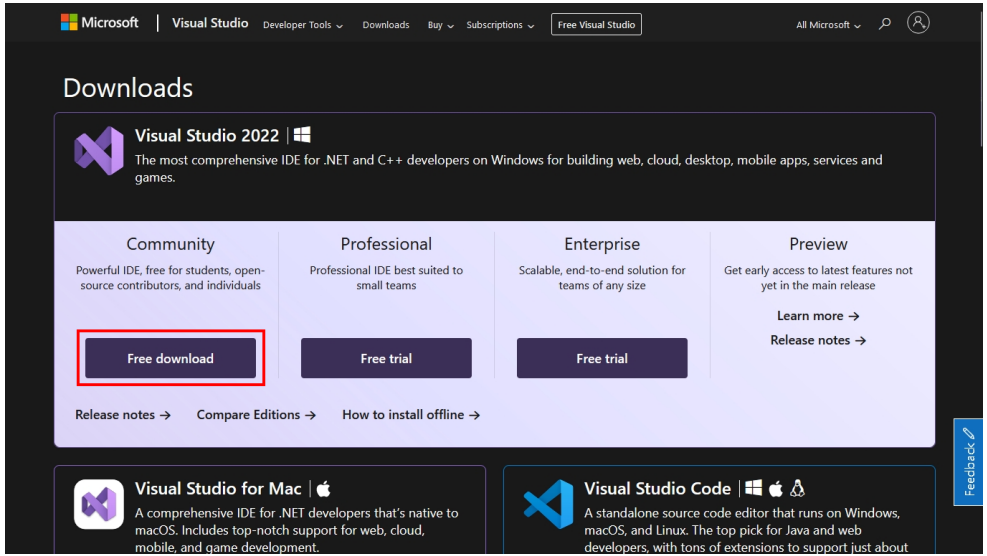
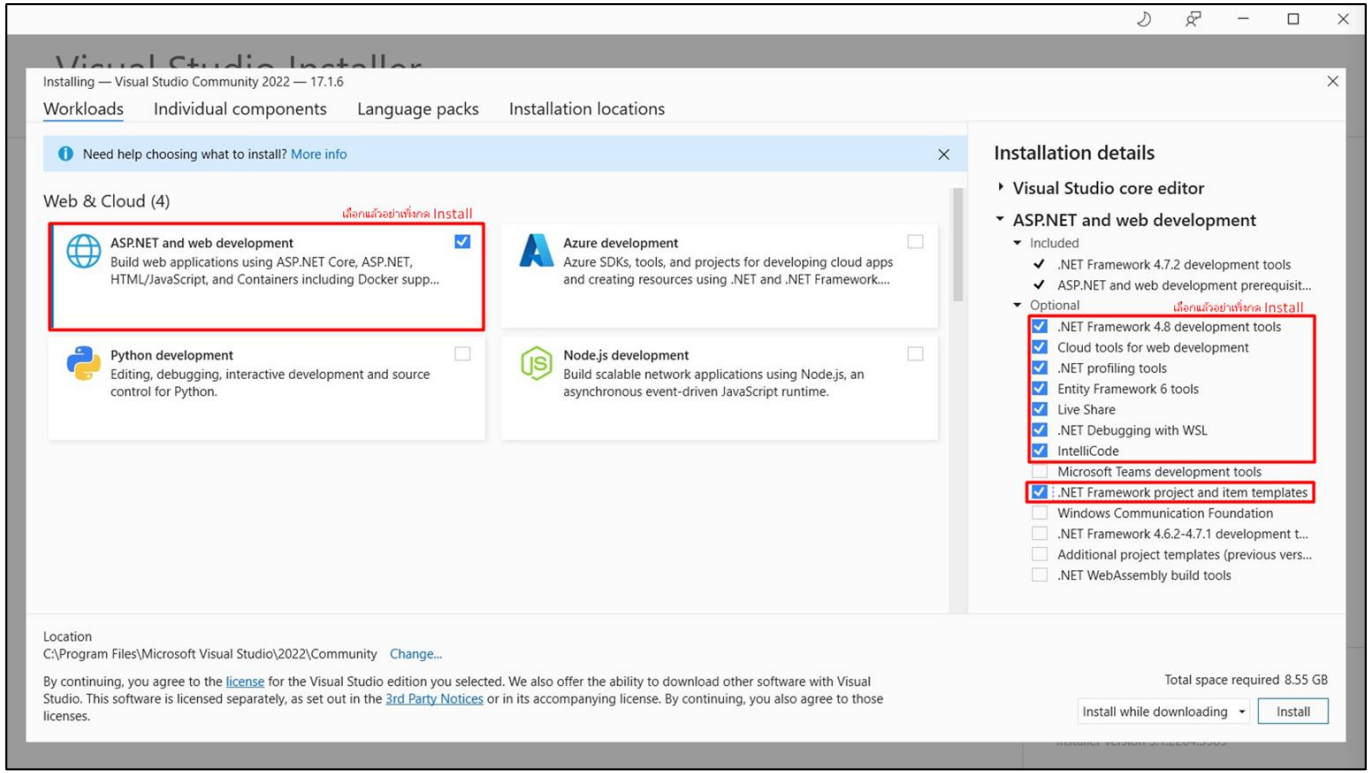
Prerequisite

1. ดาวน์โหลดและติดตั้ง Visual Studio Community ได้ที่ : https://visualstudio.microsoft.com/downloads/
2. สำหรับคนที่ไม่มี MySql อยู่ที่เครื่องให้ติดตั้ง wamp ได้ที่ : https://www.wampserver.com/en/
3. สำหรับคนที่มี Mysql แต่ไม่มีโปรแกรมสำหรับ Manage ฐานข้อมูลให้ทำการติดตั้ง Heidi ได้ที่ : https://www.heidisql.com/download.php



