



Reading material for Delegates & Lambda Functions

<https://inspireinnovativelearning.blogspot.com/2020/06/lambda-expressions-as-generic-delegates.html>

=====

ASSESSMENT LINK - TO BE COMPLETED BEFORE 2:30pm

<https://app.mymapit.in/code4/tiny/Lt1wMe>

=====

WEB API -Adding CORS Configurations

=====

STEP 1: Go to Program.cs. Just above the statement "builder.Build()", add the following

builder.Services.AddCors(

```
builder =>
{
    builder.AddPolicy(name: "AllowOrigin", corsPolicy =>
    {
        corsPolicy.AllowAnyOrigin()
            .AllowAnyHeader()
            .AllowAnyMethod();
    });
}
```

);

STEP 2: In Program.cs, Just above the line "app.Run()", add the following

app.UseCors("AllowOrigin");

STEP 3: In the KpmgController, add the following above the class

[EnableCors("AllowOrigin")]

<https://github.com/sampradan-labs/RandomNotesForPax>

<https://inspireinnovativelearning.blogspot.com/2022/09/approach-for-students-learners-of-new.html>

Reading Material

1. Class Inheritance

https://www.w3schools.com/cs/cs_inheritance.php

2. Interfaces

https://www.w3schools.com/cs/cs_interface.php

3. Abstract Class

https://www.w3schools.com/cs/cs_abstract.php

4. Virtual keyword with override

https://www.w3schools.com/cs/cs_polymorphism.php

=====

Tip for interviews / mcqs:

Overloading is known as Compile-Time polymorphism as even before you run your application, the compiler knows how many overloads for your method exist.

Overriding is known as Run-Time Polymorphism as only at run time we can know if the base class or derived class method will execute.

=====

The four pillars of OOPs

1. Inheritance

2. Encapsulation: Using public, private, protected, internal keywords so that you can share classes with other projects or not.

3. Abstraction: Using abstract keyword, using Interfaces

4. Polymorphism: Overriding methods