Chapter 5

Object and Class

- **The features of an object are described by Properties.**
- **The behaviors of an object are described by Methods .**
- **Objects refer to data with properties and methods**

1. What is class?

- Class is the collection of objects of the same attributes.
- Class itself is not a real object.

  ⇨ Class is used to define the structure of an object, and describe the properties and methods of these similar objects.

1.2 Access restrictions for class members

- There are three common modifiers for accessing class members.

1. public

Access to public members is not restricted. Public members are allowed to be used in class, subclass, or declared objects. They are at public level.

2. private

Private members can be only accessed within their own classes. They are at private level and can not be used by the outside world.

3. protected

Protected members can not only be accessed by their base classes, but also by their sub classes. They are at protection level.

1.3 Create objects and classes

How to create a class

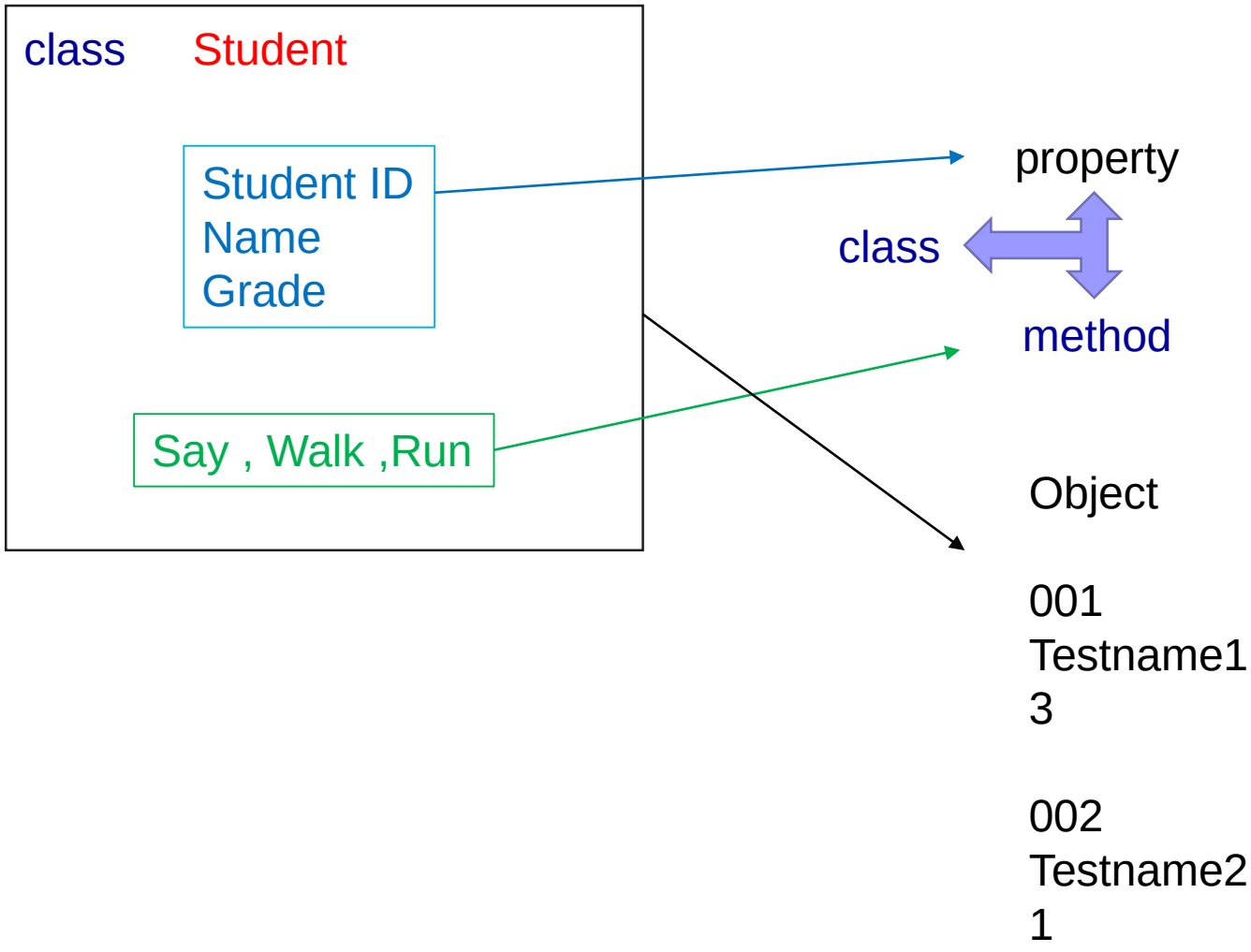
- C# uses class{...} to define a class
- The definition of classes and events:
  - ⇨ it’s cannot be placed in a method.
  - ⇨ it’s cannot be placed outside the namespace {...}.

