**Assignment 1**

1. Write a program to create class Employee with following specifications:
   1. Field Members: firstName, lastName, salary

private string \_firstName;

private string \_lastName;

private double \_salary;

* 1. Properties : FirstName, LastName, FullName, Salary

public string FirstName

{

get => \_firstName;

set

{

\_firstName = value;

}

}

public string LastName

{

get => \_lastName;

set

{

\_lastName = value;

}

}

public double Salary

{

get => \_salary;

set

{

\_salary = value;

}

}

* 1. Parameterized Constructor

public Employee(string firstName, string lastName, double salary)

{

\_salary = salary;

\_lastName = lastName;

\_firstName = firstName;

}

* 1. Method: IncrementSalary(double s), Display() Method for showing details of employee

public void IncrementSalary(double s)

{

\_salary += s;

}

public void Display()

{

Console.WriteLine("Full Name is {0} {1}", \_firstName, \_lastName);

Console.WriteLine("Salary is {0}", \_salary);

}

1. Now create an object e1 of this Class & show use of:
   * 1. Constructor

Employee e1 = new Employee("Bijaya", "Shrestha", 10000);

* + 1. Display the Full Name and Salary using Properties

Console.WriteLine("Full Name is {0} {1}", e1.FirstName, e1.LastName);

Console.WriteLine("Salary is {0}", e1.Salary);

* + 1. Change the First Name

e1.FirstName = "Bijay";

Console.WriteLine("First Name is changed as {0}", e1.FirstName);

* + 1. Increment the Salary by 10%

double incSal = 0.1 \* e1.Salary;

e1.IncrementSalary(incSal);

* + 1. Display Full Name and Salary by calling method.

e1.Display();

1. What is polymorphism? Explain the different type of approaches to implement the polymorphism with example.

The term “Polymorphism” is the combination of “poly” + “morphs” which means many forms. It occurs when we have many classes that are related to each other by inheritance. Simply, it is the ability of different objects to respond in a unique way to the same message.

The different type of approaches to implement the polymorphism are:

* Static / Compile Time Polymorphism

It is also known as Early Binding. Method overloading is an example of Static Polymorphism. It is also known as Compile Time Polymorphism because the decision of which method is to be called is made at compile time.

In the following example, a class has two methods with the same name "Add" but with different input parameters (the first method has three parameters and the second method has two parameters).

static public int Add(int a, int b, int c) => a + b + c;

static public int Add(int a, int b ) => a + b;

static void Main(string[] args)

{

Console.WriteLine($"Add 3 items {Add(5, 9, 3)}");

Console.WriteLine($"Add 2 items {Add(5,7)}");

}

* Dynamic Runtime Polymorphism

Dynamic / runtime polymorphism is also known as late binding. Method overriding is an example of dynamic polymorphism. The compiler will decide which method to call at runtime and if no method is found then it throws an error.

public class Animal

{

public string color = "white";

public virtual void eat()

{

Console.WriteLine("\tEating...");

}

}

public class Dog : Animal

{

public override void eat()

{

Console.WriteLine("Dog is Eating...");

}

}

static void Main(string[] args)

{

Animal a = new Dog();

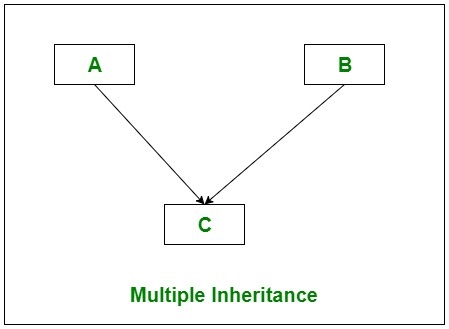
a.eat();

Console.WriteLine($"Color is {a.color}");

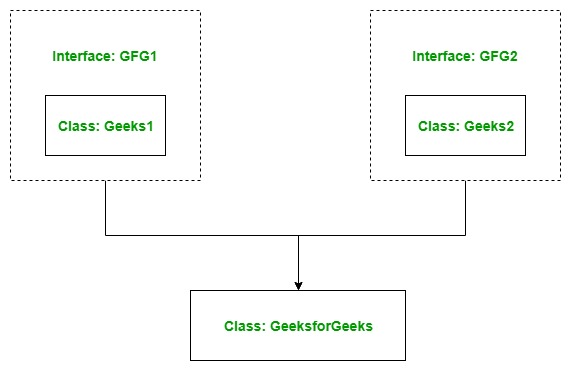
}

1. How do you create and implement interfaces to achieve multiple inheritance in C#? Explain with suitable example.

In Multiple inheritance, one class can have more than one superclass and inherit features from all its parent classes. As shown in the below diagram, class C inherits the features of class A and B.



