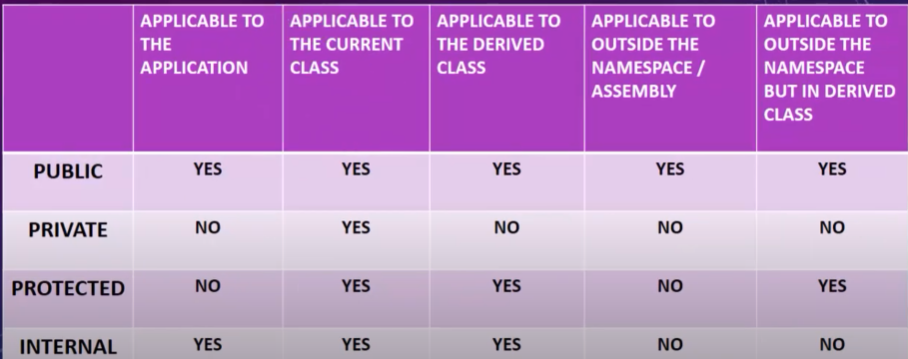
**Session 4: Access Specifier/Modifiers & Keywords**

**Keywords:** keywords are predefined, reserved identifiers that have special meaning to compiler. You cannot use keyword as identifiers unless @. Abstract, as, base, this, internal, public, private, protected, static, string,int, if, else, namespace,typeof, sizeof etc… keywords are known by the compiler.

**Access Modifiers:-** this are keywords. That define the accessibility of member,class,field, method datatype. Used to restrict unwanted data manipulation by external program or class. Each method can be used in same class.

**Types Of Access Modifiers/Specifier:**

****

**Public** is a keyword.this access modifiers can access from anywhere in project. It not restrict the access. Used in Static instance and static member. By default public access will be used. : Variable, method. **In c sharp private will used.**

* + **Applicable :-** Applicable to Application, Applicable to the Current Class, Applicable to the derived class, outside the namespace, outside namespace but in derived class.

**EG: int a = 20; >> public int a = 20;**

* **Private:-** its having least permission. it can be access only within the body of the class. Another class cant write or used the same variable or instance.
  + **Applicable :-** Applicable to the Current Class only.
* **Protected :-** same class
  + **Applicable :-** Applicable to the Current Class, Applicable to the derived class, outside namespace but in derived class.
* **Internal :-** within same project we can use.
  + **Applicable :-** Applicable to Applicaton, Applicable to the Current Class, Applicable to the derived class.

S4\_AccessModifiereAndKeyword.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace AllSession

{

public class S4\_AccessModifiereAndKeyword

{

public int no1;

int no11;

//Public

public void PublicArithmaticOperation() //Non Static Method

{

int no1 = 60, no2 = 20, sum;

Console.WriteLine($"Addition is: {no1 + no2}");

Console.WriteLine($"Substractino is : {no1 - no2}");

Console.WriteLine($"Multiplicaton is: {no1 \* no2} \n");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\* ProgramAssignment Called :- \*\*\*\*\*\*\* \n");

PrivateAssignmentOperator();

//ProtectedOperation();

}

//Private

private void PrivateAssignmentOperator() //PrivateMethod bydefault

{

string name = "Program";

Console.WriteLine("ProgramName: " + name + "\n");

//S4\_AccessModifiereAndKeyword core = new S4\_AccessModifiereAndKeyword();

//core.PublicArithmaticOperation();

}

//Protected

protected void ProtectedOperation()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\* Protected Method \*\*\*\*\*\*\*");

double num = 12.55;

int num2 = 10;

Console.WriteLine($"Addition is : {num + num2}");

}

//Internal

internal void InternalMethod()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\* Internal Method \*\*\*\*\*\*\*\*\*\*\*\*");

string firstName = "Imran", lastName = "Shaikh", final;

final = firstName + lastName;

Console.WriteLine("After Joining String " + final);

}

}

class ChildClass : S4\_AccessModifiereAndKeyword //Child Class OR Derived Class : Inheritance

{

public void PublicLogicalOperation() //Non Static Method

{

//no11 = 20; // PrivateVariable

no1 = 100;

int no12 = 200;

Console.WriteLine(no1 == no12);

Console.WriteLine(no1 > no12);

Console.WriteLine();

Console.WriteLine("\*\*\*\*\*\*\*\*\*\* Method Called \*\*\*\*\*\*\*\*\*\*");

ChildClass child = new ChildClass();

//child.PublicArithmaticOperation();

//child.PublicAssignmentOperator(); cannot accept

//S4\_AccessModifiereAndKeyword core = new S4\_AccessModifiereAndKeyword();

//core.PublicAssignmentOperator();

child.ProtectedOperation();

child.InternalMethod();

}

}

}

Program.cs

using AllSession;

using Session1.nestedNamespace; // Assembly Refrence OR Namespace OR PAckage

using System;

namespace Session1

{

class Program : S4\_AccessModifiereAndKeyword

{

static void Main(string[] args)

{

// Session 1 Introduction and NamesPace

//Namespace nameSpace = new Namespace();

//NestedName nest = new NestedName();

//nameSpace.Multiplication();

//nameSpace.Addition();

//nest.Multiplication();

//Console.ReadLine();

//Session 2 Varibale and Data Types

//S2\_VaribaleAndDatatypes varibaleDatatype = new S2\_VaribaleAndDatatypes();

//varibaleDatatype.VariablesAndUses();

//Session 3 Operator and Conversion

//S3\_OperatorAndTypeConversion operators = new S3\_OperatorAndTypeConversion();

//operators.OperatorAndConversion();

//Session 4 Access Modifieres And Keywords

S4\_AccessModifiereAndKeyword modifier = new S4\_AccessModifiereAndKeyword();

modifier.PublicArithmaticOperation();

//modifier.PrivateAssignmentOperator(); //need to PrivateMethod explain

Program prg = new Program();

//Child Class

ChildClass child = new ChildClass();

child.PublicLogicalOperation();

prg.ProtectedOperation();

}

}

}

NEW NAMESPACE(Create Project)

NameSpaceTwo.cs

using System;

using AllSession;

namespace NameSpaceTwo

{

class NameSpacesTwo : S4\_AccessModifiereAndKeyword //Child Class OR Derived Class

{

static void Main(string[] args)

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\* Inside the Second NameSpace \*\*\*\*\*\*\*\*\*\*\*");

S4\_AccessModifiereAndKeyword nameSpace2 = new S4\_AccessModifiereAndKeyword();

//nameSpace2.PublicArithmaticOperation();

NameSpacesTwo name = new NameSpacesTwo();

name.PublicArithmaticOperation();

name.ProtectedOperation();

}

}

}