3 How to create properties

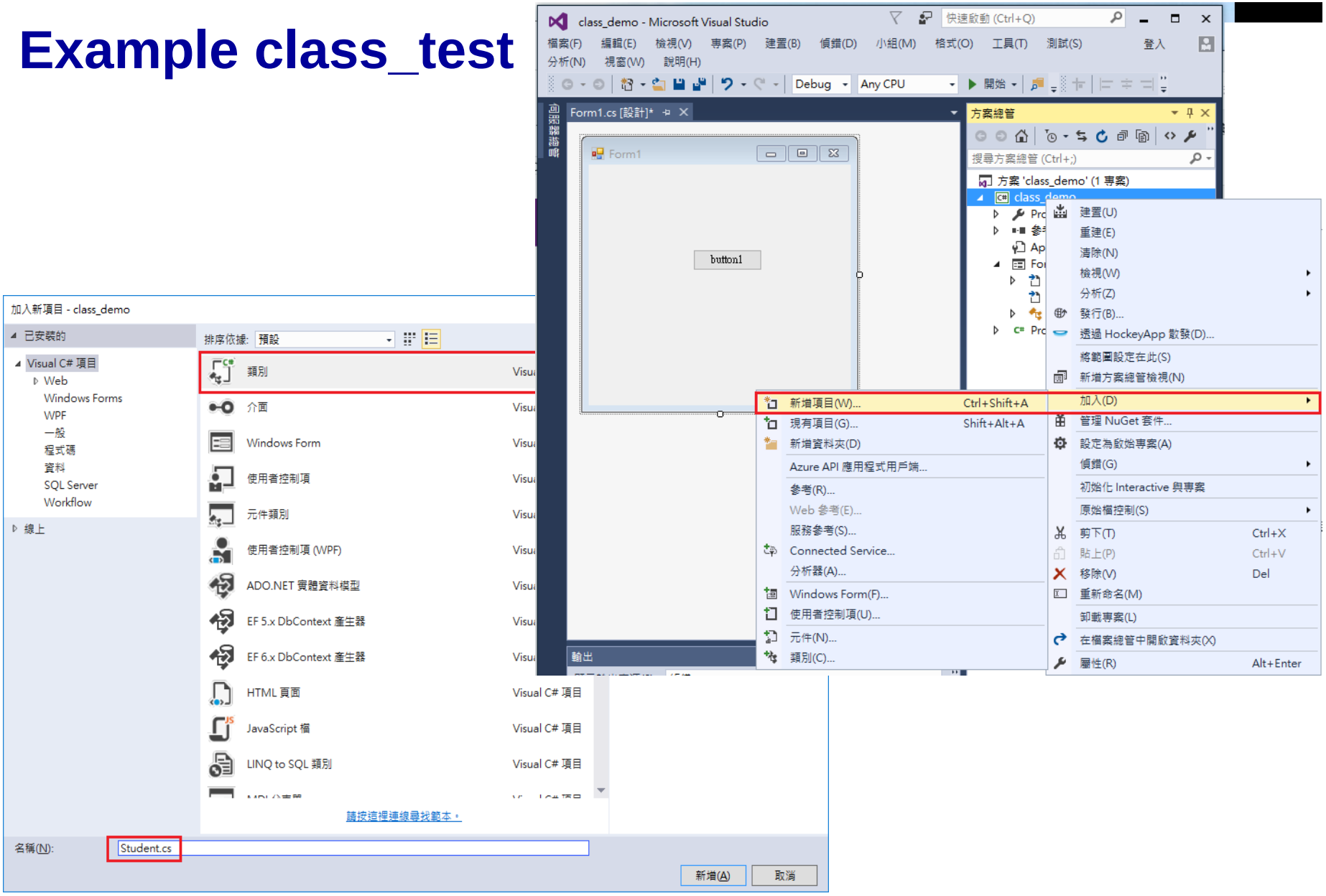
There are two ways to create properties:

1. **Directly declare public variables in class**
2. **Use get and set accessors**