But C# does not support multiple class inheritance. To overcome this problem, we can create and implement interfaces to achieve multiple class inheritance. With the help of the interface, class C (as shown in the above diagram) can get the features of class A and B.



Code from github….

1. What is collection? Explain with example.

Ans: In C#, collection is specialized classes for data storage and retrieval. It represents group of objects. Collection classes serve various purpose, such as allocating memory dynamically to elements and accessing a list of items on the basis of an index etc. We can perform various operations on objects such as

* Store object
* Update object
* Delete object
* Retrieve object
* Search object and
* Sort object

The namespaces for using collection with their classes in C# are:

* System.Collection
  + ArrayList
  + Stack
  + Queue
  + Hashtable
  + SortedList
* System.Collection.Generic
  + List
  + Dictionary

Example: Use of List



1. What is LINQ? Explain LINQ method syntax and give an example?

Language-Integrated Query (LINQ) is the name for a set of technologies based on the integration of query capabilities directly into the C# language

is uniform query syntax in C# and VB.NET to retrieve data from different sources and formats such as collections, ADO.Net DataSet, XML Docs, web service and MS SQL Server and other databases.



For an example:

// Data source

string[] names = {"Bill", "Steve", "James", "Mohan" };

// LINQ Query

var myLinqQuery = from name in names

where name.Contains('a')

select name;

