Scope

The following topics are covered in this module:

- Using Internal Classes, Methods, and Data
- Using Aggregation
- Using Namespaces
- Using Modules and Assemblies

After completing this module, you will be able to create self-contained classes and frameworks in a C# application. This includes:

- Using internal access to allow classes to have privileged access to each other.
- Using aggregation to implement powerful patterns such as Factories.
- Using namespaces to organize classes.
- Creating simple modules and assemblies.

Module 12: Operators and Events

The following topics are covered in this module:

- Introduction to Operators
- Operator Overloading
- Creating and Using Delegates
- Defining and Using Events

After completing this module, you will be able to define operators, use delegates, and add event specifications. This includes:

- Defining operators to make a class or struct easier to use.
- Using delegates to decouple a method call from a method implementation.
- Adding event specifications to a class to allow subscribing classes to be notified of changes in object state.

Module 13: Properties and Indexers

The following topics are covered in this module:

- Using Properties
- Using Indexers

After completing this module, you will be able to implement properties and indexers. This includes:

- Creating properties to encapsulate data within a class.
- Defining indexers to gain access to classes by using array-like notation.

Module 14: Attributes

The following topics are covered in this module:

- Overview of Attributes
- Defining Custom Attributes
- Retrieving Attribute Values

After completing this module, you will be able to use predefined and custom attributes. This includes:

- Using common predefined attributes.
- Creating simple custom attributes.
- Querying attribute information at run time.