

Design pattern

هو طريقة لكتابه الكود بطريقة سهلة ولللى يطلع على مشاريعك يقدر يفهمها

PATTERNS COVERED

- Singleton Pattern
- Factory Pattern
- Template Method Pattern
- Adapter Pattern
- Façade Pattern
- Strategy Pattern
- State Pattern
- Proxy Pattern
- Chain of Responsibility Pattern
- Bridge Pattern
- Composite Pattern
- Decorator Pattern
- Observer Pattern
- Builder Pattern
- Flyweight Pattern
- Repository Pattern

PatrickVideos.com

BISHOU NABIL
هنتعرف على كل نوع

بيقولك الكورس ده هيأخذك من

Junior

الى

Senior

Singleton Pattern

What is a Singleton Pattern?

If we want a class to have only one object, then we use the singleton pattern.

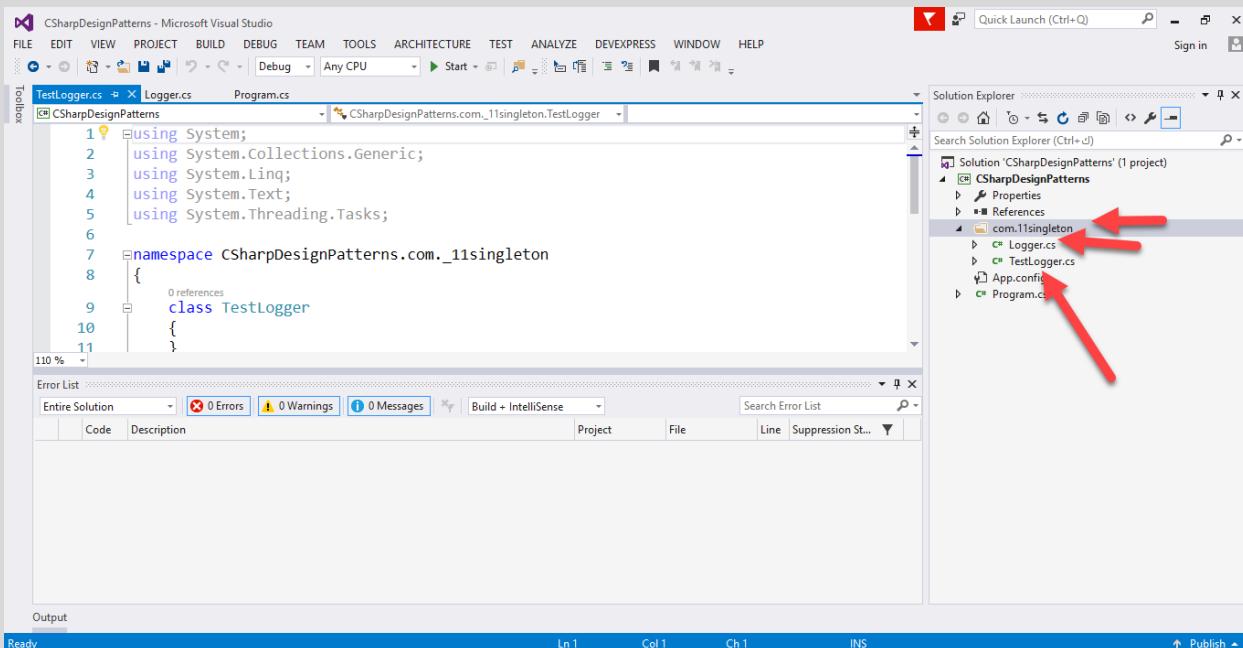
Why do we need only one object of a class?

Objects used for logging, device drivers for printers or registry setting need to have only one of a kind. If there are multiple objects we may encounter problems like incorrect program behavior, inconsistent results etc.

PatrickVideos.com

بيقولك لو عايز الكلاس بتعمل يكون اوبجيكت واحد فقط هنسخدم ال

Singleton
BISHOYNABIL



بنعمل 2 كلاس وفولدر

Screenshot of Microsoft Visual Studio showing the `Logger` class definition in the `Logger.cs` file. The code defines a `Logger` class with a private constructor. A red arrow points to the constructor line.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharpDesignPatterns.com._1singleton
8  {
9      class Logger
10     {
11         private Logger()
12         {
13         }
14     }
15 }
16
17 }
```

Screenshot of Microsoft Visual Studio showing the `TestLogger` class definition in the `TestLogger.cs` file. The code creates two instances of the `Logger` class. A red arrow points to the line where the first instance is created.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharpDesignPatterns.com._1singleton
8  {
9      class TestLogger
10     {
11         Logger obj1 = new Logger();
12         Logger obj2 = new Logger();
13     }
14 }
15
16 }
```

تعالى نرجع لкласс

Logger

CSharpDesignPatterns - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

TestLogger.cs Logger.cs* Program.cs CSharpDesignPatterns.com._11singleton.Logger Logger()

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharpDesignPatterns.com._11singleton
8  {
9      class Logger
10     {
11         private Logger()
12         {
13         }
14     }
15 }
16
17 }
18
```

110 % Output Error List

Ready Ln 11 Col 16 Ch 16 INS

Solution Explorer

- Solution 'CSharpDesignPatterns' (1 project)
 - CSharpDesignPatterns
 - Properties
 - References
 - com.11singleton
 - Logger.cs
 - TestLogger.cs
 - App.config
 - Program.cs

CSharpDesignPatterns - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

TestLogger.cs Logger.cs* Program.cs CSharpDesignPatterns.com._11singleton.TestLogger obj2

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharpDesignPatterns.com._11singleton
8  {
9      class TestLogger
10     {
11         Logger obj1 = new Logger();
12         Logger obj2 = new Logger();
13
14     }
15 }
16
```

133 % Output Error List

Item(s) Saved Ln 12 Col 20 Ch 20 INS

Solution Explorer

- Solution 'CSharpDesignPatterns' (1 project)
 - CSharpDesignPatterns
 - Properties
 - References
 - com.11singleton
 - Logger.cs
 - TestLogger.cs
 - App.config
 - Program.cs

حصلت مشكلة بسبب انى غيرت الاوبجيكت من Public الى Private

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the project 'CSharpDesignPatterns' with its files: Properties, References, and a folder 'com.11singleton' containing 'Logger.cs', 'TestLogger.cs', 'App.config', and 'Program.cs'. 'Logger.cs' is currently selected.
- Code Editor:** Displays the 'Logger.cs' file content:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;

