Contents

[OOP 3](#_Toc1901991866)

[Classes/Objects 3](#_Toc1643284359)

[Class Attributes 3](#_Toc1275800753)

[Class Methods 3](#_Toc1000472176)

[Constructors 3](#_Toc877382640)

[Modifiers 4](#_Toc1409606078)

[Encapsulation – Pillar 1 4](#_Toc1259428835)

[Inheritance – Pillar 2 4](#_Toc2094944117)

[Polymorphism – Pillar 3 4](#_Toc390133947)

[Abstraction – Pillar 4 5](#_Toc928833273)

[Packages 5](#_Toc110588919)

[API Application Programming Interface 5](#_Toc1104589464)

[Interface 5](#_Toc532332469)

[Enums 5](#_Toc1892074510)

[Exceptions 5](#_Toc421333477)

[this keyword 5](#_Toc1062020003)

[RESTful Application Programming Interface (API) 6](#_Toc1771416055)

[Design Patterns 6](#_Toc803686094)

[What is C#? 6](#_Toc733728328)

[What is .NET? 6](#_Toc1786757804)

[What is CLR (Common Language Run-time)? 7](#_Toc1189566208)

[Architecture of .NET Applications 7](#_Toc125704414)

[Variables and constants 7](#_Toc425167585)

[SQL Function sample 7](#_Toc2066183569)

[SQL Store Procedure sample 8](#_Toc1494997784)

[ISeries Run SQL Script sample 8](#_Toc1507773284)

[Recursion 8](#_Toc1776635932)

[MVC – Model View Controller 8](#_Toc1632320083)

[Object Relational Mapper (ORM) 9](#_Toc55281896)

[Entity Framework 9](#_Toc1213762575)

[Tutorial Links 9](#_Toc1754321126)

# OOP

* Object-Oriented Programming
* Based on the concept of objects
* Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

# Classes/Objects

* Class is the blueprint for the creation of an object
* Container
  + Data attributes
    - State of Application
  + Methods (functions)
    - Behavior execute code

# Class Attributes

* Class attributes are variables within a class

# Class Methods

* A block of code which only runs when it is called
* Defined only within a class
* Also known as functions
* Method names end with a parenthesis

# Constructors

* A method that is called when an instance of a class is created (new keyword)
* Used to initialize objects
* It is used to set initial values for object attributes
* Constructor has the exact name of the class, can have different parameters
* Does not have any return type in the definition of the constructor
* All classes have constructors by default
  + Created for you if you do not create one
  + If you use the default you are not able to set initial values for object values

# Modifiers

* Classes
  + Public – class is accessible by any other class
  + Default – class is only accessible by classes in the same package
* Attributes (variables), methods and constructors
  + Public – class is accessible by any other class
  + Private – code is only accessible within the declared class
  + Default – class is only accessible by classes in the same package
  + Protected – Code is accessible in the same package and subclassess

# Encapsulation – Pillar 1

* [Learn JAVA : Encapsulation in Java - #2 Java Tutorial - YouTube](https://www.youtube.com/watch?v=pWN1JKKWDaQ)
* Is the process of wrapping function and data members together in a class
* A class encapsulates the fields, which holds the state of an object and the methods which define the actions of the object
* The data is not accessible to the outside world, only the functions wrapped inside the class can access it.
* Is to prevent unauthorized and unwanted changes of data from outside of the function
* How do you implement encapsulation?
  + Making your variables private
  + Defining one pair of public setter and getter methods or properties to access private variable
* We group related variables and functions(methods) together
* Increase reusability
* Reduce complexity

# Inheritance – Pillar 2

* Inherit attributes and methods from one class to another
* Inheritance concept is broken into two categories
  + Subclass (child) – the class that inherits from another class
  + Superclass (parent) – the class being inherited from
* Use the Extends keyword
* Reduces redundant code

# Polymorphism – Pillar 3

* Means “many forms”
* A method that can accept multiple parameters with different signatures
* Method overloading
* Occurs when we have many classes that are related to each other by inheritance
* Helps eliminate If and Else or Switch and Case statements

# Abstraction – Pillar 4

* Is the process of hiding certain details and showing only essential information to the user
* You can achieve abstraction with either abstract classes or interfaces
* Is a restricted class that cannot be used to create objects – to access it, it must be inherited from another class
* Abstract method can be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from)
* Reduce complexity
* Isolate impact of changes

# Packages

* A group of related classes
* It is like a folder in a file directory
* Used to avoid name conflicts

# API Application Programming Interface

* API is a set of rules that define how applications or devices can connect to and communicate with each other

# Interface

* Is a completely “abstract” class that is used to group related methods with empty bodies
* To access the interface methods, the interface must be “implemented by another class with the implements keyword

# Enums

* Is a special class that represents a group of constants
* Create an enum, use the enum keyword
* Separate the constants with a comma
* Constants should be in uppercase