Example class_test



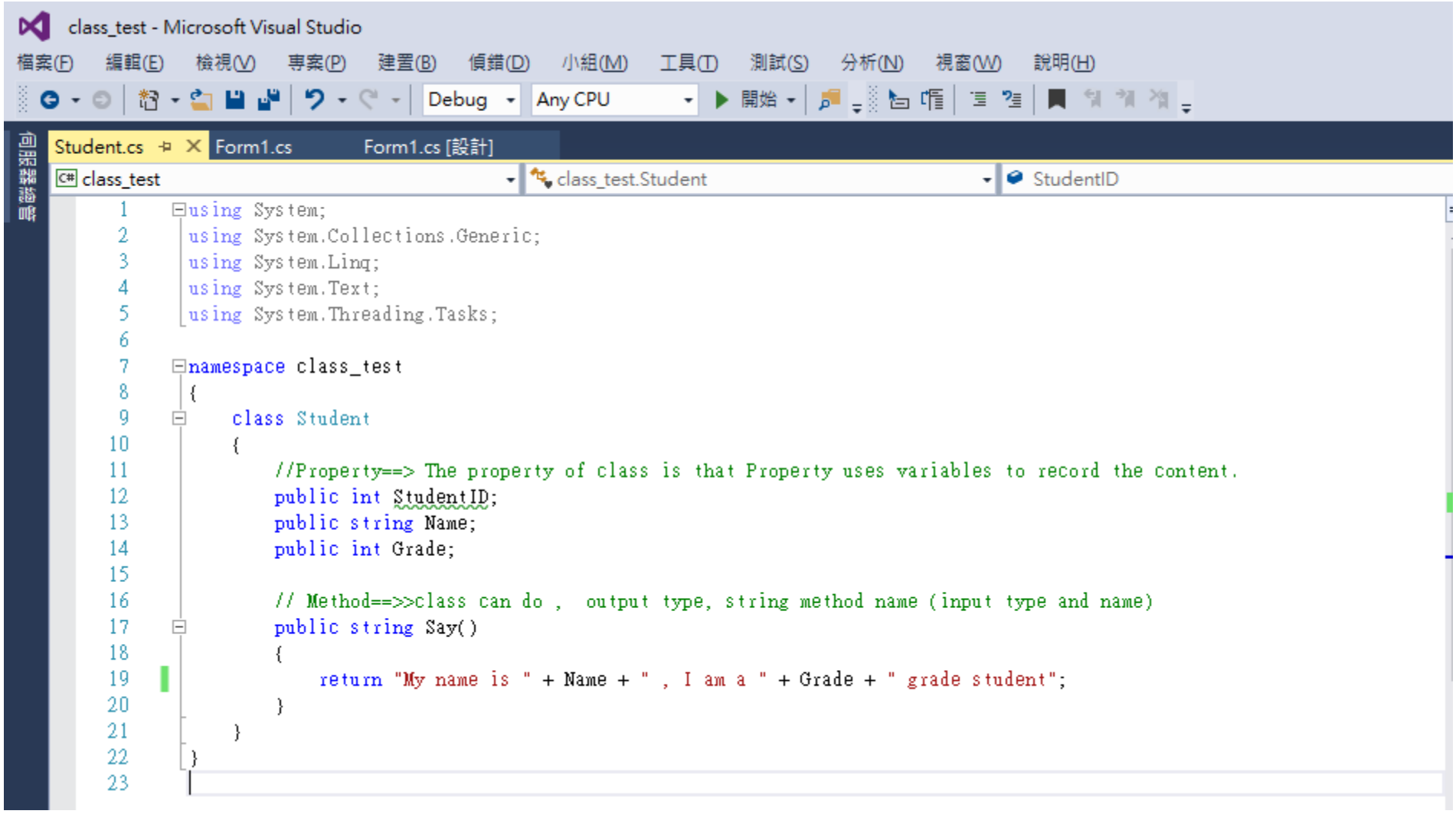
Example class_test



How to use public variables to create object properties

Example ConsoleProperty1:

```
01 namespace ConsoleProperty1
02 {
03     class Car
04     {
05         public int Speed;          // 宣告 Speed 為 public 公用變數
06     }
07     class Program
08     {
09         static void Main(string[] args)
10         {
11             Car Benz = new Car();
12             Benz.Speed = 100;      // 物件建立之後可直接使用「.」存取該屬性
13             Console.WriteLine("Benz.Speed = {0} ",Benz.Speed);
14             Console.Read();
15         }
16     }
17 }
```


class_test - Microsoft Visual Studio
檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 工具(T) 測試(S) 分析(N) 視窗(W) 說明(H)
Debug
Any CPU
開始
Student.cs
Form1.cs
Form1.cs [設計]
class_test
class_test.Student
StudentID
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace class_test
{
class Student
{
//Property==> The property of class is that Property uses variables to record the content.
public int StudentID;
public string Name;
public int Grade;
// Method==>>class can do , output type, string method name (input type and name)
public string Say()
{
return "My name is " + Name + " , I am a " + Grade + " grade student";
}
}
}