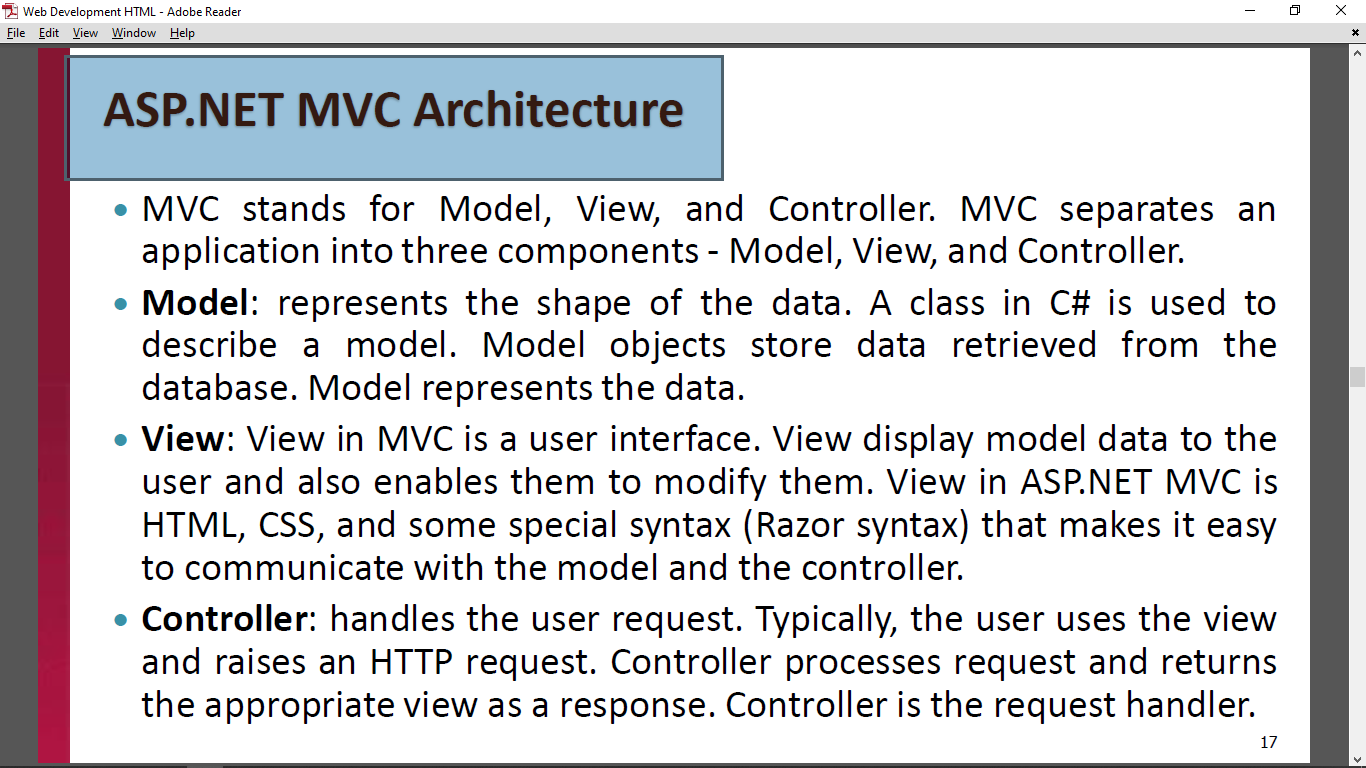
// Query execution

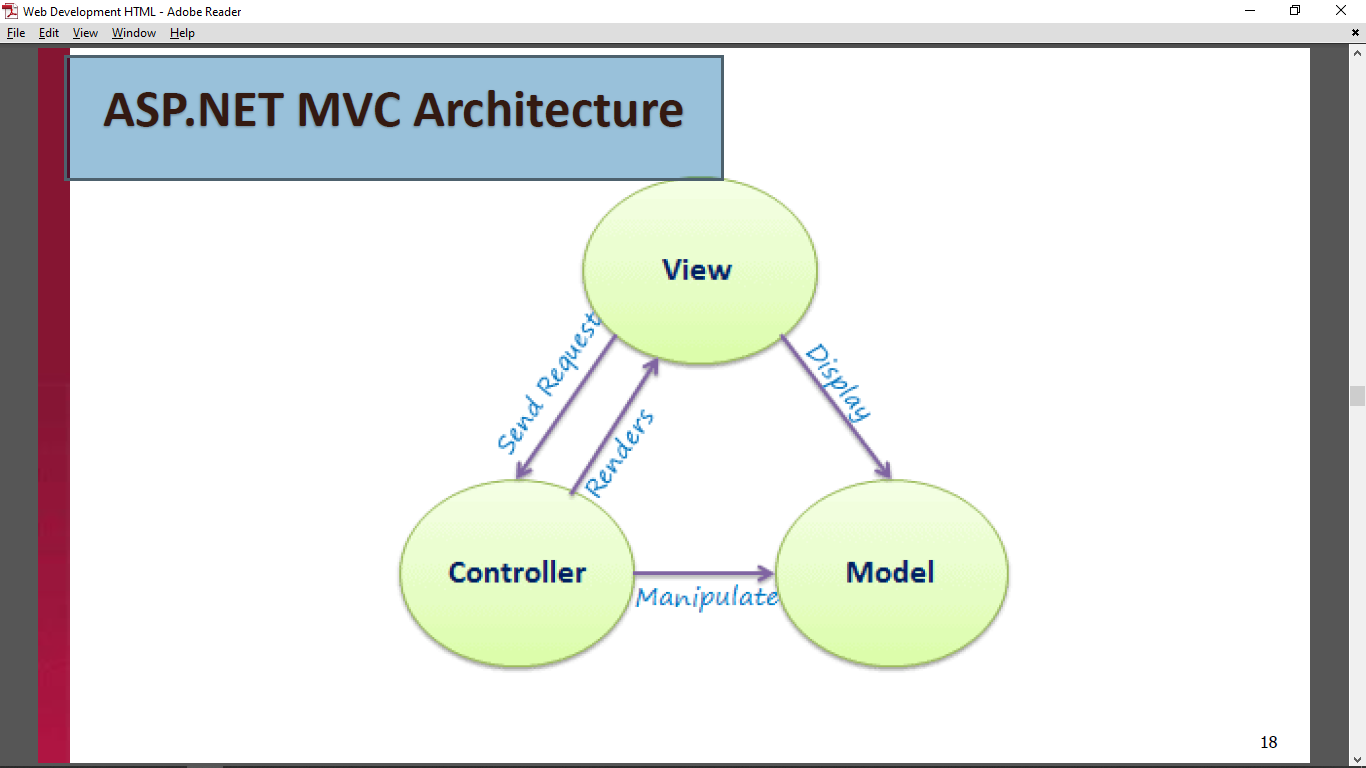
foreach(var name in myLinqQuery)