6
7  namespace CSharpDesignPatterns.com._11singleton
8  {
9      class Logger
10     {
11         private Logger logger; // Red arrow points here
12         private Logger()
13         {
14
15         }
16
17     }
18 }
```
- Status Bar:** Shows 'Item(s) Saved' at the bottom left, and 'Ln 11 Col 31 Ch 31 INS' at the bottom center.

بنعمل نسخه من الاوبجيكت بتاعنا

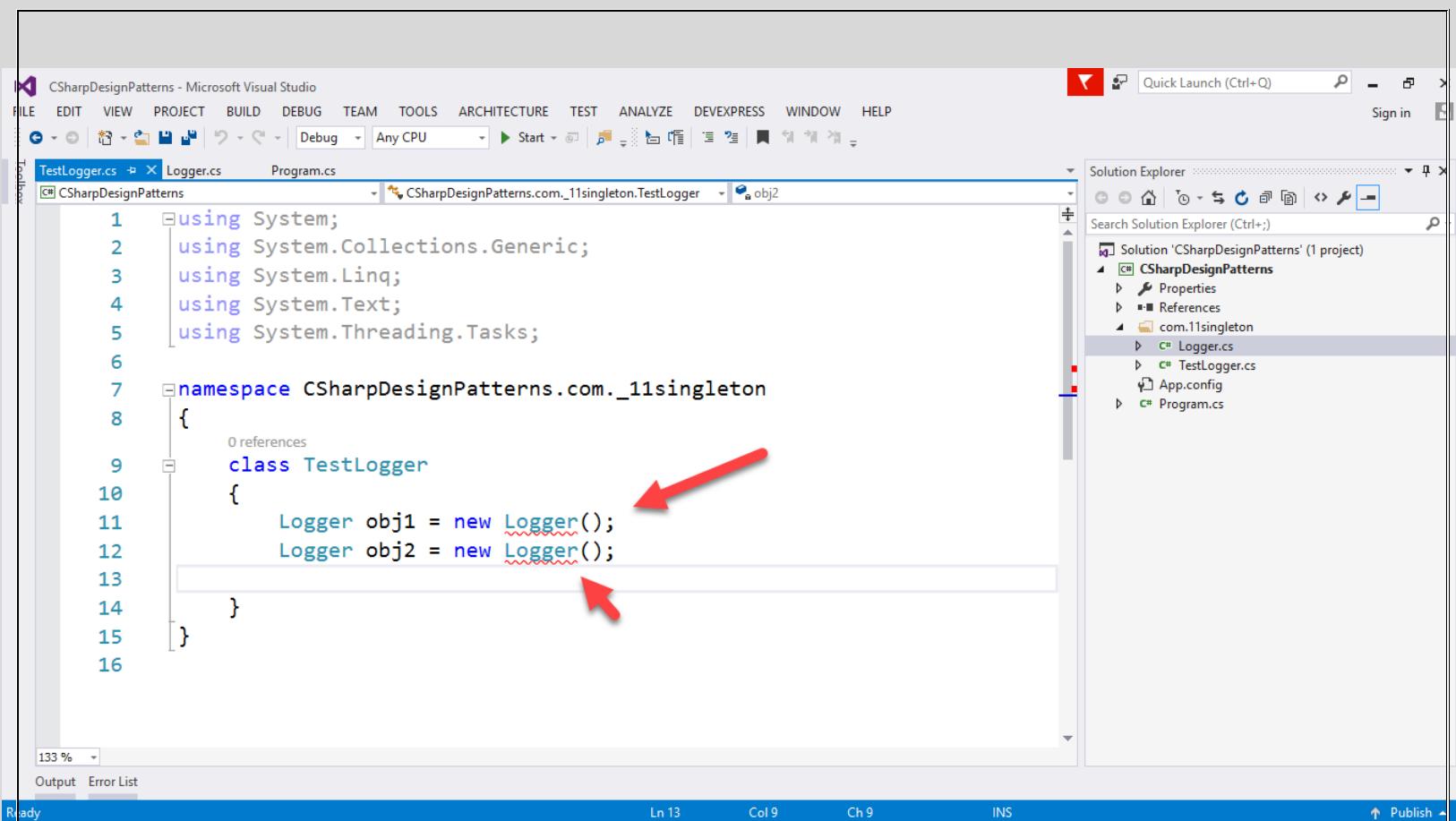
```
10  {
11      private Logger logger; ←
12
13      private Logger()
14      {
15
16      }
17
18      public Logger GetInstance()
19      {
20          if (logger == null)
21          {
22              logger = new Logger();
23          }
24          return logger; ←
25
26      }
27 }
```

المثال سهل وبسيط
هنا يحصل ايه بالضبط

GetInstance
بتشيك الاوبيجكت
LOGGER
موجود ولا وهل

فاضى ولا مش فاضى لو فاضى هتروح تعمل نسخه جديدة من كلاس
Logger

طب لو مش فاضى هيرجعلى
Logger
انا اتلغبط
:D



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharpDesignPatterns.com._11singleton
8  {
9      class TestLogger
10     {
11         Logger obj1 = new Logger();
12         Logger obj2 = new Logger();
13     }
14 }
```

اها هنا عايز انادى على الميثود بتاعته من الكلاس بس مش عارف لازم اخد نسخه منه
طب اعمل ايه طه

تعالى نشوف طريقة اخرى ننادى بيه على الميثود

فيه طريقة ننادى بيه على الميثود بدون ما اخد نسخه من الكلاس ازاي عن طريق

Static

بص

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** CSharpDesignPatterns - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEXPRESS WINDOW HELP
- Toolbars:** Standard, Debug, Start, Task List, Solution Explorer, Properties, Task List, Status Bar.
- Code Editor:** TestLogger.cs* (C# file) is open. The code implements a Singleton pattern with a private constructor and a static GetInstance() method. A red arrow points to the static method definition.

```
10  {
11      private Logger logger;
12      private Logger()
13      {
14      }
15      0 references
16      public static Logger GetInstance()
17      {
18          if (logger == null)
19          {
20              logger = new Logger();
21          }
22          return logger;
23      }
24  }
25
26
27 }
```

- Solution Explorer:** Shows the project structure: CSharpDesignPatterns (1 project) containing com.1singleton (Logger.cs, TestLogger.cs, App.config, Program.cs).

BISHOY NABIL
كتينا

Static

وال

Static

لها رولز

بتقول

انها تقدر هيا بس تنادى على ميثود اخرى نوعها

Static

Or

Data

CSharpDesignPatterns - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

TestLogger.cs* Logger.cs* Program.cs

Solution Explorer

Search Solution Explorer (Ctrl+.)

Solution 'CSharpDesignPatterns' (1 project)

- CSharpDesignPatterns
- Properties
- References
- com.11singleton
 - Logger.cs
 - TestLogger.cs
 - App.config
 - Program.cs

8 {
9 8 references
10 class Logger
11 {
12 private static Logger logger;
13 3 references
14 private Logger()
15 {
16 }
17 0 references
18 public static Logger GetInstance()
19 {
20 if (logger == null)
21 {
22 logger = new Logger();
23 }
24 return logger;
}

133 %

Output Error List

Ready Ln 11 Col 23 Ch 23 INS

BISHOY NABIL



CSharpDesignPatterns - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

TestLogger.cs* Logger.cs* Program.cs

Solution Explorer

Search Solution Explorer (Ctrl+.)

Solution 'CSharpDesignPatterns' (1 project)

- CSharpDesignPatterns
- Properties
- References
- com.11singleton
 - Logger.cs
 - TestLogger.cs
 - App.config
 - Program.cs

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace CSharpDesignPatterns.com._11singleton
8 {
9 0 references
10 class TestLogger
11 {
12 //Logger obj1 = new Logger();
13 //Logger obj2 = new Logger();
14 Logger obj1 = Logger.GetInstance();
15 Logger obj2 = Logger.GetInstance();
16 }
17 }
18

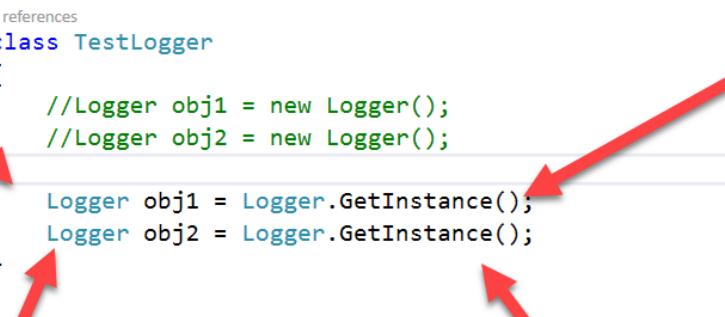
133 %

Output Error List...