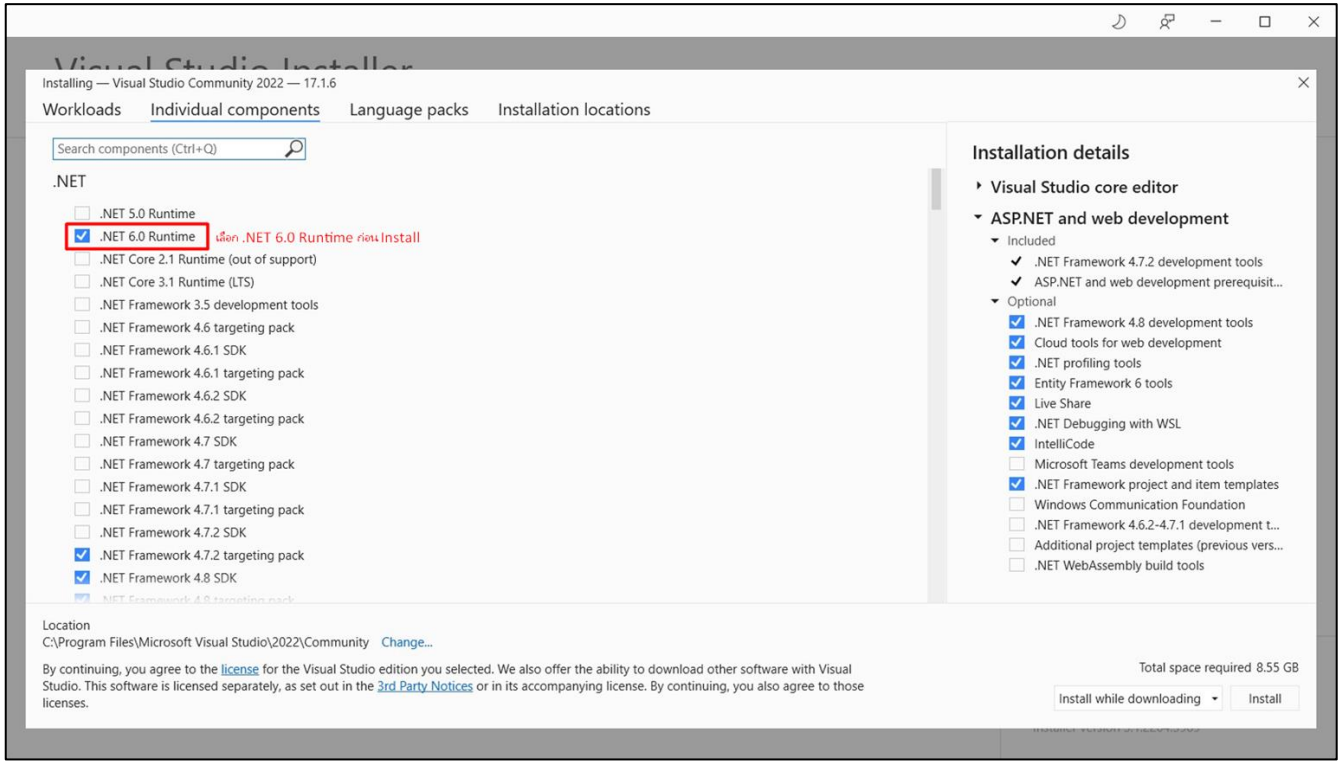
Prerequisite

ติดตั้ง Visual Studio



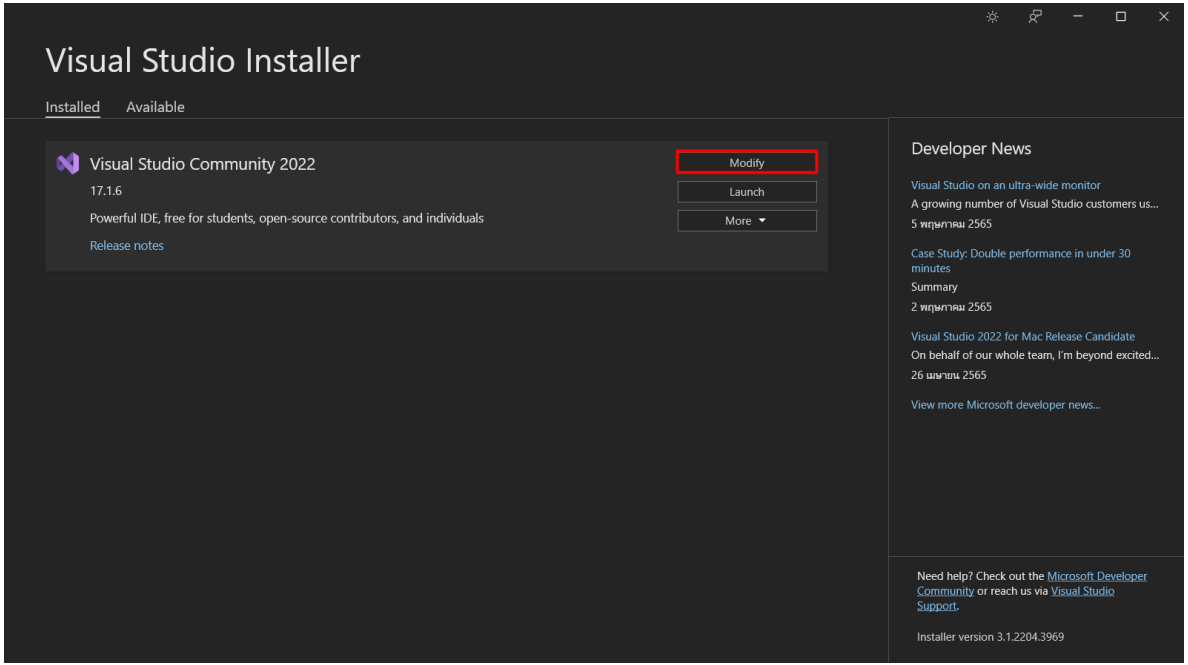
Prerequisite

ติดตั้ง Visual Studio



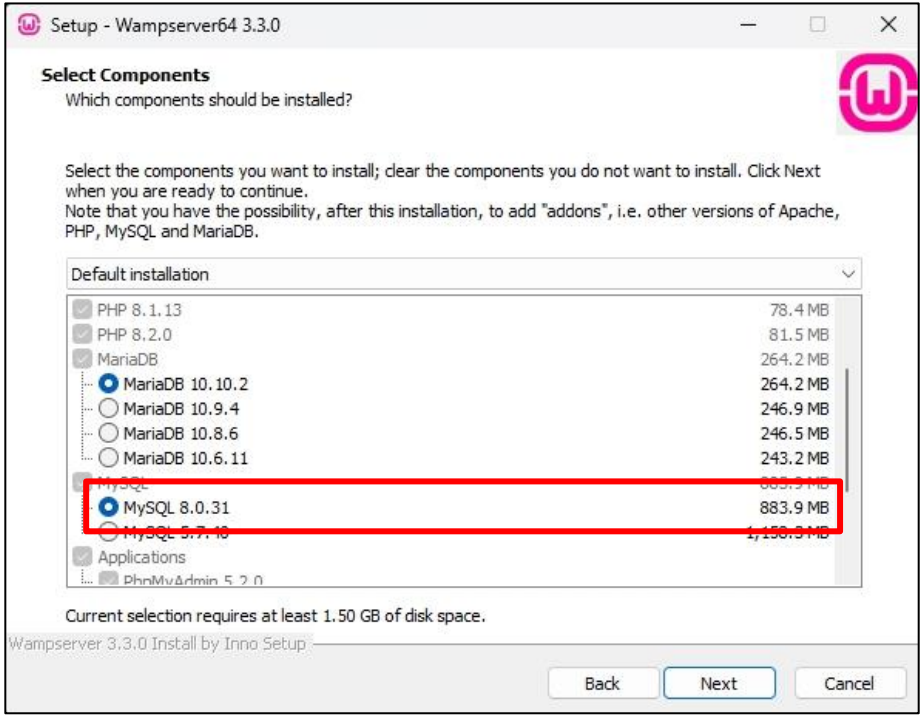
Prerequisite

กรณีที่เคยติดตั้ง Visual Studio แต่ยังไม่ได้ลง ASP .NET ให้เปิด Visual studio installer และเลือก Modify



