

Teaching Guidelines for  
**MS.Net Technologies**  
PG-DAC September 2022

---

**Duration:** 84 hours (42 theory hours + 34 lab hours + 8 revision/practice hours)

**Objective:** To acquire the knowledge of Microsoft.NET 6.

**Prerequisites:** Students are expected to know any OOP. They should have undergone the Web Programming module which includes HTML, CSS, JavaScript, JSON, and XML. Knowledge of any database is required.

**Note:** *Training will be carried out on .Net 6 using Visual Studio 2022*

**Evaluation:** 100 marks

**Weightage:** Theory exam – 40%, Lab exam – 40%, Internals – 20%

**Text Book:**

- Pro C# 10 with .Net 6 - Foundational Principles and Practices in Programming by Andrew Troelsen & Philip Japikse / Apress

**References:**

- C# 10 and .Net 6 - Modern Cross-Platform Development by Mark J. Price / Packt
- 

(Note: Each Session is of 2 hours)

**Session 1:**

**Lecture:**

Introduction to the .Net Framework

Intermediate Language (IL)

Assemblies and their structure, EXEs/DLLs

CLR and its functions

- JIT Compilation
- Memory Management
- Garbage Collection
- AppDomain Management
- Memory Management
- CLS, CTS
- Security

**NO LAB**

**Session 2:**

**Lecture:**

.Net Framework, .Net Core, Mono, Xamarin differences

Versions of the Framework

Managed and Unmanaged Code

Introduction to Visual Studio

Using ILDASM

**NO LAB**

### **Session 3:**

#### **Lecture:**

Console Applications and Class Libraries .Net Core

C# Basics

Project References, using

Classes

Data Types in .net and CTS equivalents

Methods

- Method Overloading
- Optional Parameters
- Named Parameters and Positional Parameters
- Using params
- Local functions

Properties

- get, set
- Readonly properties
- Using property accessors to create Readonly property

Constructors

Object Initializer

Destructors

Discussion on IDisposable. To be implemented after interfaces

#### **Lab:**

Create a class that has Properties, Fields, Methods, Constructors (Trainer can specify any class of his choice, e.g. Student, Employee, etc)

### **Session 4:**

#### **Lecture:**

Static Members of a Class

- Fields
- Methods
- Properties
- Constructors

Static Classes

Static local functions

Inheritance

- Access Specifiers
- Constructors in a hierarchy
- Overloading in derived class
- Hiding, using new
- override
- sealed methods
- Abstract Classes
- Abstract Methods
- Sealed Classes

#### **Lab:**

Create multiple classes that use Inheritance based concepts

### **Session 5:**

#### **Lecture:**

Interfaces

- Implementing an interface
- Explicitly implementing an interface
- Inheritance in interfaces
- Default interface methods

Operator overloading

**Lab:**

Create and implement interfaces for the classes created in Lab 4

Implement IDisposable, IComparable

**Session 6:**

**Lecture:**

Reference and Value Types

Value Types

- struct
- enum

out and ref

nullable types

nullable reference types

?? and ??=

Working with Arrays (single, multidim, jagged), Array Class members

Indices and ranges

Indexers

**Lab:**

Lab based on array examples.

Also create an array of the class created in Lab 1.

**Session 7:**

**Lecture:**

Generic classes

Generic methods

Generic Constraints

Collections – generic and non-generic

Collection Examples based on ICollection, IList, IDictionary (both generic and non-generic)

Iterating collections using foreach

Using Tuples to pass multiple values to a function

**Lab:**

Lab based on collection examples.

Also create a collection of the class created in Lab 1.

**Session 8:**

**Lecture:**

Delegates

- Calling methods using delegates
- Uses of delegates
- Multicast delegates
- Action, Func, Predicate delegates

Anonymous methods

Lambdas

**Lab:**

Lab based on delegates examples.