Item(s) Saved

Ln 13 Col 9 Ch 9 INS

BISHOY NABIL



شوف قدرت استخدم الميثود بدون ما اخد نسخه جديدة من الكلاس

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** CSharpDesignPatterns - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
- Toolbar:** Standard toolbar with icons for file operations.
- Solution Explorer:** Shows a single project 'CSharpDesignPatterns' with a sub-project 'com.11singleton' containing files: Logger.cs, TestLogger.cs, App.config, and Program.cs.
- Toolbox:** Standard .NET toolbox with various controls.
- Code Editor:** Displays the 'TestLogger.cs' file with the following code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharpDesignPatterns.com._11singleton
8  {
9      class TestLogger
10     {
11         //Logger obj1 = new Logger();
12         //Logger obj2 = new Logger();
13         static void Main()
14         {
15             Logger obj1 = Logger.GetInstance();
16             Logger obj2 = Logger.GetInstance();
17
18             Console.WriteLine(obj1.GetHashCode());
19             Console.WriteLine(obj2.GetHashCode());
20         }
21     }
22 }
```

Annotations in the code editor:

- A green arrow points to the first two commented-out lines: `//Logger obj1 = new Logger();` and `//Logger obj2 = new Logger();`.
- Three red arrows point to the two calls to `GetInstance()`: `Logger obj1 = Logger.GetInstance();` and `Logger obj2 = Logger.GetInstance();`.

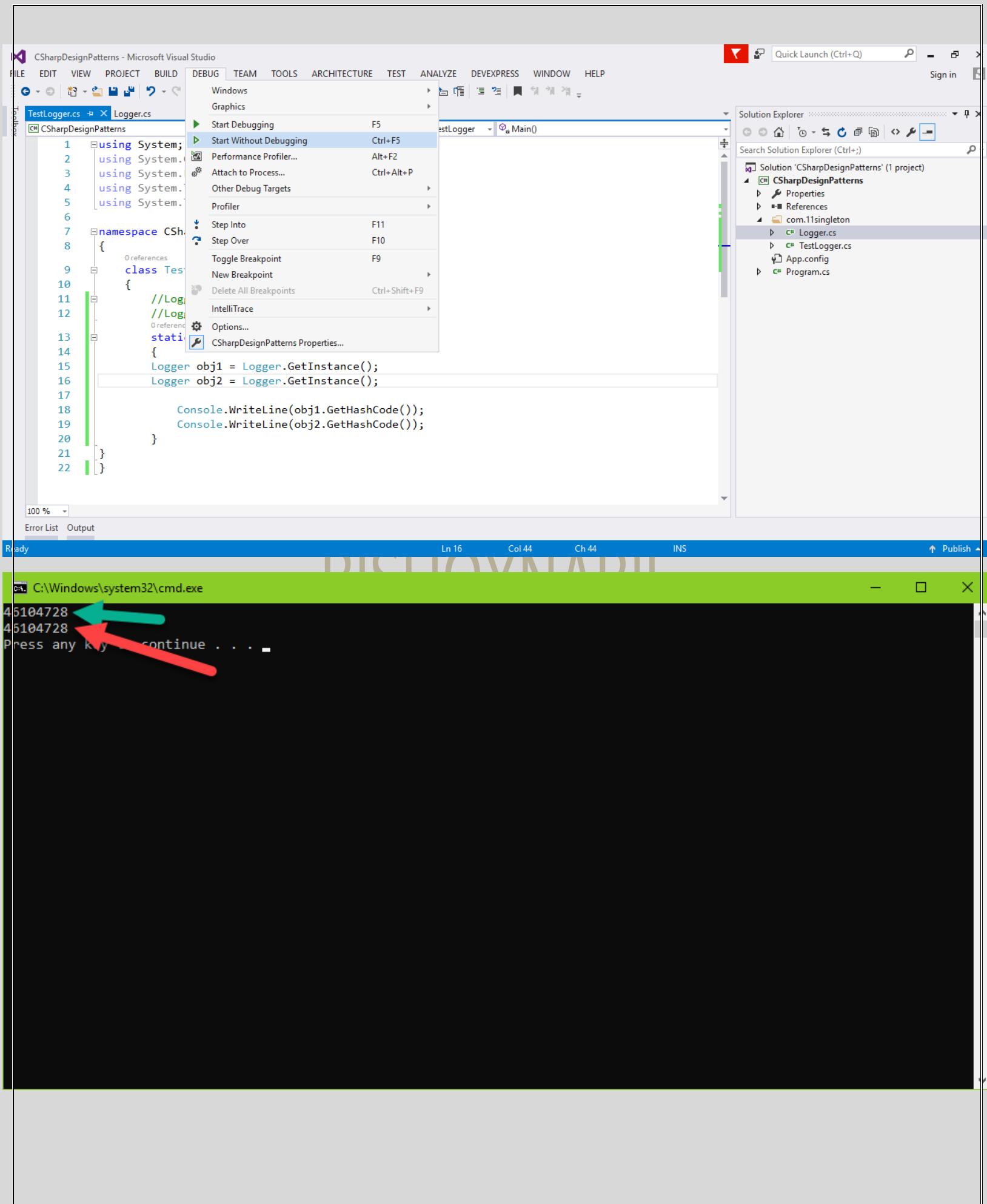
Bottom status bar: Ready, Ln 16, Col 44, Ch 44, INS.

هنا ضيف ال

Main

ووقفت

Main in program



The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** CSharpDesignPatterns - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
- Toolbars:** Standard, Debug, Start, Task List, Solution Explorer, Properties, Task List, Status Bar.
- Solution Explorer:** Shows the project 'CSharpDesignPatterns' with one solution and one project named 'com.11singleton'. Inside 'com.11singleton', there are files: Logger.cs, TestLogger.cs, App.config, and Program.cs.
- Code Editor:** The file 'TestLogger.cs' is open. The code demonstrates the Singleton pattern with a static factory method 'GetInstance()'. Red arrows highlight several parts of the code:
 - Line 15: `Console.WriteLine("-----Singleton---");`
 - Line 16: `Logger obj1 = Logger.GetInstance();`
 - Line 17: `Logger obj2 = Logger.GetInstance();`
 - Line 22: `Console.WriteLine("-----Non Singleton---");`
 - Line 23: `Test obj3 = new Test();`
 - Line 24: `Test obj4 = new Test();`
 - Line 26: `Console.WriteLine(obj3.GetHashCode());`
 - Line 27: `Console.WriteLine(obj4.GetHashCode());`
- Status Bar:** Ready, Ln 13, Col 27, Ch 27, OVR, Publish

تعالی نشووف الفرق بین ال

Singleton

واي کلاس اخر بص

ضیفت کلاس جدید و خدت نسخه منه و نجرب

```
C:\Windows\system32\cmd.exe
----Singleton----
45104728 ←
45104728 ←
----Non Singleton----
12289376 ←
43495525 ←
Press any key to continue . . .
```

شوفت الفرق في الهاش كود ازاي

بيقولك تعريف ال

Singleton

هو لو عندك

نسخه واحدة من اوبجيكت الكلاس وانتا عايز تعيid استخدام مره ومره كذا مره يعني
ساعتها هتسخدم

Singleton

Singleton Pattern

What is a Singleton Pattern?

If we want a class to have only one object, then we use the singleton pattern.

Why do we need only one object of a class?

Objects used for logging, device drivers for printers or registry setting need to have only one of a kind. If there are multiple objects we may encounter problems like incorrect program behavior, inconsistent results etc.

PatrickVideos.com

بيقولك ليه استخدم الـ

Singleton

ووفى ايه بستخدمها زى عمليات تسجيل الدخول وتعريفات الجهاز للبرنتر وعمليات حفظ التسجيل

وبيكولك لو عملت اكتر من اوبيجيكت هتعمل مشاكل كتير جدا وهتخرب الدنيا

بيقولك فيه سينتكس اخر لكتابه الكود تعالى نشوفه

غيرت الاسم وعملت ميثود جديدة