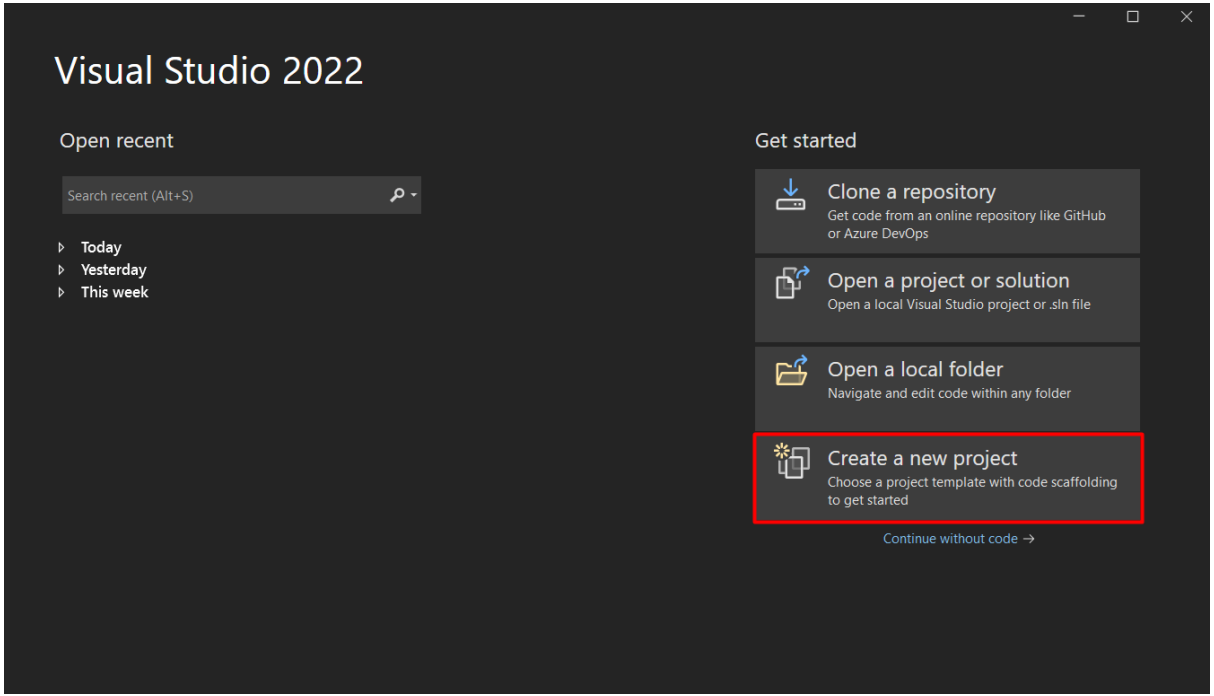
Prerequisite

ติดตั้ง Wamp Server



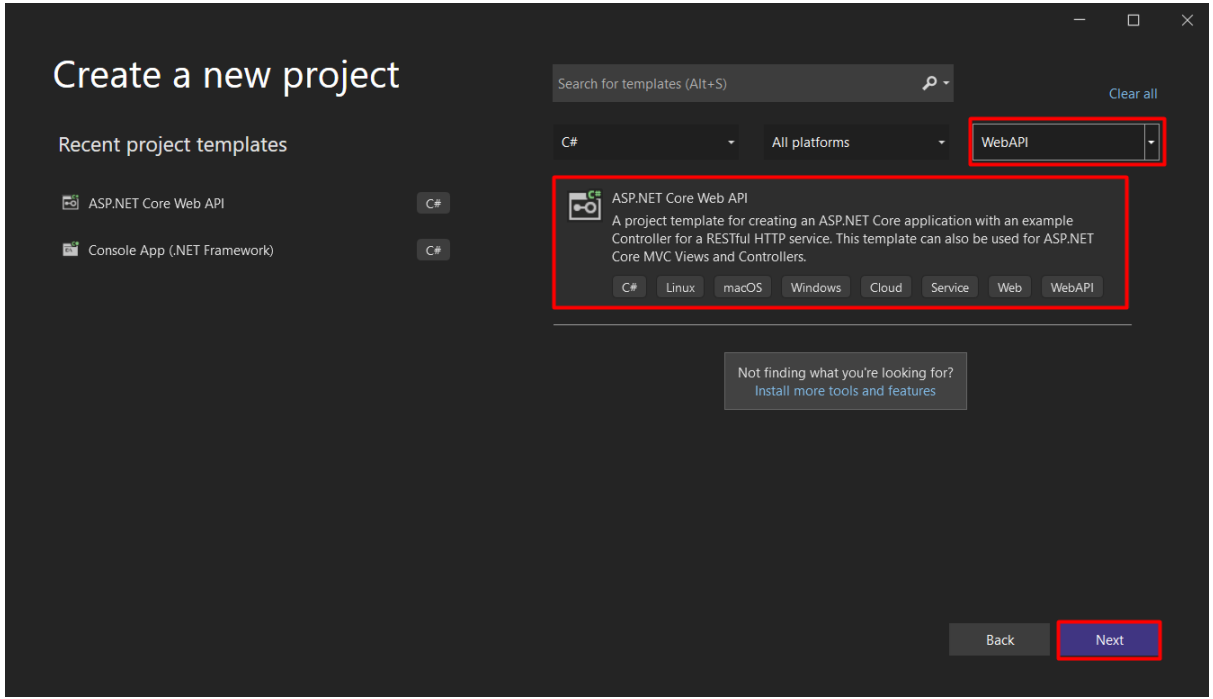
Prerequisite

ทดสอบสร้าง Project



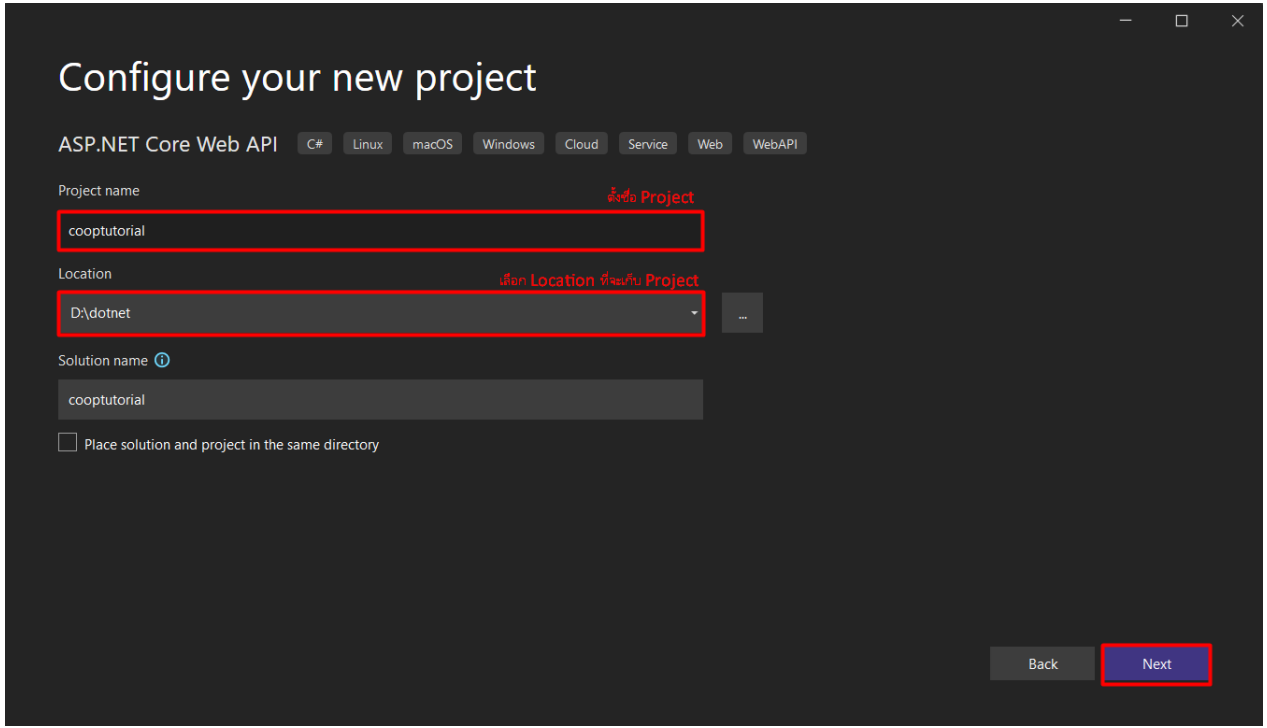
Prerequisite

ทดสอบสร้าง Project



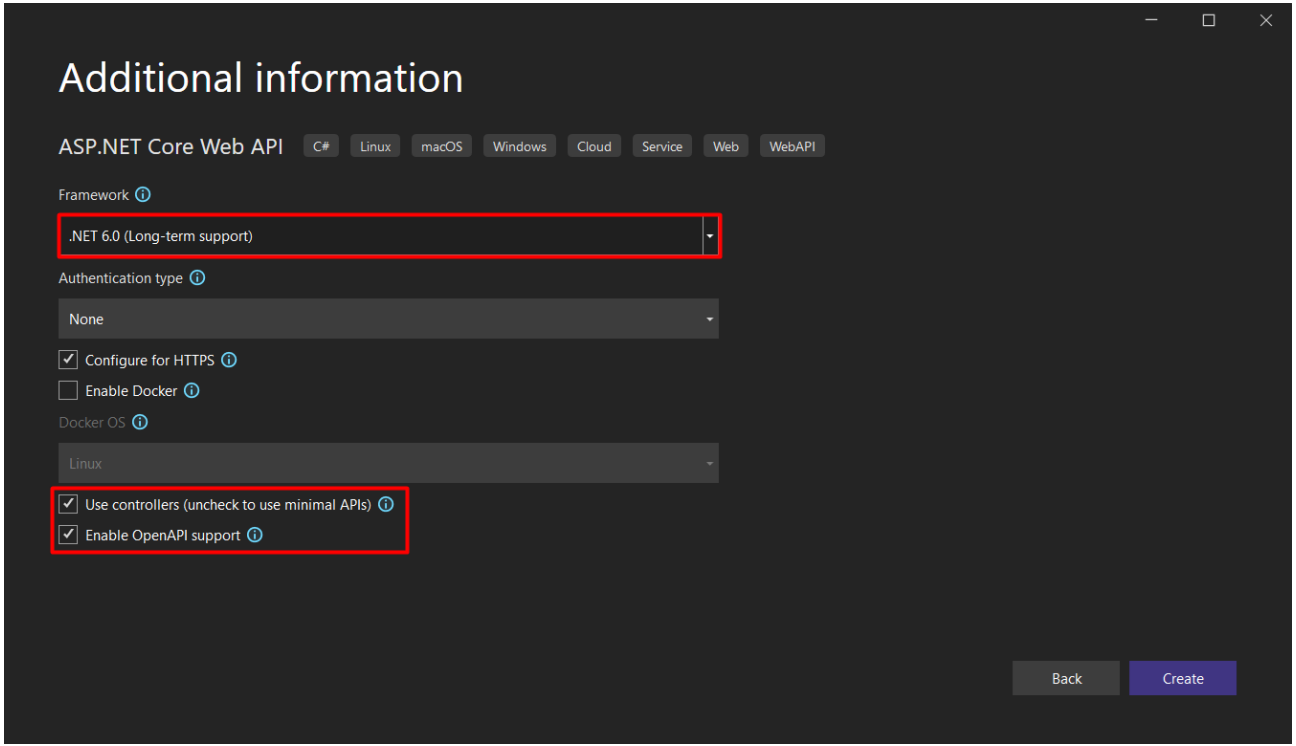
Prerequisite

ทดสอบสร้าง Project



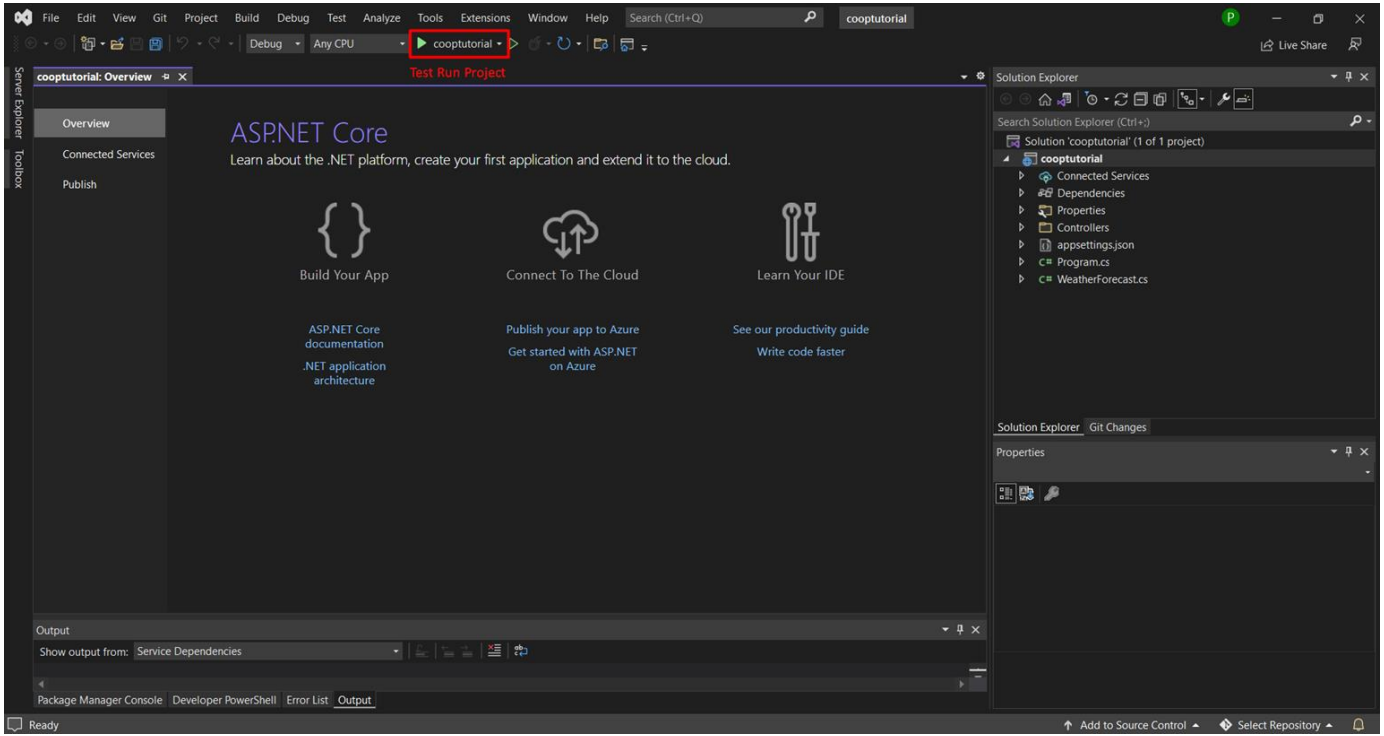
Prerequisite

ทดสอบสร้าง Project



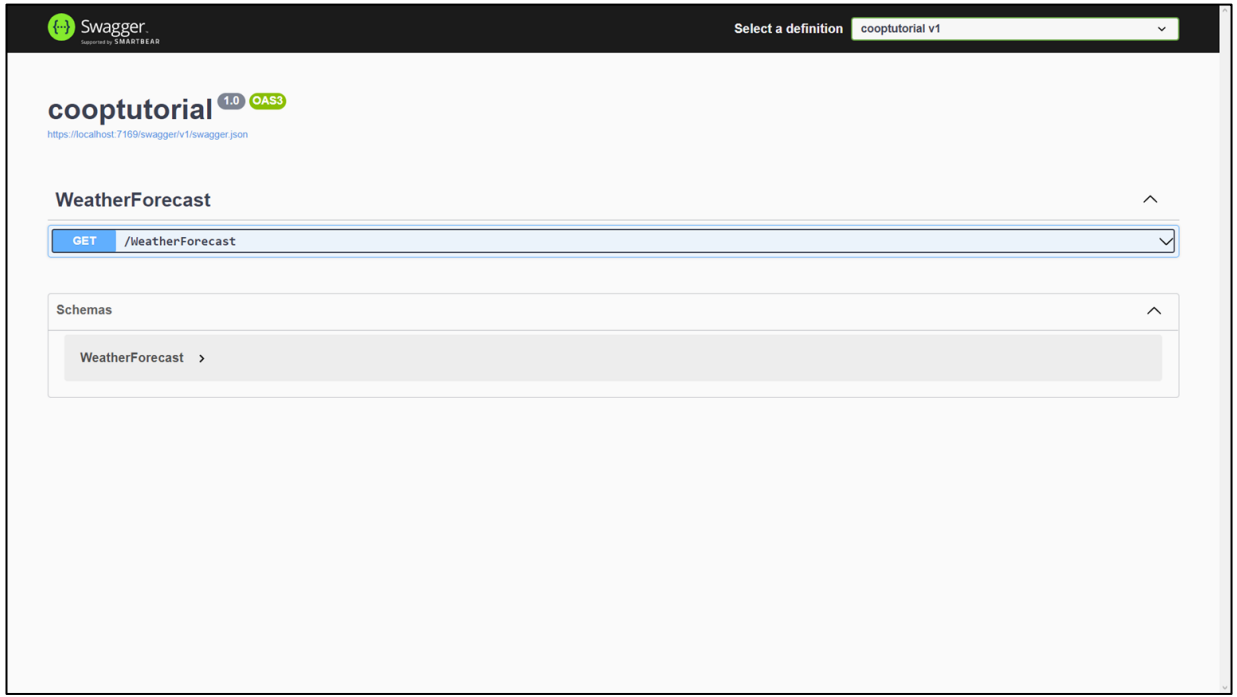
Prerequisite

ทดสอบ Run Project



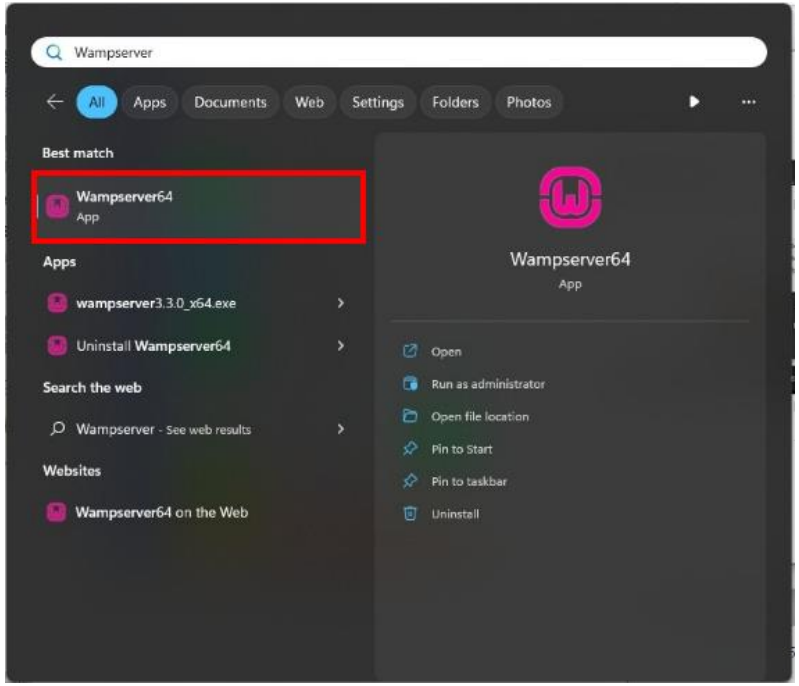
Prerequisite

ทดสอบ Run Project



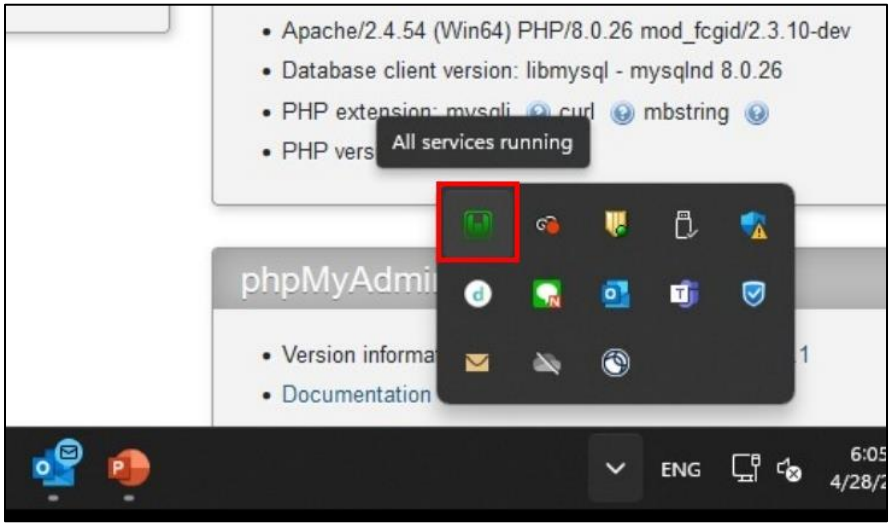
Prerequisite

ทดสอบ Start Wamp Server



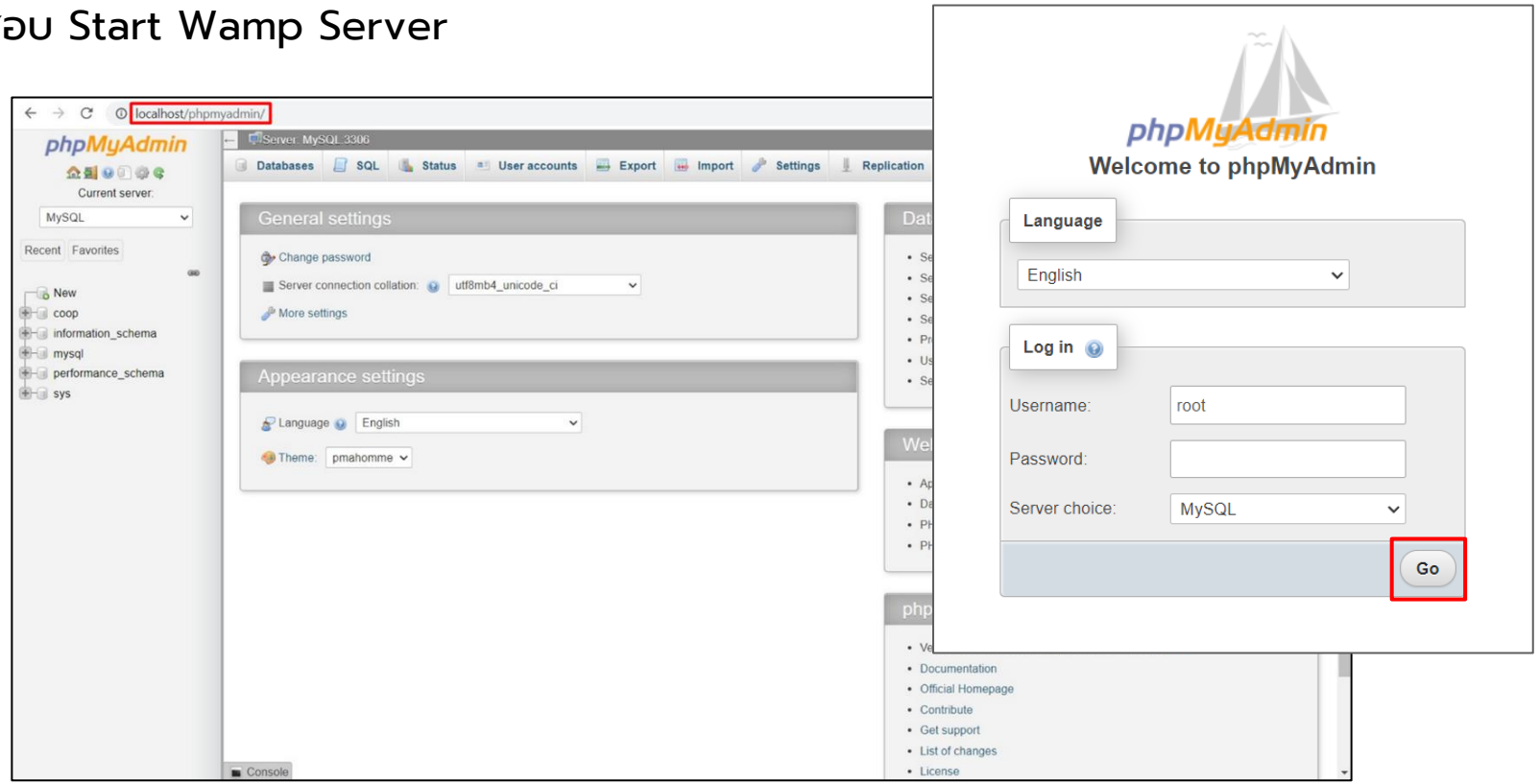
Prerequisite

ทดสอบ Start Wamp Server



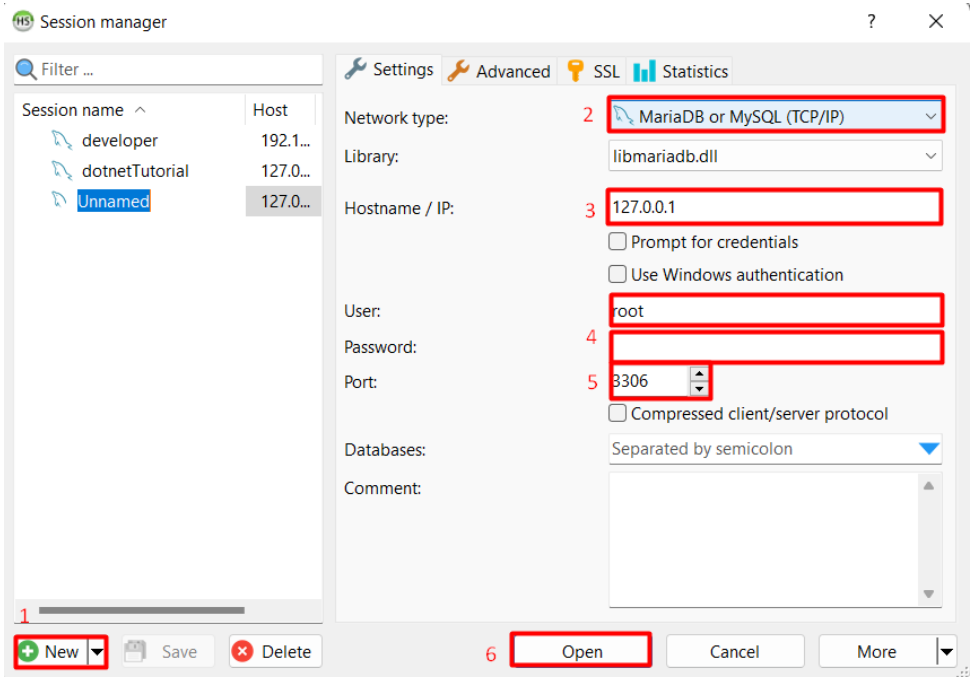
Prerequisite

ทดสอบ Start Wamp Server



Prerequisite

การใช้ HeidiSql สำหรับคนที่ไม่มี Database Management System



User และ Password ที่ใช้เข้า Database


C#

What is C#

C# คือ ภาษา Programming Language แบบ OOP หรือ Object-Oriented Programming โดย C# เป็นภาษาที่มีพื้นฐาน C++ และได้รับการออกแบบให้สามารถทำงานกับ .NET Platform ของ Microsoft ได้

Content

1. Basic C#
2. Operators
3. If else and switch
4. Loop
5. Collection Elements
6. Method and Class

Basic C#

Structure of C#

```
using System;                                   // Include Library & Function
namespace BasicProgramming                      // Used to organize your code
{
    0 references
    internal class Program                      // Container for data and methods
    {
        0 references
        static void Main(string[] args)         // Any code inside Main will be executed
        {
            Console.WriteLine("Hello World!");  // Your Code...
        }
    }
}
```

Basic C#

Console and Debug

- สามารถใช้ Console.WriteLine และ Console.Write ในการแสดง output ทาง console

```
Console.WriteLine("Hello World");
Console.WriteLine("Welcome to A-host");

Console.Write("This session is C# Tutorial");
Console.Write("First day");
```

```
Hello World
Welcome to A-host
This session is C# TutorialFirst day
