Extension class & uses

1. Using extension we can extend features of existing class as well as we can make code more readable and more maintainable.

Like in Startup class we have no if services where we as in Configure service method make of code structure non maintainable.

Public Static class ApplicationServices

{

Public static IServiceCollection **AddApplicationServices** (this IServiceCollection services)

{

services.AddScoped<IProductRepository, ProductRepository>(); // no of services we can add here

// For Generic

services.AddScoped<typeof(IGenericRepository<>),typeof(GenericRepository<>)>();

return services;

}

}

// In Startup.cs class

Public void ConfigureServices(IServiceCollection services)

{

Services. **AddApplicationServices**();

}

1. To Extend functionality of any any without using inheritance ( Because if we use inheritance we have create object of child class to use new feature )

Eg. We have Maths class having Add, Sub method , and Maths in some dll

Using Maths

Public class Static MathExtension

{

Public static int Multiply (this Maths, int I , int j)

{

Return i\*j;

}

}

// use

Maths m = new Maths();

m.Add();

m.Multiply(10,20); // output 200