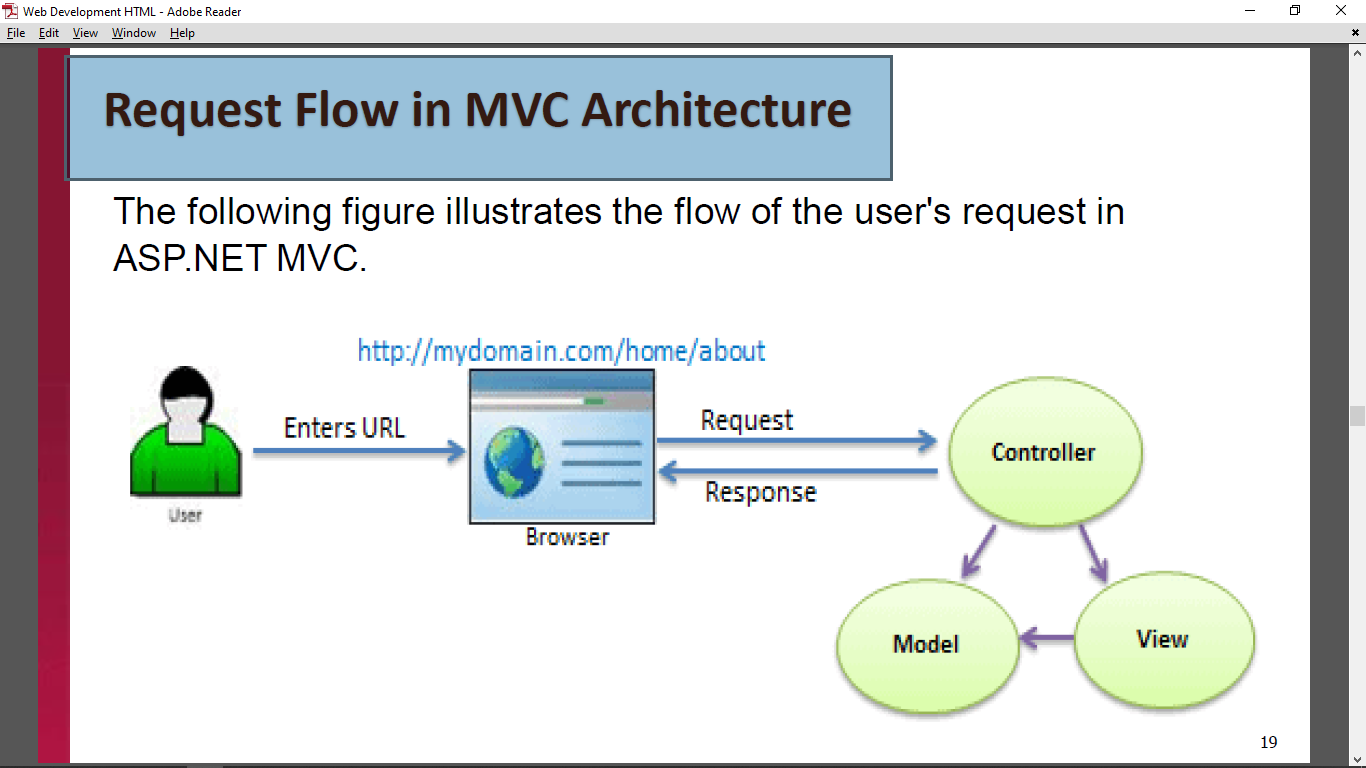
Console.Write(name + " ");

In the above example, string array name is a data source. In a variable myLinqQuery, a LINQ Query is assigned, which gets all the strings from an array which contains ‘a’. Finally, a foreach loop is used to output the results.