```

Basic C#

Console and Debug

```
Console.WriteLine("Hello {0} and {1}!","World","A-host");
Console.WriteLine("Anniversary {0} years",2023 - 1999);
```

```
Hello World and A-host!
Anniversary 24 years
```

Basic C#

Variable

```
type variableName = value;
```

Basic C#

Variable

```
int height = 173;
long miles = 100L;
float weight = 63.2F;
double BMI = 21.12D;
```

Basic C#

Data Type

Data Type	Size	Description
int	4 bytes	สามารถเก็บตัวเลขได้ตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
long	8 bytes	สามารถเก็บตัวเลขได้ตั้งแต่ -9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807
float	4 bytes	สำหรับเก็บเลขทศนิยม สามารถเก็บทศนิยมได้ 6 ตำแหน่ง
double	8 bytes	สำหรับเก็บเลขทศนิยม สามารถเก็บทศนิยมได้ 15 ตำแหน่ง
bool	1 bit	เก็บข้อมูลที่เป็น true , false
char	2 bytes	สำหรับเก็บข้อมูล 1 ตัวอักษร โดยข้อมูลจะต้องอยู่ภายใต้ ' '
string	2 bytes/character	สามารถเก็บข้อมูลตัวอักษรหลายตัวได้ โดยข้อมูลจะต้องอยู่ภายใต้ " "

Basic C#

Data Type

```
int validMaxNum = 2147483647;
int validMinNum = -2147483648;
int InvalidMaxNum = 2147483648;
int InvalidMinNum = -2147483649;
```

Basic C#

Type Casting

Implicit Casting คือ การแปลงประเภทข้อมูลจากประเภทที่เล็กไปประเภทที่ใหญ่กว่า โดยสามารถแปลงได้แบบอัตโนมัติ

Char > Int > Long > Float > Double

Explicit Casting คือการแปลงประเภทข้อมูลจากประเภทที่ใหญ่กว่าไปประเภทที่เล็กกว่า โดยต้องเพิ่มคำสั่งที่ใช้ในการแปลงไปเป็นประเภทข้อมูลนั้นๆ

Char < Int < Long < Float < Double

Basic C#

Type Casting

```
static void Main(string[] args)
{
    //Implicit Casting
    char myChar = '1';
    int myInt = myChar;
    long myLong = myInt;
    float myFloat = myLong;
    double myDouble = myFloat;

    Console.WriteLine(myDouble);
}
```

Output = 49

Basic C#

Type Casting

```
static void Main(string[] args)
{
    //Explicit Casting
    double myDouble = 77.3D;
    float myFloat = (float) myDouble;  // 77.3 casting to float = 77.3
    long myLong = (long) myFloat;      // 77.3 casting to long  = 77
    int myInt = (int) myLong;          // 77   casting to int   = 77