```
8     {
9         8 references
10        class Logger
11        {
12            private static Logger instance;
13            1 reference
14            private Logger()
15            {
16            }
17
18            // public static Logger GetInstance() ...
19
20            0 references
21            public static Logger Instance
22            {
23                get{
24                    if (instance == null)
25                    {
26                        instance = new Logger();
27                    }
28                    return instance;
29                }
30            }
31        }
32    }
33
34
35
36
37 }
```

غيرت الاسم وعملت ميثود جديدة

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharpDesignPatterns.com._1singleton
8  {
9      0 references
10     class TestLogger
11     {
12         //Logger obj1 = new Logger();
13         //Logger obj2 = new Logger();
14         0 references
15         static void Main()
16         {
17             Console.WriteLine("----Singleton---");
18             Logger obj1 = Logger.Instance;
19             Logger obj2 = Logger.Instance; ←
20
21             Console.WriteLine(obj1.GetHashCode());
22             Console.WriteLine(obj2.GetHashCode());
23
24             Console.WriteLine("----Non Singleton---");
25             Test obj3 = new Test();
26             Test obj4 = new Test();
27         }
28     }
29
30
31
32
33
34
35
36
37 }
```

```
C:\Windows\system32\cmd.exe
----Singleton----
45104728
45104728
----Non Singleton----
12289376
43495525
Press any key to continue . . . ■
```

BISHOYNABIL

مفيش حديد شغاله عادي

بيقولك الطريقة دى او اللي فاتت الاتنين صح

CSharpDesignPatterns - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Program.cs

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace CSharpDesignPatterns.com._11singleton
8 {
9     class Logger
10    {
11        private static Logger instance;
12        private Logger()
13        {
14        }
15
16        // public static Logger GetInstance() ...
17
18        public static Logger Instance
19        {
20            get{
21                if (instance == null)
22                {
23                    instance = new Logger();
24                }
25            }
26        }
27    }
28
29
30

```

Logger.cs

Solution Explorer

- Solution 'CSharpDesignPatterns' (1 project)
 - CSharpDesignPatterns
 - Properties
 - References
 - bin
 - com.11singleton
 - Logger.cs

Context menu for Logger.cs:

- Open
- Open With...
- View Code F7
- View Class Diagram
- New Solution Explorer View
- Show on Code Map
- Exclude From Project**
- Cut Ctrl+X
- Copy Ctrl+C
- Delete Del
- Rename
- Properties Alt+Enter

Error List Output

Ready Ln 1 Col 1 Ch 1 INS Publish

بتحفي الكلاسات

CSharpDesignPatterns - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

A.cs

Program.cs

```

1 using System;
2
3 namespace CSharpDesignPatterns.com._12abasicconcepts
4 {
5     class A
6     {
7     }
8
9     class B : A
10    {
11    }
12
13     class Test
14    {
15    }
16
17     static void Main()
18    {
19        A aobj = new A();
20
21        B boobj = new B();
22
23        A obj1 = new B();
24
25        B obj2 = new A();
26
27    }
28
29
30
31

```

Solution Explorer

- Solution 'CSharpDesignPatterns' (1 project)
 - CSharpDesignPatterns
 - Properties
 - References
 - bin
 - com.11singleton
 - com.12abasicconcepts
 - A.cs
 - obj
 - App.config
 - Program.cs

Item(s) Saved Ln 29 Col 30 Ch 30 INS Publish

بص هنا الشرح بسيط

قدر

A

يأخذ من

B

لكن

B

فشل يأخذ بداخله نسخه من

A

BISHOYNABIL

هناك الكورس ونشوف كورس غيره

In this session we will learn

- What is Singleton Design Pattern
- Singleton as Creational Pattern
- Implementation Guidelines
- How do we Implement a Singleton class

Design Pattern Series

Part 1 – Design Patterns Tutorial

2

Facebook.com/PragimTech | Twitter.com/Kudvenkat | PragimTech.com

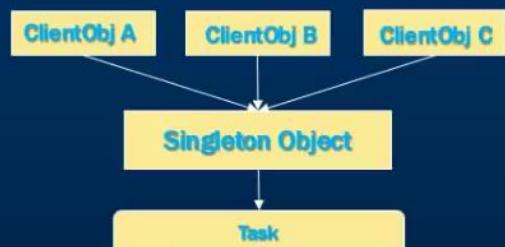
اللى هيتم شرحه هنا

What is Singleton Design Pattern

Singleton Pattern

Singleton Pattern belongs to Creational Type pattern.

This pattern is used when we need to ensure that only one object of a particular class need to be created. All further References to the objects are referred to the same underlying instance created.



3

Facebook.com/PragimTech | Twitter.com/Kudvenkat | PragimTech.com

يقولك نوع

Singleton

يقولك بنسخدمه لما نكون عايزين نعمل اوبجيكت واحد فقط

Singleton Advantages and Guidelines

Advantage of Singleton

- Singleton controls concurrent access to the resource.
- It ensures there is only one object available across the application in a controlled state.

Implementation Guidelines

- Ensure that only one instance of the class exists
- Provide global access to that instance by
 - Declaring all constructors of the class to be private
 - Providing static method that returns a reference to the instance
 - The instance is stored as a private static variable.

SingletonDemo - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze DevExpress Window Help

Singleton.cs Program.cs

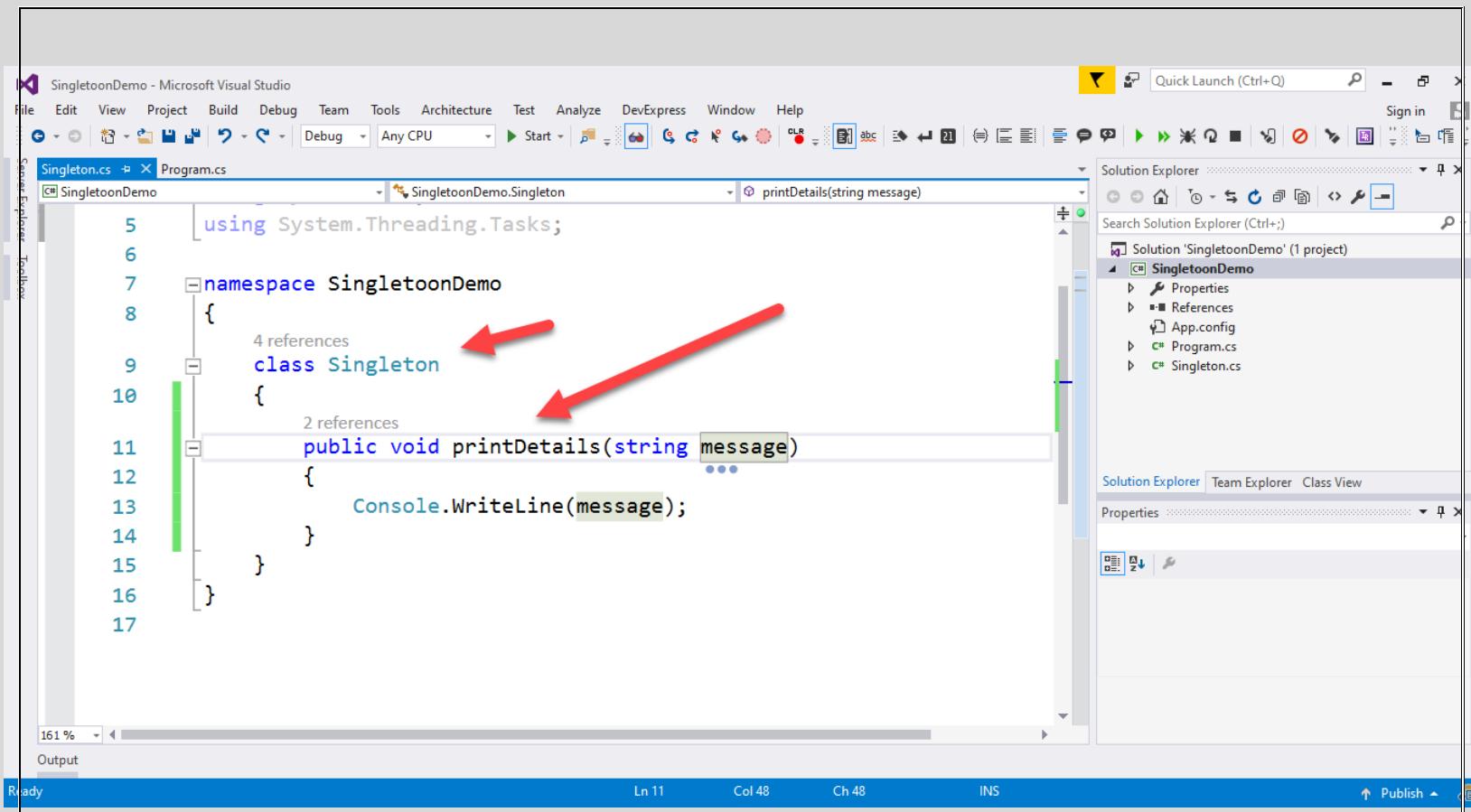
Solution Explorer Toolbox