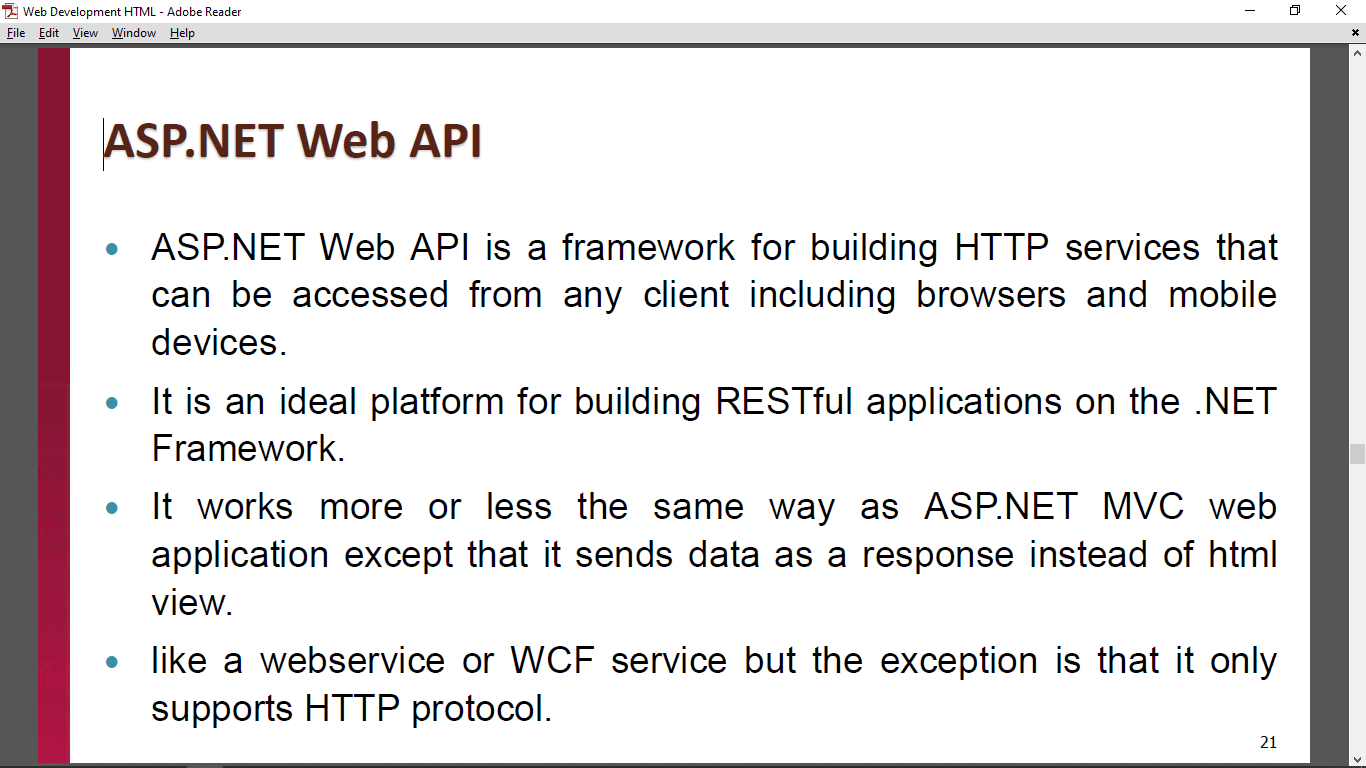
1. Explain the Model, Controller and View in MVC.