    char myChar = (char) myInt;        // 77   casting to char  = M in ASCII Code

    Console.WriteLine(myChar);
}
```

Output = M

Basic C#

Type Casting with Convert Method

- สามารถใช้ Method Convert ตามด้วยประเภทของข้อมูลที่จะเปลี่ยนได้

```
static void Main(string[] args)
{
    //use Convert Method
    Char character = 'M';
    Console.WriteLine("{0} is {1} in ASCII Code"
                      ,character
                      ,Convert.ToInt32(character));
}
```

Output = M is 77 in ASCII Code

Basic C#

Type Casting with Parse Method

- สามารถใช้ Method Parse เพื่อเปลี่ยนประเภทของข้อมูล

```
static void Main(string[] args)
{
    string numtext = "100";
    int num = int.Parse(numtext);
    long numL = long.Parse(numtext);
    float numF = float.Parse(numtext);
    double numD = double.Parse(numtext);

    Console.WriteLine($"int = {num}, long = {numL}, float = {numF}, double = {numD}");
}
```

Output = int = 100, long = 100, float = 100, double = 100

Basic C#

Input

- สามารถใช้ Console.ReadLine() ในการรับข้อมูล String (เท่านั้น)

```
static void Main(string[] args)
{
    //use Console.ReadLine() to input Data

    Console.Write("Input Username : ");
    string username = Console.ReadLine();

    Console.WriteLine("Your username is {0}", username);
}
```

```
Input Username : Peach
Your username is Peach
```

User Input

Output

Basic C#

Input

- หากต้องการรับข้อมูลเป็นประเภทอื่นๆ เช่น int float ให้เพิ่ม Method Convert ตามภาพ

```
static void Main(string[] args)
{
    Console.Write("Input age : ");
    int age = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Your Age is {0}", age);
}
```

```
Input age : 22
Your Age is 22
```

Basic C#

Math Method

Method	Description	Example	Output
Math.Max(x,y)	หาค่าที่มากที่สุดระหว่างสองค่า	Math.Max(5,10)	10
Math.Min(x,y)	หาค่าที่น้อยที่สุดระหว่างสองค่า	Math.Min(5,10)	5
Math.Sqrt(x)	หาค่า square root	Math.Sqrt(64)	8
Math.Abs(x)	หาค่า Absolute	Math.Abs(-9.7)	9.7
Math.Round(x)	ปัดเลขทศนิยมตามหลักคณิตศาสตร์	Math.Round(9.7)	10
Math.Pow(x,y)	ยกกำลัง	Math.Pow(2,3)	8