2. Constructor functions

- When declare and create an object
  ⇨ use constructor to initialize object
- The method which has the same name as class is called constructor
- Constructor
  ⇨ automatically run in background method when creating object
- ⇨ initialize members in class

Use constructor

- Use new to create object of class
  ⇨ constructor is run immediately
- Constructor has no return value
  ⇨ no return statement and no void declaration
- Every class has a constructor without any parameter and statement, it is default constructor
- If no constructor is defined, default constructor is executed
  ⇨ use default value to initialize every field
  ⇨ initialize integer to 0, float to 0.0
  ⇨ reference is initialized to null

Form1.cs | Form1.cs [設計] | Student.cs

Constructor_test | Constructor_test.Form1

```
7   using System.Text;
8   using System.Threading.Tasks;
9   using System.Windows.Forms;
10
11  namespace Constructor_test
12  {
13      public partial class Form1 : Form
14      {
15          public Form1()
16          {
17              InitializeComponent();
18          }
19
20          private void button1_Click(object sender, EventArgs e)
21          {
22              Student s1 = new Student(); // Question 1. Write a
23
24              s1.Name = "testname1";
25              s1.StudentID = 10602001;
26
27              Student s2 = new Student();
28              s1.Name = "testname2";
29              s1.StudentID = 10602002;
30
31              MessageBox.Show(s2.Say());
32          }
33      }
34  }
```

[Reminder]
Create object: new class name();

orm1.cs | Form1.cs [設計] | Student.cs

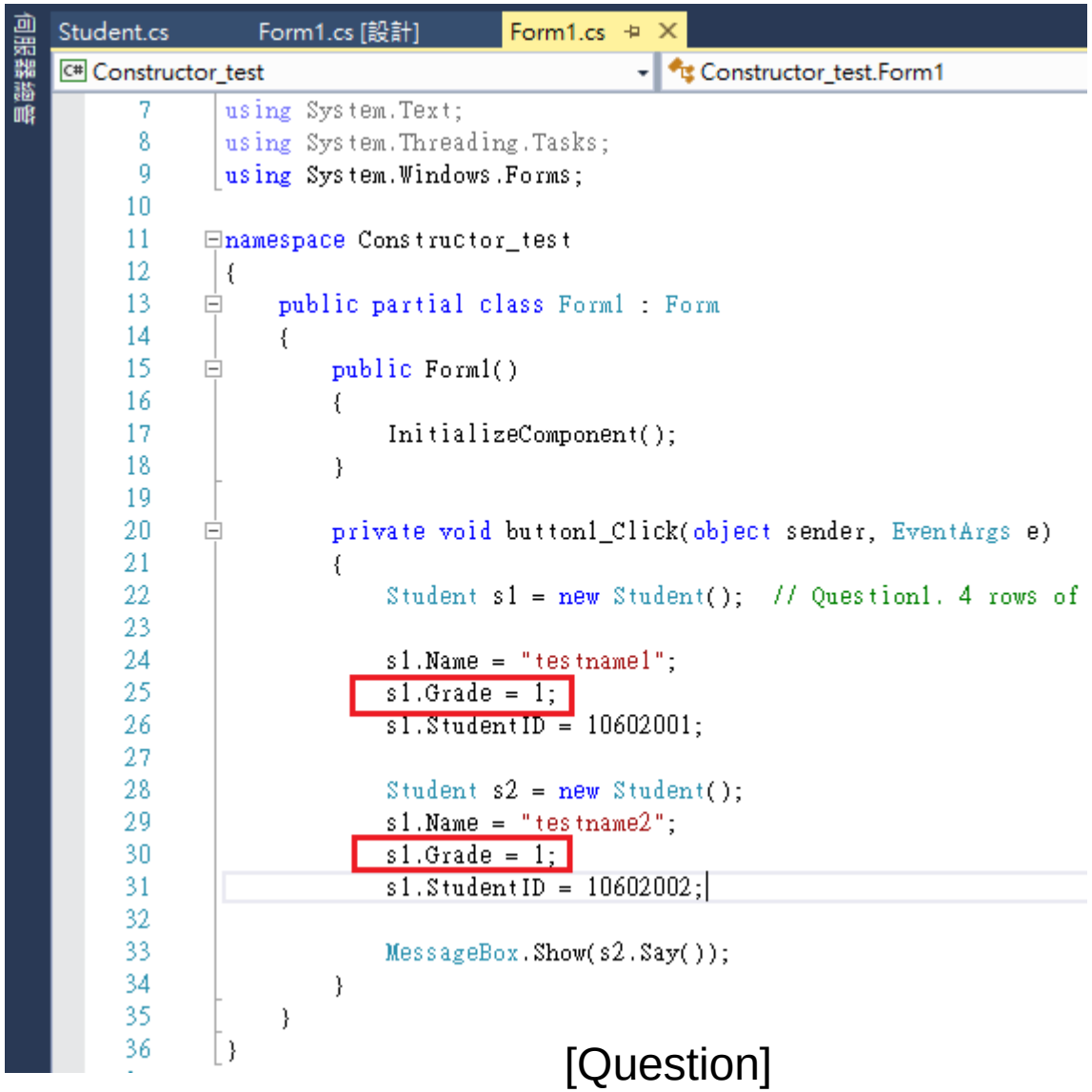
Constructor_test | Constructor_test.Student | Grade