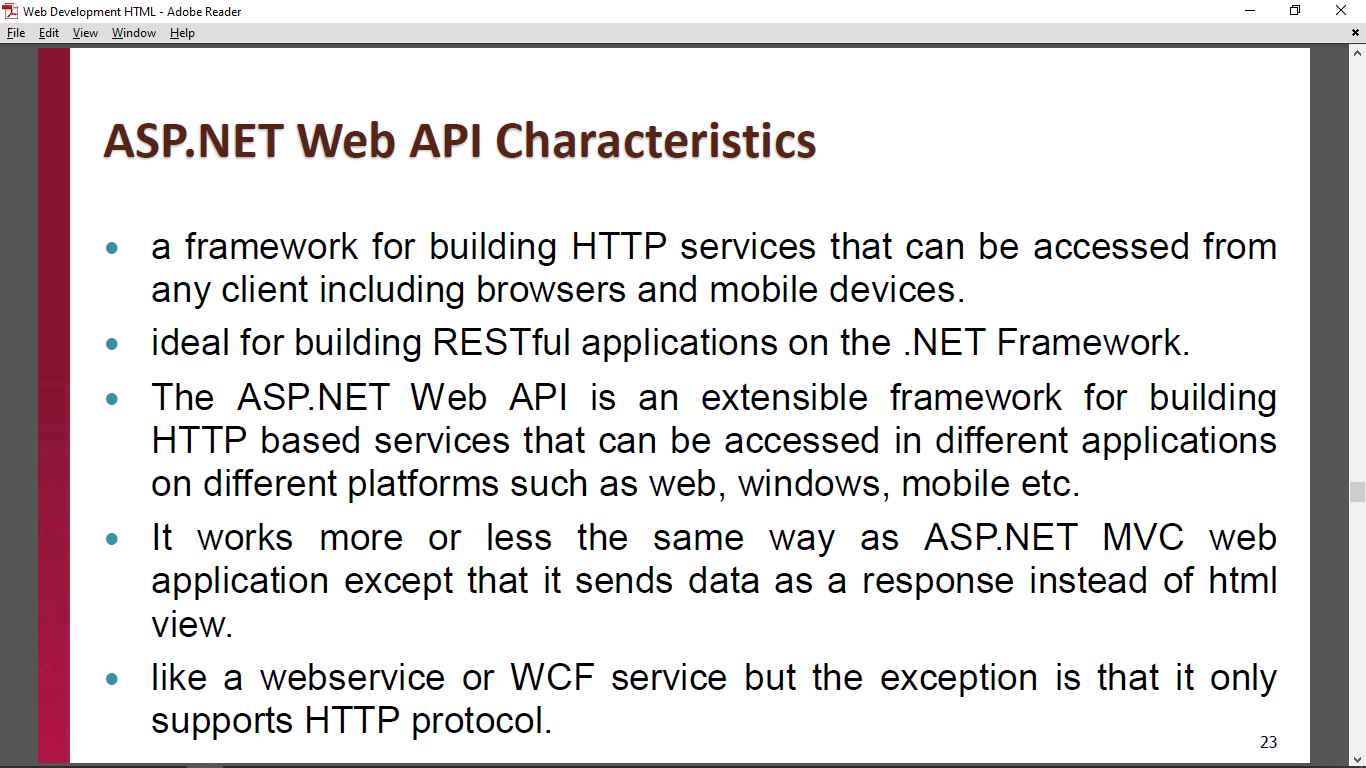




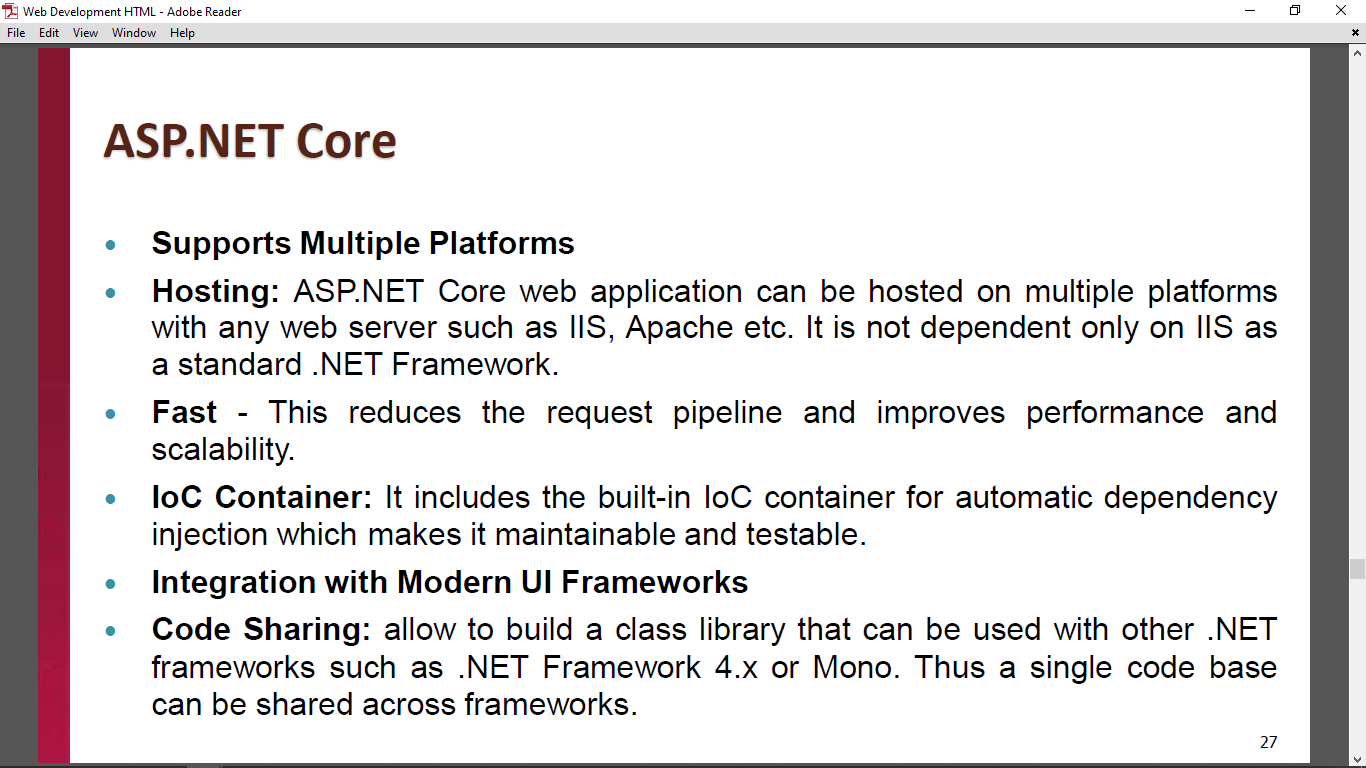