**Session 9:****Lecture:**

Error Handling (Exceptions Handling)

- Checked & Unchecked Statements
- The try, catch, finally
- Dos & Don'ts of Exception Handling

User Defined Exception classes

Declaring and raising events

Handling events

**Lab:**

Lab based on exceptions and events examples.

**Session 10:****Lecture:**

Anonymous types

Extension methods

Partial classes

Partial methods

LINQ to objects

Writing LINQ queries

Deferred execution

LINQ methods

PLINQ

**Lab:**

Lab based on LINQ examples

Students to try tutorial for 101 LINQ Queries

**Session 11:****Lecture:**

Creating a shared assembly

Creating Custom Attributes

Using Reflection to explore an Assembly

Using Reflection to load an Assembly dynamically

Files I/O and Streams

- Working with drivers, Directories, and Files
- Reading and Writing files

**NO LAB****Session 12:****Lecture:**

Threading

- ThreadStart, Parameterized ThreadStart
- ThreadPool
- Synchronizing critical data using lock, Monitor and Interlocked

Working with Tasks

- Calling functions with and without return values
- Using async, await

Using the Task Parallel Library

**Lab:**

Threading related examples

Task related examples

## **Sessions 13-19:**

### **Lecture:**

#### **Introduction to Asp.Net MVC CORE**

- Architecture of an ASP .Net MVC application
- Understanding Folder structures and configuration files

#### **Understanding Controllers and Action**

- Create a controller
- How actions are invoked
- HttpGet , HttpPost , NoAction Attributes
- Running Action result.

#### **Understanding Views & Models**

- Creating Models & ViewModel
- Creating Razor Views
- HTML Helper Functions
- Understanding ViewBag
- Create a view using ViewBag
- Validation using Data Annotations
- Client side and server side validation
- Self validated model
- Creating Strongly Types Views
- Using Various Scaffold Templates
- CRUD operation using Model

#### **MVC State Management**

- ViewBag , TempData , Session , Application
- Cookies , QueryString

#### **MVC Module**

- Partial View
- Action Method and child action

#### **Data Management with ADO.NET**

- Microsoft.Data.SqlClient introduction
- Connection object, Command object, DataReader, DataAdapter, DataSet and DataTable.
- Asynchronous command Execution
- Asynchronous Connections

#### **Understanding Routing & Request Life Cycle**

- Routing Engine & Routing Table
- Understanding and configuring Routing Pattern in RouteConfig File
- Understanding 404 error and resource not found.
- Using Attributes Routing
- Understanding Request Life Cycle

#### **Layouts , Bundle , Minification**

- Creating Layout and using with associated views
- Understanding Bundling and Minification
- Using BundleConfig file
- Attaching css , js , bootstrap in bundles
- Custom Helper Function
- Asynchronous Actions
- Error Handling in MVC with Log Entry
- Filters and Custom Action Filter

#### **MVC Security**

- Using Authorize & Allow Anonymous attributes

- Implementing Forms Based Authentication
- Preventing Forgery Attack using AntiForgeryToken
- Preventing Cross Site Scripting Attack

#### **Entity Framework**

- Introduction to EF
- Different Approaches
- Using Code First Approach
- Using various Data Annotations
- Using Validation, Primary Key , Foreign Key etc
- Using Fluent APIs
- Database Migrations
- CRUD operation using EF

#### **Developing MVC application using EF Code First Approach**

##### **Introduction to Razor Pages**

##### **Lab:**

Lab exercise covering the concepts covered in the class

#### **Session 20:**

##### **Lecture:**

Localization in MVC (Demo Only)

Deploying ASP .NET MVC application (Demo only)

**NO LAB**

#### **Session 21**

##### **Lecture:**

##### **Web APIs**

- Creating ASP.NET MVC Web API
- Configuring for CORS
- Different Verbs
- Consuming using a client
- Using Newtonsoft APIs

##### **Lab:**

Create a RESTful service using WEB API. Create a consumer.