```
5     using System.Threading.Tasks;
6
7     namespace SingletonDemo
8     {
9         class Singleton
10        {
11            public void printDetails(string message)
12            {
13                Console.WriteLine(message);
14            }
15        }
16    }
17
```

161% Output Ready Ln 11 Col 48 Ch 48 INS Publish

4 references
2 references

BISHOYNABIL بنعمل ميثود جوه کلاس



SingletonDemo - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze DevExpress Window Help

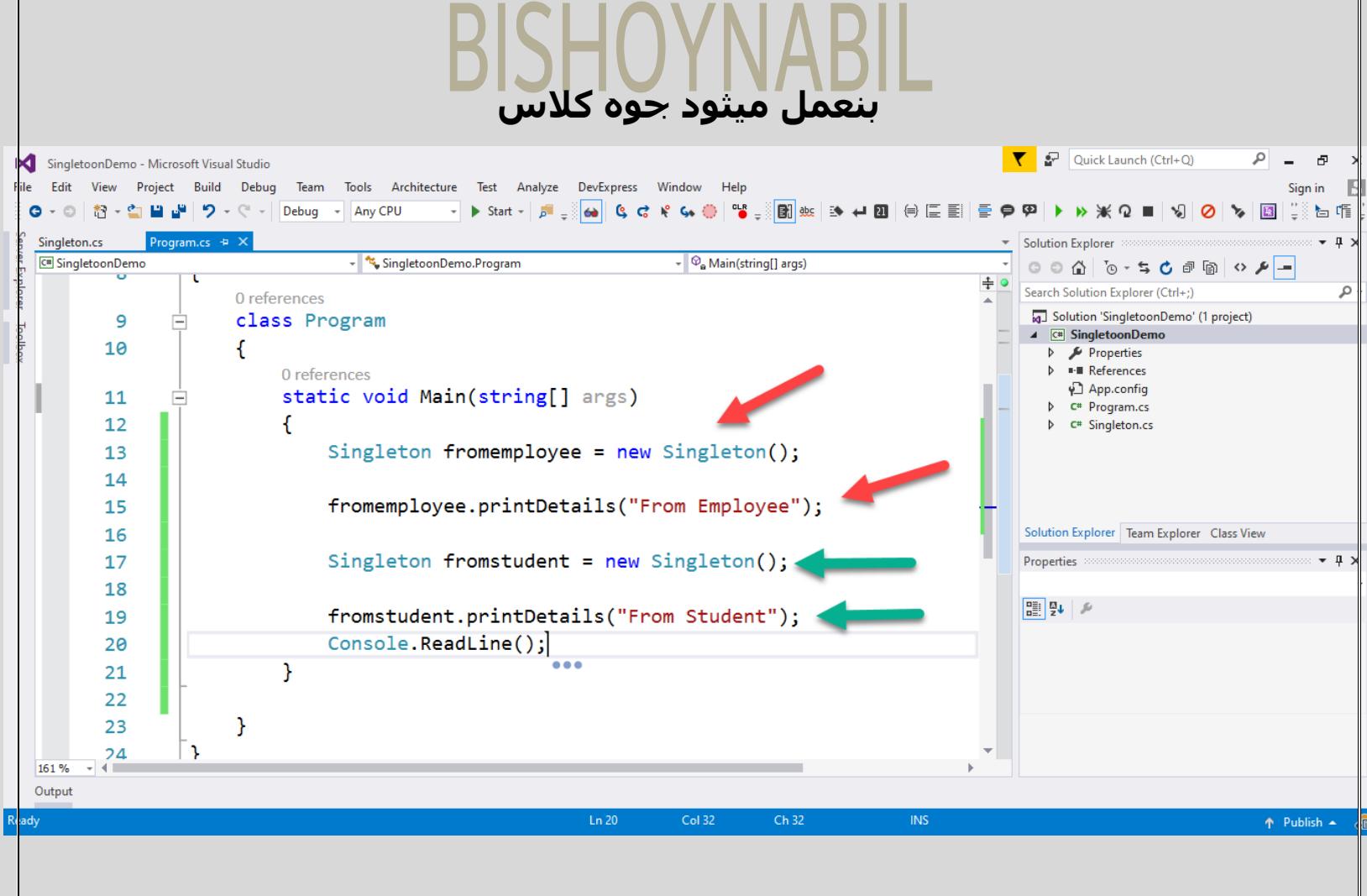
Singleton.cs Program.cs

Solution Explorer Toolbox

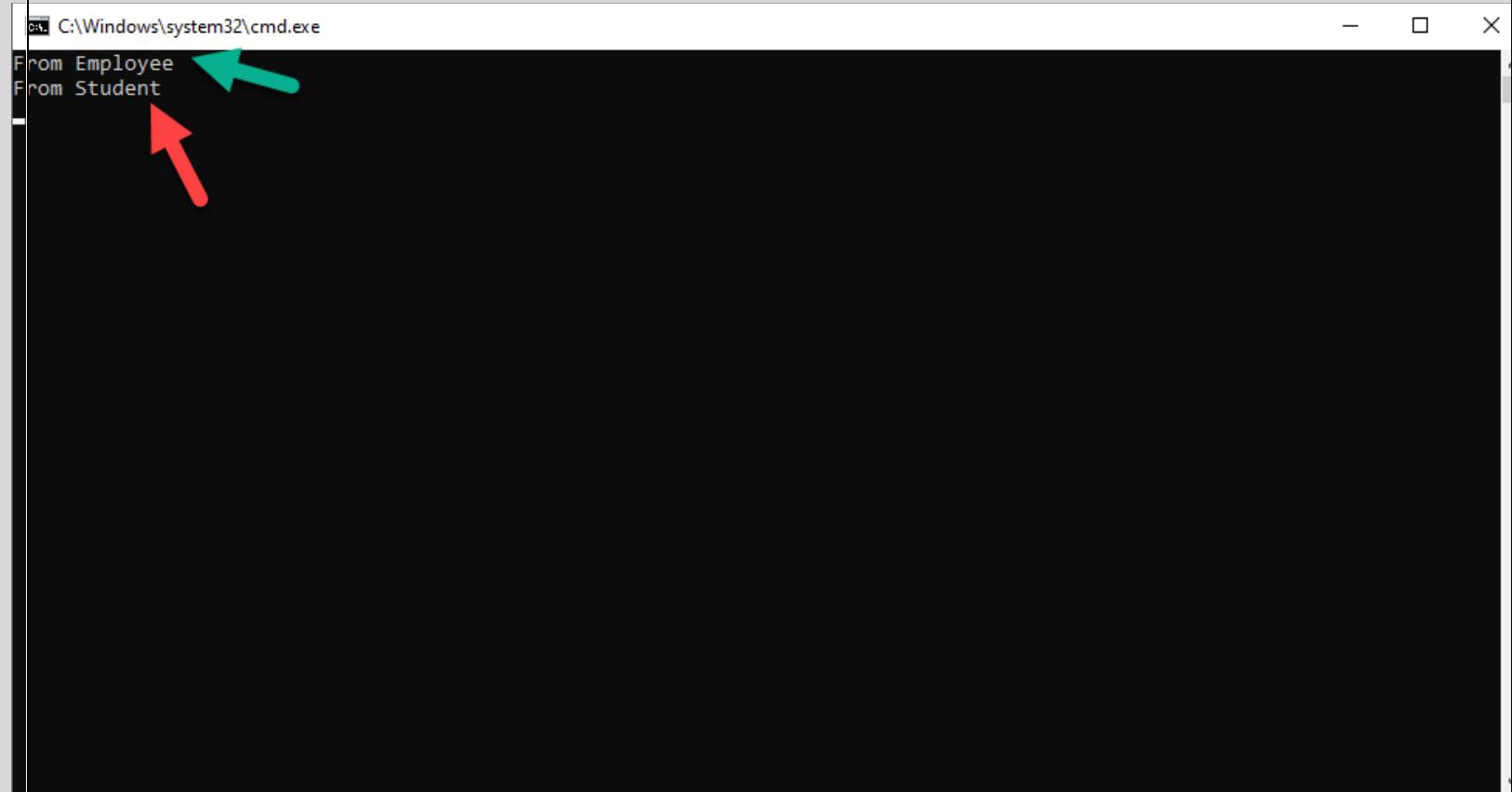
```
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Singleton fromemployee = new Singleton();
14
15            fromemployee.printDetails("From Employee");
16
17            Singleton fromstudent = new Singleton();
18
19            fromstudent.printDetails("From Student");
20            Console.ReadLine();
21        }
22    }
23
24
```

161% Output Ready Ln 20 Col 32 Ch 32 INS Publish

0 references
0 references



مِنْ مُحْتَاجِينَ شَرْح