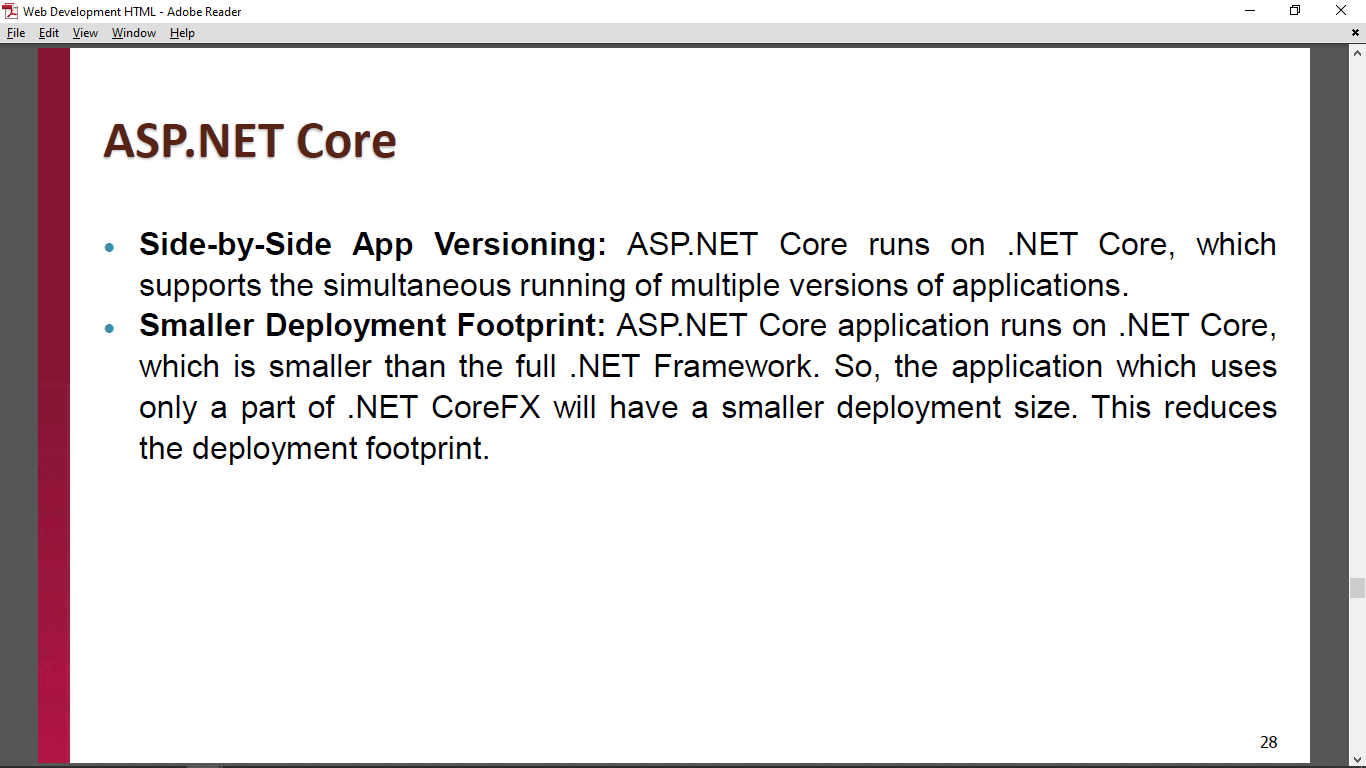
1. Define ASP.NET Web API and write its characteristics.