Basic C#

String Method

Name	Description	Example (name = Ahost)	Output
Length	หาจำนวนตัวอักษร	name.Length	5
Substring	ตัดตัวอักษรโดยระบุตำแหน่งที่จะตัดและจำนวนตัวอักษร	name.SubString(1,3)	hos
IndexOf	หาตำแหน่งของตัวอักษรหรือข้อความที่ระบุ	name.IndexOf("ho")	1
Replace	แก้ไขตัวอักษรหรือข้อความที่กำหนดเป็นคำใหม่	name.Replace("A","B")	Bhost
Upper	แปลงตัวอักษรเป็นตัวพิมพ์ใหญ่	name.ToUpper()	AHOST
Lower	แปลงตัวอักษรเป็นตัวพิมพ์เล็ก	name.ToLower()	ahost

Lab 01 – Vairable and Debug (10 mins)

Input

- สร้างตัวแปรรับข้อมูล **ชื่อ**, **น้ำหนัก** (kg) , **ส่วนสูง** (cm) ผ่านทาง Console ทั้งหมด 2 คน และให้คำนวณหาค่า BMI ตามสูตรที่ให้ด้านล่าง

Output

- ให้แสดงชื่อที่รับเข้ามาเป็นตัวพิมพ์ใหญ่, ค่า BMI (ทศนิยม 2 ตำแหน่ง) และ ให้แสดงค่า BMI ที่มากที่สุด ตามตัวอย่างด้านล่าง

```
---Student 1---
Name : Peach
Input Weight(kg) : 64
Input Height(cm) : 175
---Student 2---
Name : Book
Input Weight(kg) : 64
Input Height(cm) : 190

PEACH BMI = 20.9
BOOK BMI = 17.73

Max BMI = 20.9
```

สูตรคำนวน BMI

น้ำหนักตัว[kg] / (ส่วนสูง[m] ยกกำลังสอง)

Operators

Arithmetic Operators

X = 7, Y = 3

Operator	Name	Description	Example
+	Addition	บวก	X + Y = 10
-	Subtraction	ลบ	X – Y = 4
*	Multiplication	คูณ	X * Y = 21
/	Division	หาร	X / Y = 2.33
%	Modulus	คำนวนหาเศษจากการหาร	X % Y = 1
++	Increment	เพิ่มจำนวนทีละ 1	++X , X++ = 8
--	Decrement	ลดจำนวนทีละ 1	--Y , Y-- = 2

Operators

Assignment operators

X = 7

Operator	Example	Same as	Output
=	X = 3	X = 3	3
+=	X += 3	X = X + 3	10
-=	X -= 3	X = X – 3	4
*=	X *= 3	X = X * 3	21
/=	X /= 3	X = X / 3	2.33
%=	X %= 3	X = X % 3	1

Operators

Comparison Operators

X = 7, Y = 3

Operator	Name	Example	Output
==	Equal to	X == Y	False
!=	Not Equal	X != Y	True
>	Greater than	X > Y	True
<	Less than	X < Y	False
>=	Greater than or Equal to	X >= Y	True
<=	Less than or Equal to	X <= Y	False

Operators

Logical Operators

X = 7

Operator	Name	Description	Example	Result
&&	And	Return True เมื่อทั้งสองเงื่อนไขเป็นจริง	(x < 5) && (x < 10)	False
||	Or	Return True เมื่อเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง	(x < 5) || (x < 10)	True
!	Not	Return ค่าตรงข้าม	! ((x < 5) && (x < 10))	True

If else and Switch

If...else

```
string Role = "Admin";

if (Role == "Admin") //Check Condition1
{
    Console.WriteLine("Welcome Admin you can use all feature");
}
else if (Role == "User") //if Condition 1 invalid Check Condition2
{
    Console.WriteLine("Welcome User you can use some feature");
}
else // if no valid condition then do something...
{
    Console.WriteLine("Welcome Guest you can see Blog in homepage");
}
```

If else and Switch

Switch

```
static void Main(string[] args)
{
    string Role = "Admin";
    switch (Role)
    {
        case "Admin":
            Console.WriteLine("Welcome Admin you can use all feature");
            break;
        case "User":
            Console.WriteLine("Welcome User you can use some feature");
            break;
        default:
            Console.WriteLine("Welcome Guest you can see Blog in homepage");
            break;
    }
}
```

Lab 02 – Salary Increase condition(10 mins)

1. ให้กำหนดตัวแปร position สำหรับเก็บข้อมูลตำแหน่งงานของเราเป็น String
2. กำหนดตัวแปร salary สำหรับเก็บข้อมูลเงินเดือนเป็น float ตามที่ผู้ใช้กรอกข้อมูล
3. ให้ทำการสร้างเงื่อนไขโดยใช้ if...else หรือ switch ดังนี้
4. ทำการแสดง output ว่าเงินเดือนเพิ่มขึ้นมาเป็นเท่าไหร่

position	Description
Assistant Developer	เพิ่มเงินเดือน 5%
IT Support	เพิ่มเงินเดือน 7%
Senior Developer	เพิ่มเงินเดือน 10%
Administrator	เพิ่มเงินเดือน 8%
อื่นๆ	เพิ่มเงินเดือน 3%

Lab 02 – Salary Increase condition(10 mins)

```
Your Position : Assistant Developer
Your Salary : 28000
Your Salary will be increased to = 29400
```

Loop

While Loop

- วนลูปเมื่อเงื่อนไขเป็นจริง

```
static void Main(string[] args)
{
    int i = 0;
    while (i < 5) // 1. Check Condition
    {
        // 2. if Condition is True execute the code block
        Console.WriteLine(i);
        i++;
        // 3. Check Condition in while again
    }
}
```

Loop

Do-While Loop

- รันคำสั่งใน do ก่อน 1 ครั้งและวนลูปเมื่อเงื่อนไขใน while เป็นจริง

```
static void Main(string[] args)
{
    int i = 0;
    do // 1. execute the code block once
    {
        Console.WriteLine(i);
        i++;
    }
    while (i < 5); // 2. check condition if true execute code block again
}
```

Loop

For Loop

- วนลูปตามจำนวนที่กำหนด

```
static void Main(string[] args)
{
    for (int i = 0; // 1. Assign default data for first time
         i < 5; // 2. Chcek Condition
         i++) // 4. Increase data and Check Condition again
    {
        Console.WriteLine(i); // 3. Execute code block
    }
}
```

Loop

Foreach Loop

- วนลูปตามจำนวนข้อมูลใน Collections

```
static void Main(string[] args)
{
    string[] cars = { "Volvo", "BMW", "Ford", "Mazda" };
    foreach (string i in cars) // i represent of data in array
    {
        Console.WriteLine(i);
    }
}
```

Loop

Break and Continue

Break คือ คำสั่งที่ใช้ในการออกจาก Switch หรือ Loop ที่เหลือทั้งหมด

Continue คือ คำสั่งที่ใช้ในการ Skip Loop 1 ครั้งและหากเงื่อนไขในการวนซ้ำยังเป็นจริงอยู่ก็จะเริ่มการวนซ้ำรอบถัดไป

Loop

Break

```
static void Main(string[] args)
{
    for (int i = 0; i < 10; i++)
    {
        if (i == 4)
        {
            break; // break out of loop when i == 4
        }
        Console.Write("{0} ",i);
    }
}
```

Output = 0 1 2 3

Loop

Continue

```
static void Main(string[] args)
{
    for (int i = 0; i < 10; i++)
    {
        if (i == 4)
        {
            continue; // skip 4
        }
        Console.Write("{0} ",i);
    }
}
```

Output = 0 1 2 3 5 6 7 8 9

Collection Element

Array

- เป็นโครงสร้างข้อมูลที่เก็บข้อมูลชนิดเดียวกันเช่น String, int เป็นต้น โดย Array นั้นจะมีการกำหนดขนาดไว้ตั้งแต่ตอนสร้างชัดเจน

```
string[] carsEmpty;
```

Collection Element

Array Method

Method	Description	Example
Sort	เรียงข้อมูลจากมากไปน้อย	Array.Sort(**x**);
Reverse	สลับตำแหน่ง Index จากหลังมาหน้า	Array.Reverse(**x**);
Copy	คัดลอกข้อมูลใน Array ไปให้ Array อื่น	Array.Copy(**x**,**y**,index);
IndexOf	ค้นหาค่าใน Array โดยจะส่งข้อมูลกลับมาเป็น Index ของข้อมูล	Array.IndexOf(**x**, value);
Resize	เปลี่ยนขนาดของ Array ใหม่	Array.Resize(ref **x**, length);
Clear	ลบข้อมูลทั้งหมดใน Arrays	Array.Clear(**x**)

Collection Element

List

- เป็นโครงสร้างข้อมูลที่เก็บข้อมูลชนิดเดียวกันเหมือน Array แต่ List ไม่ต้องมีการระบุขนาดตอนสร้างเหมาะกับการเก็บข้อมูลที่มีจำนวนไม่แน่นอน

```
List<string> carsList;
```

