

Generics

Tuesday, February 4, 2025 9:17 AM

Attributes -- // to get some metadata information or provide some of the information to the developers .They also apply the attributes to add the metadata annotations

Some of the annotations are user defined and some of them are custom defined

```
Employee.cs
{
    Id ,name , salary etc....{get; set}
    printEmployee()
    showEmployee()
    ADD()
    Update()
    Delete()

}
```

Reflection // run time

Type class --- to get the information of a type (Class) GetMethods() , GetProperties() , GetConstructors()

Testing : -- BDD & TDD

```
Add() testAdd()
Subtract() testSubtract()
Multiply()
Divide()
```

In Dot Net : NUnit testing
NUNIT

Create a dll of login class where you have
One method DoLogin(string username , string password)

```
{
    // Logic - 1)string values It should be not be empty // Credentials must be provided
    2. Username is it equal admin and the password is admin
    3. Otherwise it will return invalid credentials
}
```

For the above three create three test cases
Check_username_null
Check_password_null
check

CLR -- Common language runtime
CTS -- Common Type Safety

GENERIC --> It is concept that allows us to define classes and methods with some placeholder

C# compiler replaces these placeholders with the specified type at compile time .It is used to create a general purpose classes and methods

```
List = new list(); // Here you can store any type of objects
List<T>
```

One scenario is :

Let's say you want to check the values of two integers

```
Logic -- AreEqual()
```

main

```
Bool equality = Operations.AreEqual("ABC" , "def")
```

```
Public static bool AreEqual(object value1 , object value2)
{
    Return value1 == value2
}
```

Drawback is to do the boxing and unboxing

2.

```
Public static bool AreEqual(int value1 , int value2)
{
    Return value1 == value2
}
```

```
Public static bool AreEqual(string value1 , string value2)
{
    Return value1 == value2
}
```

Drawback is that we are repeating the logic

So, to overcome with this problem we will generics in C#
Syntax : public static bool AreEqual<T>(T value1 ,T value2)

```
public static bool AreEqual<int>(int value1 ,int value2)
```

Create a generic class where you need to define T type of name and address and with that it will also print the message as you are welcome to join our community .