# Exceptions

# this keyword

* Represents the current object

# RESTful Application Programming Interface (API)

* Architectural style
* API is a set of rules that define how applications or devices can connect to and communicate with each other
* REST is an API that conforms to the design principles of the REST, or representational state transfer architectural style
  + Uniform interface
  + Client-server decoupling
  + Statelessness
  + Cache ability
  + Layered system architecture
  + Code on demand
* Communicates via HTTP requests
  + GET request – retrieve a record
  + POST request – create a record
  + DELETE request – delete a record
  + PUT request – update a record
* [What is a REST API? | IBM](https://www.ibm.com/cloud/learn/rest-apis)

# Design Patterns

* Singleton
  + design pattern that restricts the instantiation of a class to one object
  + A singleton is a class designed to only ever have one instance.
  + Class itself is responsible for enforcing this design requirement
  + Usually used for share resources
    - File system
    - Network resource
    - Scanner
    - Print spooler
    - Expensive one-time configuration
* Simpler Interface
* Reduce the Impact of Change
* Hide the details and the complexity and just show the essentials

# What is C#?

* C# is an Object Oriented Programming (OOP) language

# What is .NET?

* .NET is a framework for building applications on Windows
* It supports multiple languages f#, vb.net, c#, etc…
* Comprised of two components
* CLR (Common Language Runtime)
* Class Library

# What is CLR (Common Language Run-time)?

* CLR exists because different hardware or different operating systems could be using your application
* CLR is an application that is in memory, if a machine has CLR installed your application will be able to run
* When a program is compiled, it is not compiled into machine language, it is translated into an intermediate language (IL) called byte code which is independent of the computer on which it’s running
* CLR transform the IL code into the native code or the machine that is running the application
* CLR is an application that is in memory to translate intermediate language (IL) code into the machine code and this process is called just in time (JIT) compilation

# Architecture of .NET Applications

* Classes
* Container of attributes and methods
* Namespaces
* container for related classes
* Assemblies
* container for related namespaces
* it’s a file on the disk which can be either
* Executable
* DLL – dynamically linked library

# Variables and constants

* Variable is a name given to a storage location in memory
* Constant is an immutable – a value that cannot be changes during the run of an application
* Is the process of wrapping function and data members together in a class
* Is to prevent unauthorized and unwanted change of data from outside of the function
* How do you implement encapsulation?
* Making your variables private
* Defining one pair of public setter and getter methods or properties to access private variables

# SQL Function sample

# SQL Store Procedure sample

# ISeries Run SQL Script sample

# Recursion

* Recursion is the technique of making a function call itself

# MVC – Model View Controller

* Architectural pattern
* MVC is a design pattern helps to enforce separation of concerns to help you avoid mixing presentation logic, business logic, and data access logic together.
* Model = classes (objects)
* Manages the behavior and data
* Data related
* Consists of classes / objects with properties
* Uses SQL statements
* Supplies the controller with lists of objects
* View = web page (Razor HTML)
* Manages the display of data
* HTML CSS Code (or similar)
* Usually gets a list of data from the controller
* Dynamically combines the data with HTML in a template
* Razor (ASP.NET)
* Controller = connects models, business logic and web pages.
* Public method in a controller is callable as an HTTP endpoint
* Handles and responds to user input and interaction. Handle browser requests
* Retrieve model data
* Call view templates that return a response
* Handles page events and navigation between pages

# Object Relational Mapper (ORM)

* Use to map our data in a relational database into objects of our applications.
* Maps between a class (that has fields/columns defined) and a table
* Developer does not create the sequel statements to create tables, the migration tool will generate the tables from the class
* Entity Framework is the name of the ORM
* 30:00

# Entity Framework

* Entity Framework does the following.
  + Opening a connection to the database
  + Reads the data
  + Maps it to objects and adds them back to the DB sets
  + Returns the DB context back to us
* Tool to access a database
* when we ask it to persist the changes, again it will automatically generate SQL statements and executes them on our database. Persistence means data survives after the process with which it was created has ended
* Before Entity Framework
* You had to take care of the following
* Connection to the database
* Executing a command
* Reading the data
* Closing the connection
* Use a class called DB context
  + - Can have one or more DB sets
    - DB sets represents tables in our database
    - Uses LINQ to query DB sets
    - Entity framework will translate our LINQ queries into SQL queries at runtime

# Tutorial Links

* [C# Tutorial For Beginners - Learn C# Basics in 1 Hour - YouTube](https://www.youtube.com/watch?v=gfkTfcpWqAY)
* [ASP.NET Core Crash Course - C# App in One Hour - YouTube](https://www.youtube.com/watch?v=BfEjDD8mWYg&t=515s)
* [Learn React In 30 Minutes - YouTube](https://www.youtube.com/watch?v=hQAHSlTtcmY&t=654s)
* [Get started with ASP.NET Core MVC | Microsoft Docs](https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-6.0&tabs=visual-studio)