Collection Element

List Method

Method	Description	Example
Add	เพิ่มข้อมูลใน List	carsList.Add(**x**);
Remove	ลบข้อมูลที่อยู่ใน List	carsList.Remove(**x**);
Clear	ลบข้อมูลทั้งหมดที่อยู่ใน List	carsList.Clear();
Sort	เรียงลำดับข้อมูลใน List	carsList.Sort();
Contain	ตรวจสอบว่ามีข้อมูลที่ระบุอยู่ใน List หรือไม่	carsList.Contains(**x**);
IndexOf	หา index ของข้อมูลที่ต้องการใน List	carsList.IndexOf(**x**);

Collection Element

Set

- เป็นโครงสร้างข้อมูลที่เก็บข้อมูลชนิดเดียวกัน โดย Set จะไม่สามารถเก็บข้อมูลที่มีค่าซ้ำกันได้

```
HashSet<string> fruits;
```

Collection Element

Set Method

Method	Description	Example
Add	เพิ่มข้อมูลใน Set	fruits.Add(**x**);
Remove	ลบข้อมูลที่อยู่ใน Set	fruits.Remove(**x**);
Contain	ตรวจสอบว่ามีข้อมูลที่ระบุอยู่ใน Set หรือไม่	fruits.Contains(**x**);
Count	นับจำนวนข้อมูลใน Set	fruits.Count();
IntersectWith	Return ค่าที่ Intersect กันระหว่าง 2 set	fruits.IntersectWith(fruits2);
UnionWith	Return ค่าที่ Union กันระหว่าง 2 set	fruits.UnionWith(fruits2);
ExceptWith	Return ค่าที่ Except กันระหว่าง 2 set	fruits.ExceptWith(fruits2);

Collection Element

Dictionary

- เป็นโครงสร้างข้อมูลที่เก็บข้อมูลแบบ Key-Value pairs โดยจะใช้ Key เป็นตัวกำหนดข้อมูล และ Value คือข้อมูลที่เก็บไว้

```
Dictionary<string, int> citiesPopulation;
```

Collection Element

Dict Method

Method	Description	Example
Add	เพิ่มข้อมูลใน Dict	myDict.Add(“Key", Value);
Remove	ลบข้อมูลที่อยู่ใน Dict	myDict.Remove(“Key”);
ContainsKey, ContainsValue	ตรวจสอบว่ามีข้อมูล Keys หรือ Value ที่ระบุอยู่ใน Dict หรือไม่	myDict.ContainsKey(“Key"), myDict.ContainsValue(Value)
Count	นับจำนวนข้อมูลใน Dict	myDict.Count;
TryGetValue	หา value ของ key และ return ค่าออกมาเป็น Boolean	myDict.TryGetValue(“Key", out value)
Keys, Values	Return ค่า Keys หรือ Value ออกมาเป็น Collection	myDict.Keys, myDict.Values

Lab 03 – Loop insert Student score(20 mins)

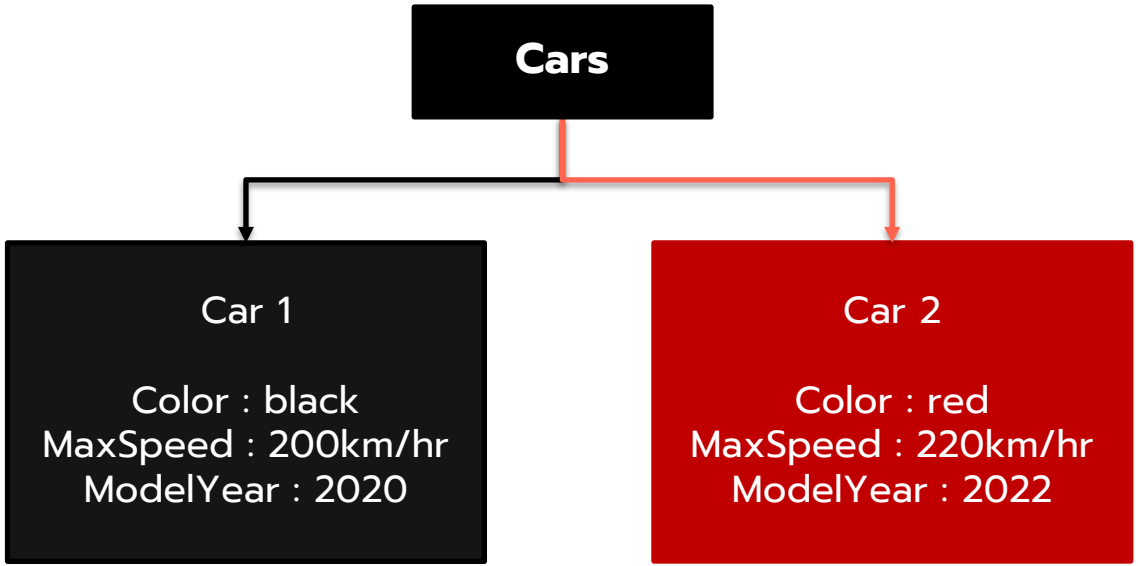
- สร้างตัวแปรเก็บข้อมูลจำนวนนักเรียนที่จะกรอกคะแนน
- วนลูปรับข้อมูลรหัสนักเรียน และคะแนนพร้อมแสดงผลตามตัวอย่าง
- หลังรับข้อมูลนักเรียนครบตามจำนวนที่ระบุให้ขึ้นคำถาม Exit? : ตามตัวอย่าง หากผู้ใช้กรอก Y ให้ทำการวนลูปซ้ำอีกครั้ง
- หากผู้ใช้ Exit ออกจากลูปให้แสดงผลตามตัวอย่างด้านล่าง

```
Number of Student? : 2
----------------------
Student Id : STU1
Score : 20
Student Id STU1 score is 20

Student Id : STU2
Score : 19
Student Id STU2 score is 19

Exit? : N

Number of Student? : 2
----------------------
Student Id : STU3
Score : 25
Student Id STU3 score is 25

Student Id : STU1
Score : 22
Update Score Student STU1 from 20 to 22

Exit? : Y

Result :
----------------------
All Student : 3
Max Score Student is STU3 : 25
Min Score Student is STU2 : 19
```

C# Class

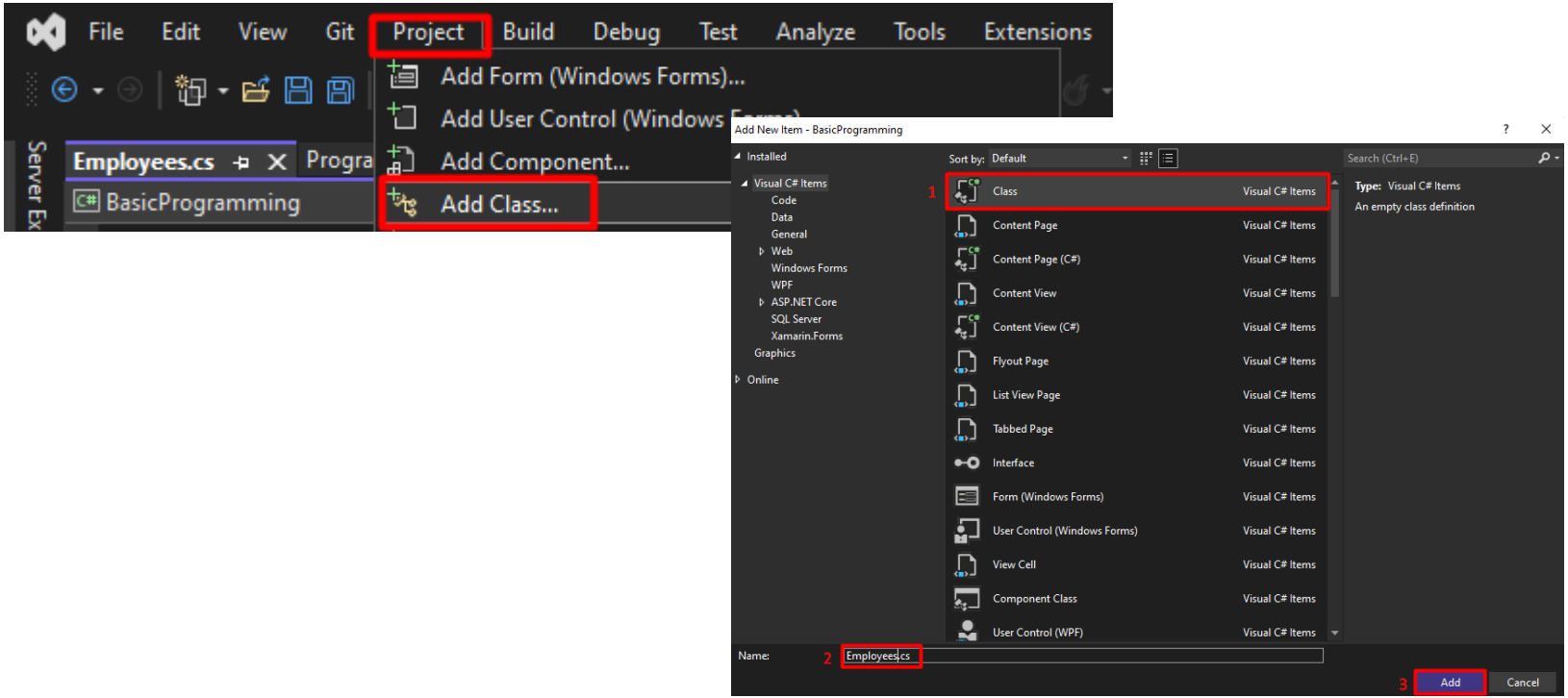
Class

- Class เป็นเหมือนต้นแบบ ที่ใช้ในการสร้าง Object โดยจะมีการกำหนด Attributes และ Method เพื่อใช้เป็นโครงสร้างพื้นฐานของ Object เราต้องการจะสร้าง