```
C:\Windows\system32\cmd.exe
From Employee
From Student
```

شغالين نكمل

```
6
7  namespace SingletonDemo
8  {
9      class Singleton
10     {
11         private static int counter = 0; ←
12         public Singleton() ←
13         {
14             counter++;
15             Console.WriteLine("counter Value" + counter.ToString()); ←
16         }
17         public void printDetails(string message)
18         {
19             Console.WriteLine(message);
20         }
21     }
22 }
23
```

The screenshot shows the Microsoft Visual Studio IDE interface with the code editor open to a file named `Singleton.cs`. The code implements a Singleton pattern. Three red arrows point to specific lines of code: the declaration of a static counter variable, the constructor, and the `Console.WriteLine` statement that outputs the current value of the counter.

هنا عملت عداد ونشغل

```
C:\Windows\system32\cmd.exe
```

```
counter Value1  
From Employee  
counter Value2  
From Student
```



بيقولك انتا عامل كذا نسخه من الکلاس تعالی نکمل

The screenshot shows the Microsoft Visual Studio interface with the project "SingletonDemo" open. The code editor displays the file "Singleton.cs" containing the following C# code:

```
6
7     namespace SingletonDemo
8     {
9         public class Singleton
10        {
11            private static int counter = 0;
12            private Singleton()
13            {
14                counter++;
15                Console.WriteLine("counter Value" + counter.ToString());
16            }
17            public void printDetails(string message)
18            {
19                Console.WriteLine(message);
20            }
21        }
22    }
23
```

The class definition is highlighted with a yellow background. The constructor and the `printDetails` method are also highlighted with yellow backgrounds. The code editor shows various status bars at the bottom, including "133 %", "Output", "Item(s) Saved", "Ln 9", "Col 10", "Ch 10", and "INS".

بص هنا غيرت اللي فوق ل
Public
واللي تحت ل
Private
نكمـل

```
6
7 namespace SingletoonDemo
8 {
9     public sealed class Singleton
10    {
11        private static int counter = 0;
12        private Singleton()
13        {
14            counter++;
15            Console.WriteLine("counter Value" + counter.ToString());
16        }
17        public void printDetails(string message)
18        {
19            Console.WriteLine(message);
20        }
}
```

Error List...

Code	Description	Project	File	Line	Suppression St...
CS0122	'Singleton.Singleton()' is inaccessible due to its protection level	SingletoonDemo	Program.cs	13	Active
CS0122	'Singleton.Singleton()' is inaccessible due to its protection level	SingletoonDemo	Program.cs	17	Active

Output

Item(s) Saved Ln 9 Col 17 Ch 17 INS Publish

وصیف کلمة
Sealed
بتمنع انه اى حد یعمل
Inherit
للكلاس بتاعی
نشغل

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** SingletonDemo - Microsoft Visual Studio
- Menu Bar:** File, Edit, View, Project, Build, Debug, Team, Tools, Architecture, Test, Analyze, DevExpress, Window, Help
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Solution Explorer:** Shows the project structure with files Singleton.cs and Program.cs.
- Toolbox:** Standard .NET development tools.
- Code Editor:** Displays the following C# code for Singleton.cs:

```
6
7     namespace SingletoonDemo
8     {
9         public sealed class Singleton
10        {
11            private static int counter = 0;
12            private Singleton()
13            {
14                counter++;
15                Console.WriteLine("counter Value" + counter.ToString());
16            }
17            public void printDetails(string message)
18            {
19                Console.WriteLine(message);
20            }
21        }
22    }
```
- Error List:** Shows two CS0122 errors:

Code	Description	Project	File	Line	Suppression St...
CS0122	'Singleton.Singleton()' is inaccessible due to its protection level	SingletoonDemo	Program.cs	13	Active
CS0122	'Singleton.Singleton()' is inaccessible due to its protection level	SingletoonDemo	Program.cs	17	Active
- Output:** Item(s) Saved
- Status Bar:** Ln 9, Col 17, Ch 17, INS

شوف رفض بيقولك محمية

عشان غيرنا من

Public

لـ

Private

طب تعالى هنشوف نكريت نسخه من الاوبيجيت ازاي

The screenshot shows the code editor in Microsoft Visual Studio with the file `Singleton.cs` open. The code implements the Singleton pattern:

```
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletoonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12
13         private static Singleton instance = null;
14
15         public static Singleton GetInstance
16         {
17             get
18             {
19                 if (instance == null)
20                     instance = new Singleton();
21
22                 return instance;
23             }
24         }
25
26         private Singleton()
27         {
28             counter++;
29             Console.WriteLine("counter Value" + counter.ToString());
30         }
31
32         public void printDetails(string message)
33         {
34             Console.WriteLine(message);
35         }
36     }
37 }
```

Annotations with red arrows point to the following parts of the code:

- An arrow points to the field `private static Singleton instance = null;`.
- An arrow points to the method `public static Singleton GetInstance`.
- An arrow points to the body of the `get` block.
- An arrow points to the condition `if (instance == null)`.
- An arrow points to the assignment `instance = new Singleton();`.

هنا بقوله اعملی متغير جديد اسمه

Instance

نوعه

Singleton

ويكون فارغ وبرايغية

بنزل تحت اعمل ميثود جديد اسمها

GetInstance

نوعها

Singleton

ويعمل فيه

If

يقوله لو المتغير

Instance

فاضى

اعمل الاتى هات المتغير وحطه بداخله نسخه جديدة من اوبيجكت ال

Singleton

واعمل

Return

لـ

BISHOYNABIL
Instance

```
6
7     namespace SingletonDemo
8     {
9         class Program
10        {
11            static void Main(string[] args)
12            {
13                Singleton fromemployee = new Singleton();
14
15                fromemployee.printDetails("From Employee");
16
17                Singleton fromstudent = new Singleton();
18
19                fromstudent.printDetails("From Student");
20                Console.ReadLine();
21            }
22        }
23    }
24
25
```

The screenshot shows the Microsoft Visual Studio interface with the code editor open. The code implements the Singleton design pattern. It defines a `Singleton` class with a private constructor and a public static method `getInstance`. Inside the `Main` method, two instances of `Singleton` are created: `fromemployee` and `fromstudent`, both calling the `printDetails` method.

ایروور نکمل

```
6
7     namespace SingletonDemo
8     {
9         class Program
10        {
11            static void Main(string[] args)
12            {
13                Singleton fromemployee = Singleton.GetInstance;
14
15                fromemployee.printDetails("From Employee");
16
17                Singleton fromstudent = Singleton.GetInstance;
18                ...
19
20                fromstudent.printDetails("From Student");
21                Console.ReadLine();
22            }
23        }
24    }
25
```

مسحت الايور وعدلت الكود طبقا للتعديلات اللي حصلت في الكلاس
كما موضح ونجرب

C:\Windows\system32\cmd.exe

counter Value1
From Employee
From Student



شوف حصل ايه طبعلى الرسائلتين بس
خلى بالك قالك انتا عامل نسخه واحدة من الاوبيجيكت مش زى الاول نسختين

In this session we will learn

- Lazy Initialization in Singleton
- How to use Multithreads in Singleton
- How to implement a Thread Safe singleton class

Facebook.com/PragimTech | Twitter.com/Kudvenkat | PragimTech.com

2

DATA STRUCTURE

SingletonDemo - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEXPRESS WINDOW HELP

Singleton.cs Program.cs

SingletonDemo

```
1 using System.Collections.Generic;
2 using System.Linq;
3 using System.Text;
4 using System.Threading.Tasks;
5
6 namespace SingletonDemo
7 {
8     class Program
9     {
10         static void Main(string[] args)
11         {
12             Singleton fromemployee = Singleton.GetInstance();
13             fromemployee.printDetails("From Employee");
14
15             Singleton fromstudent = Singleton.GetInstance();
16             fromstudent.printDetails("From Student");
17             Console.ReadLine();
18         }
19     }
20 }
```

Quick Actions and Refactorings... Ctrl+j

Run active test(s)

Debug active test(s)

Repeat last test(s)

Show coverage

Quick Actions and Refactorings... Ctrl+j

Rename... Ctrl+R, Ctrl+R

Organize Usings

Show on Code Map Ctrl+j

Find All References on Code Map

Show Related Items on Code Map

Create Unit Tests

Create IntelliTest

Run IntelliTest

Insert Snippet... Ctrl+K, Ctrl+X

Surround With... Ctrl+K, Ctrl+S

Peek Definition Alt+F12

Go To Definition F12

Go To Implementation Ctrl+F12

Find All References Shift+F12

View Call Hierarchy Ctrl+K, Ctrl+T

Breakpoint

Run To Cursor Ctrl+F10

Run To Cursor Ignoring Breakpoints

Run Flagged Threads To Cursor

Execute in Interactive Ctrl+E, Ctrl+E

Cut Ctrl+X

Copy Ctrl+C

Paste Ctrl+V

Insert Comment

Outlining

Find Matching Clones in Solution

Jump to...

Refactor!...

Code!...

133 %

Output Error List

Ready Ln 13 Col 5 Ch Embed Selection Publish

SingletonDemo - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEXPRESS WINDOW HELP

Singleton.cs Program.cs

SingletonDemo

```
1 using System.Collections.Generic;
2 using System.Linq;
3 using System.Text;
4 using System.Threading.Tasks;
5
6
7 namespace SingletonDemo
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            Singleton fromemployee = Singleton.GetInstance;
16            ...
17            Singleton fromemployee = Singleton.GetInstance;
18            NewMethod();
19            fromemployee.printDetails("From Employee");
20            ...
21            Singleton fromstudent = Singleton.GetInstance;
22            ...
23            ...
24            ...
25        }
26    }
27 }
```

Extract Method

...
fromemployee = Singleton.GetInstance;
NewMethod();
fromemployee.printDetails("From Employee");
Singleton fromstudent = Singleton.GetInstance;
...
private static void NewMethod()
{
 Singleton fromemployee = Singleton.GetInstance;
 fromemployee.printDetails("From Employee");
}
...

Preview changes

133 %

Output Error List

Ready Ln 13 Col 5 Ch 5 INS Publish

SingletonDemo - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEXPRESS WINDOW HELP

Singleton.cs Program.cs* # X

SingletonDemo

Quick Launch (Ctrl+Q) Sign in

Server Explorer Toolbox Notifications Solution Explorer Team Explorer Class View Properties

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletonDemo
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             NewMethod();
14             ...
15
16             Singleton fromstudent = Singleton.GetInstance;
17
18             fromstudent.printDetails("From Student");
19             Console.ReadLine();
20         }
21
22         private static void NewMethod()
23         {
24             Singleton fromemployee = Singleton.GetInstance;
25
26             fromemployee.printDetails("From Employee");
27         }
28     }
29 }
```

Rename: NewMethod

Modify any highlighted location to begin renaming.