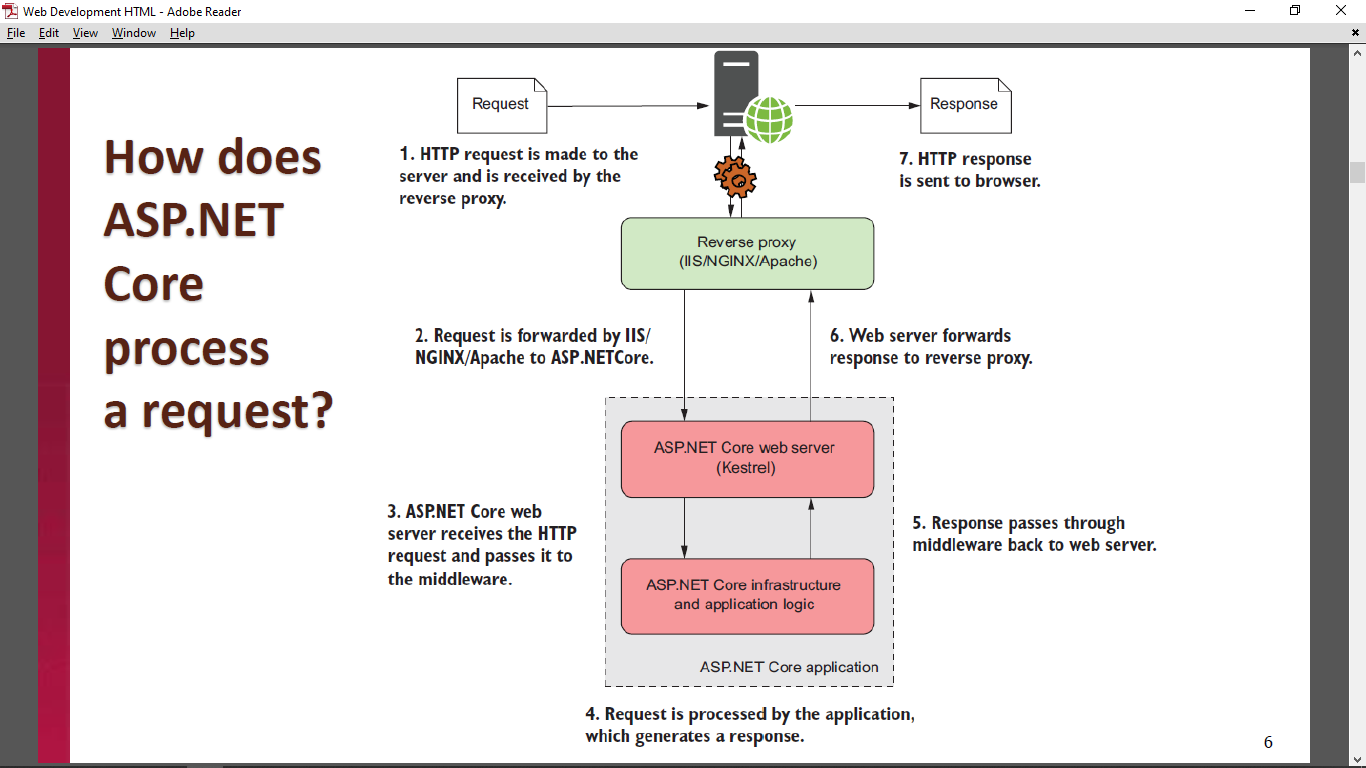




1. What are the features provided by ASP.NET Core? How does ASP.NET Core process a request?







1. Explain the common web application architectures.