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Windows.Forms; //使用 messagebox
6   using System.Threading.Tasks;
7
8   namespace Constructor_test
9   {
10      class Student
11      {
12          //Property==> The property of class is that Property uses variables to record the
13          public int StudentID;
14          public string Name;
15          public int Grade;
16
17          public Student() // does not output any type, same name as the class
18          {
19              MessageBox.Show("This is a test!!");
20          }
21
22          public string Say()
23          {
24              return "My name is " + Name +
25          }
26
27          public string Talk(Student s) //
```

Create a Constructor

1. One kind of Method
2. Implement when creating objects

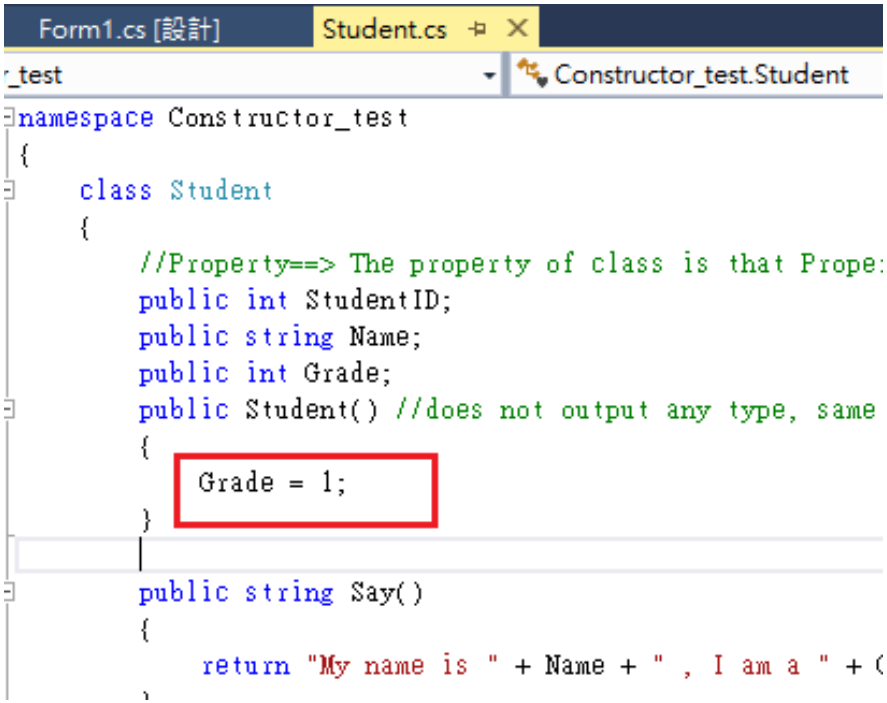
```
namespace Constructor_test
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            new Student();
        }
    }
}
```



[Question]

1.Four rows of code are needed when creating one object

2. Repeat setting the same value



If the default of attribute is known at the beginning, it can be applied to the sub constructor to set objects beforehand

The End