C# Class

Create Class



C# Class

ส่วนประกอบภายใน Class

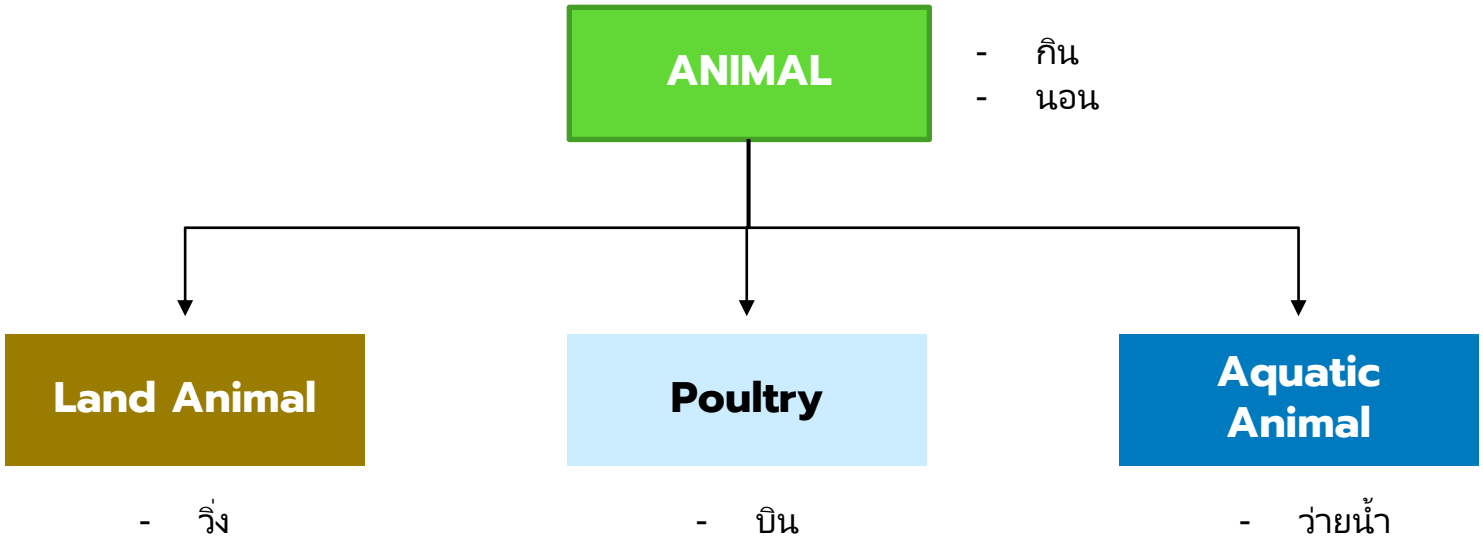
C# Class

Create Object

```
0 references
public class Program
{
    0 references
    public static void Main()
    {
        // Create Object
        person P1 = new person();

        // Set Property
        P1.Name = "Peach";
        P1.Birthday = new DateTime(2000, 4, 2);

        // Call Method
        P1.PrintInfo();
        P1.SayHello("Book");
    }
}
```

C# Class

Class inheritance



C# Method

Method

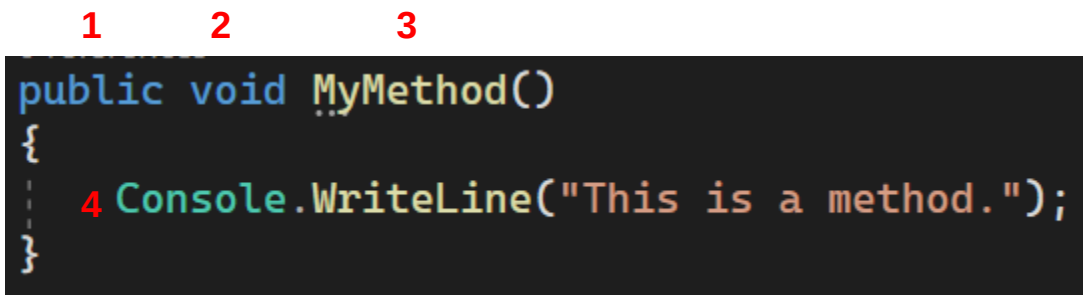
- Method เป็นกลุ่มของ code ที่ใช้เพื่อทำงานบางอย่าง มักถูกเขียนเพื่อใช้งานซ้ำๆ ช่วยลดการเขียน code ซ้ำซ้อนและทำให้ code ของเรามีความยืดหยุ่น

```
public void MyMethod()
{
    Console.WriteLine("This is a method.");
}
```

C# Method

ในการสร้าง Method จะต้องประกอบไปด้วย

1. Access Modifier ใช้เพื่อระบุระดับการเข้าถึงของ Method หรือตัวแปรนั้นๆ
2. Return Type ใช้เพื่อระบุประเภทของตัวแปรที่จะ Return ออกมา
3. ชื่อของ Method
4. ชุด code ที่เราต้องการให้ทำงาน



C# Method

Method Parameters

```
1 reference
static void ShowName(string name)
{
    Console.WriteLine("Hello My Name is {0}", name);
}

string Name = "Peach";
ShowName(Name);
```

C# Method

Instance Method

- Instance Method เป็นการสร้าง Method ที่ต้องมีการเรียกใช้ผ่าน Instance (Object ที่สร้างจาก Class)

Static Method

- Static Method เป็นการสร้าง Method ที่สามารถเรียกใช้ผ่าน Class ได้เลยโดยตรงไม่จำเป็นต้องมีการสร้าง Object เหมือน Instance Method

C# Method

```
// Instance Method
1 reference
public void SayHello(string yourName)
{
    Console.WriteLine($"Hello {yourName} nice to meet you, I'm {name}");
}
// Static Method
1 reference
public static void SayGoodbye(string yourName)
{
    Console.WriteLine($"Goodbye {yourName}");
}
```

C# Method

```
0 references
public class Program
{
    0 references
    public static void Main()
    {
        Person person1 = new Person();
        person1.Name = "Peach";
        person1.Birthday = new DateTime(2000, 4, 2);

        string Name = "Somchai";
        // Call Instance Method
        person1.SayHello(Name);

        // Call Static Method
        Person.SayGoodbye(Name);
    }
}
```

C# Method

Method Overloading

- การสร้าง Method ที่มีชื่อเดียวกัน แต่ต้องมีการรับ Parameter ที่ต่างกัน

```
2 references
public class Calculator
{
    1 reference
    public int Add(int x , int y)
    {
        return x + y;
    }
    1 reference
    public double Add(double x, double y)
    {
        return x + y;
    }
    1 reference
    public double Add(double x, double y, double z)
    {
        return x + y + z;
    }
}
```

C# Method

Constructor Method

- การสร้าง Method ที่มีชื่อเดียวกันกับ Class ใช้เพื่อกำหนด properties ในตอนที่สร้าง Class

```
1 reference
public Person()
{
    this.name = "John Wick";
    this.birthday = new DateTime(1964,9,2);
}
1 reference
public Person(string Name, DateTime BirthDay){
    this.name = Name;
    this.birthday = BirthDay;
}
```

```
Person John = new Person();
Person person1 = new Person("Peach",new DateTime(2000,4,2));
```

C# Method

Method Override

- คือการสร้าง Method ใน Class ลูกที่มีชื่อเดียวกันกับ Class แม่ ซึ่งจะทำให้ Class ลูกสามารถเรียกใช้ Method ชื่อเดียวกันแต่การทำงานภายในแตกต่างออกไปได้

```
public class Animal
{
    private string sex;
    private double normalSpeed;
    2 references
    public string Sex { get { return sex; } set { sex = value; } }
    3 references
    public double NormalSpeed { get { return normalSpeed; } set { normalSpeed = value; } }

    3 references
    public virtual double Run()
    {
        return normalSpeed * 2.5;
    }
}
```

C# Method

```
3 references
public class Cheetah : Animal
{
    1 reference
    public Cheetah(string sex,double normalSpeed)
    {
        Sex = sex;
        NormalSpeed = normalSpeed;
    }
    3 references
    public override double Run()
    {
        return NormalSpeed * 4.5;
    }
}

3 references
public class Tortoise : Animal
{
    1 reference
    public Tortoise(string sex, double normalSpeed)
    {
        Sex = sex;
        NormalSpeed = normalSpeed;
    }
}
```

```
0 references
public class Program
{
    0 references
    public static void Main()
    {
        Cheetah cheetah = new Cheetah("M",79.5);
        Tortoise tortoise = new Tortoise("F",7.2);

        double cheetahRun = cheetah.Run();
        double tortoiseRun = tortoise.Run();

        Console.WriteLine($"Cheetah run speed = {cheetahRun}");
        Console.WriteLine($"Tortoise run speed = {tortoiseRun}");

    }
}
```

```
Cheetah run speed = 357.75
Tortoise run speed = 18
```

Lab 04 – Parent and Child Class(25 mins)

1. ให้ทำการสร้าง Parent Class ขึ้นมา 1 Class โดยจะต้องประกอบไปด้วย Attribute และ Method อะไรก็ได้
2. ให้สร้าง Child Class ที่สืบทอดคุณสมบัติมาจาก Parent Class ก่อนหน้าโดยใน Child Class จะต้องมี Attribute และ Method อื่นๆของตัวเอง
3. ให้สร้าง Constructor Method ของ Child Class 2 Method
4. ใน Class Program Method Main ให้ทำการสร้าง Object จาก Child Class และเรียกใช้ Method ที่สร้าง