Include comments
 Include strings
 Preview changes

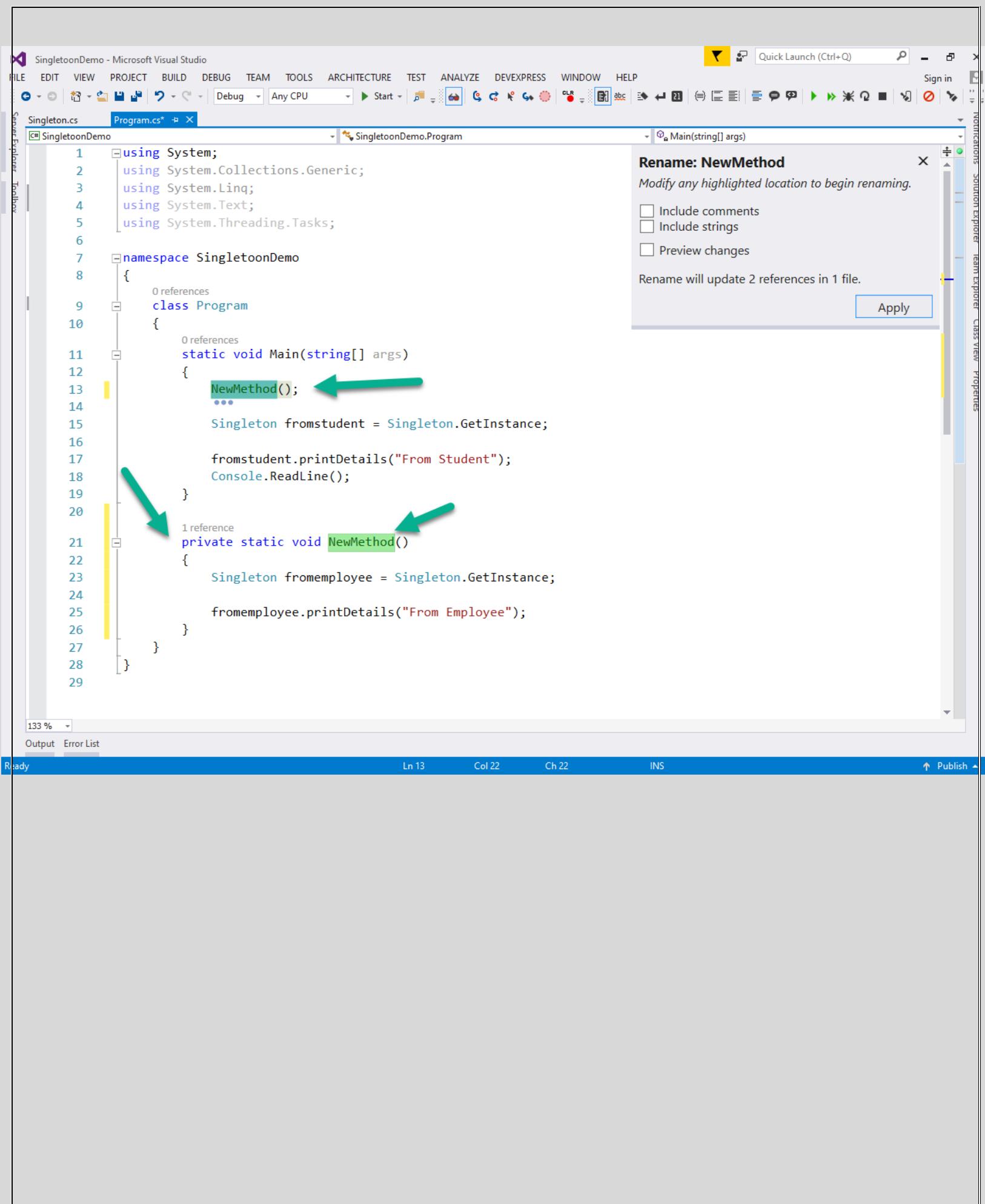
Rename will update 2 references in 1 file.

Apply

133 %

Output Error List

Ready Ln 13 Col 22 Ch 22 INS ↑ Publish ▾



The screenshot shows the Microsoft Visual Studio interface with the 'SingletonDemo' project open. The 'Program.cs' file is selected. A context menu is open over the 'PrintStudenDetails()' method, and a 'Rename' dialog box is displayed. The 'New name:' field contains 'PrintStudenDetails'. There are three checkboxes: 'Include comments', 'Include strings', and 'Preview changes', all of which are unchecked. Below the dialog, a note states 'Rename will update 2 references in 1 file.' with an 'Apply' button.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SingletonDemo
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            PrintStudenDetails();
14
15            Singleton fromstudent = Singleton.GetInstance;
16
17            fromstudent.printDetails("From Student");
18            Console.ReadLine();
19        }
20
21        private static void PrintStudenDetails()
22        {
23            Singleton fromemployee = Singleton.GetInstance;
24
25            fromemployee.printDetails("From Employee");
26        }
27    }
28 }
29
```

غير الاسم

احنا هنا هنشوف ازاي نعمل

Multi thread

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** SingletonDemo - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Code Editor:** Displays the `Program.cs` file from the `SingletonDemo` project. The code implements the Singleton design pattern. Two specific lines of code are highlighted with green arrows:
 - `printEmployeeDetails();` at line 15
 - `fromstudent.printDetails("From Student");` at line 23
- Status Bar:** Shows "Item(s) Saved", "Ln 16", "Col 10", "Ch 10", and "INS".
- Solution Explorer:** Shows the project structure with files like `Singleton.cs` and `Program.cs`.
- Toolbox:** Standard .NET toolbox items.
- Task List:** Shows "Main(string[] args)".
- Properties:** Standard file properties.

نکمل
تعالی نشوف ازای لما نعمل
multi-threaded
هیحصل ایه

SingletonDemo - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Singleton.cs Program.cs*

SingletonDemo

```
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SingletonDemo
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Parallel.Invoke()
14                ^ 1 of 2 void Parallel.Invoke(params Action[] actions)
15                Executes each of the provided actions, possibly in parallel.
16                actions: An array of Action to execute.
17            printEmployeeDetails();
18        }
19
20        private static void printEmployeeDetails()
21        {
22            Singleton fromstudent = Singleton.GetInstance();
23
24            fromstudent.printDetails("From Student");
25            Console.ReadLine();
26        }
27
28        private static void PrintStudenDetails()
29        {
30            Singleton fromemployee = Singleton.GetInstance();
31
32            fromemployee.printDetails("From Employee");
33        }
34    }
35 }
```

Output Error List

Ready Ln 13 Col 29 Ch 29 INS ↑ Publish ▾

بنستخدم

Parallel.Invoke

عشان ننادي على الميثود الآتنين

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "SingletonDemo - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, DEVEEXPRESS, WINDOW, and HELP. The toolbar has various icons for file operations like Open, Save, Print, and Build. The status bar at the bottom shows "Item(s) Saved", "Ln 16", "Col 40", "Ch 40", and "INS".

The code editor window displays the file "Program.cs" with the following content:

```
namespace SingletonDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Parallel.Invoke(
                () => PrintStudentDetails(),
                () => printEmployeeDetails());
        }

        private static void printEmployeeDetails()
        {
            Singleton fromstudent = Singleton.GetInstance;

            fromstudent.printDetails("From Student");
            Console.ReadLine();
        }

        private static void PrintStudentDetails()
        {
            Singleton fromemployee = Singleton.GetInstance;

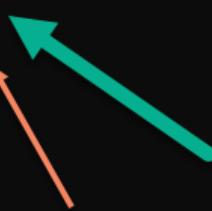
            fromemployee.printDetails("From Employee");
        }
    }
}
```

Two green arrows point to the lines of code: `Parallel.Invoke(` and `() => printEmployeeDetails());`.

بنستخدم اللمبادا اكسبريسين

```
C:\Windows\system32\cmd.exe
```

```
counter Value1  
counter Value2  
From Employee  
From Student
```



شوف حايلى 2 او بجيكت او 2

Value

بص كده طلب مرتبين الميثود

حصل

multi-threaded

وده غلط وكمان عشان بينادى على

GetInstance

وده غلط ده عكس مبادى ال

Singleton

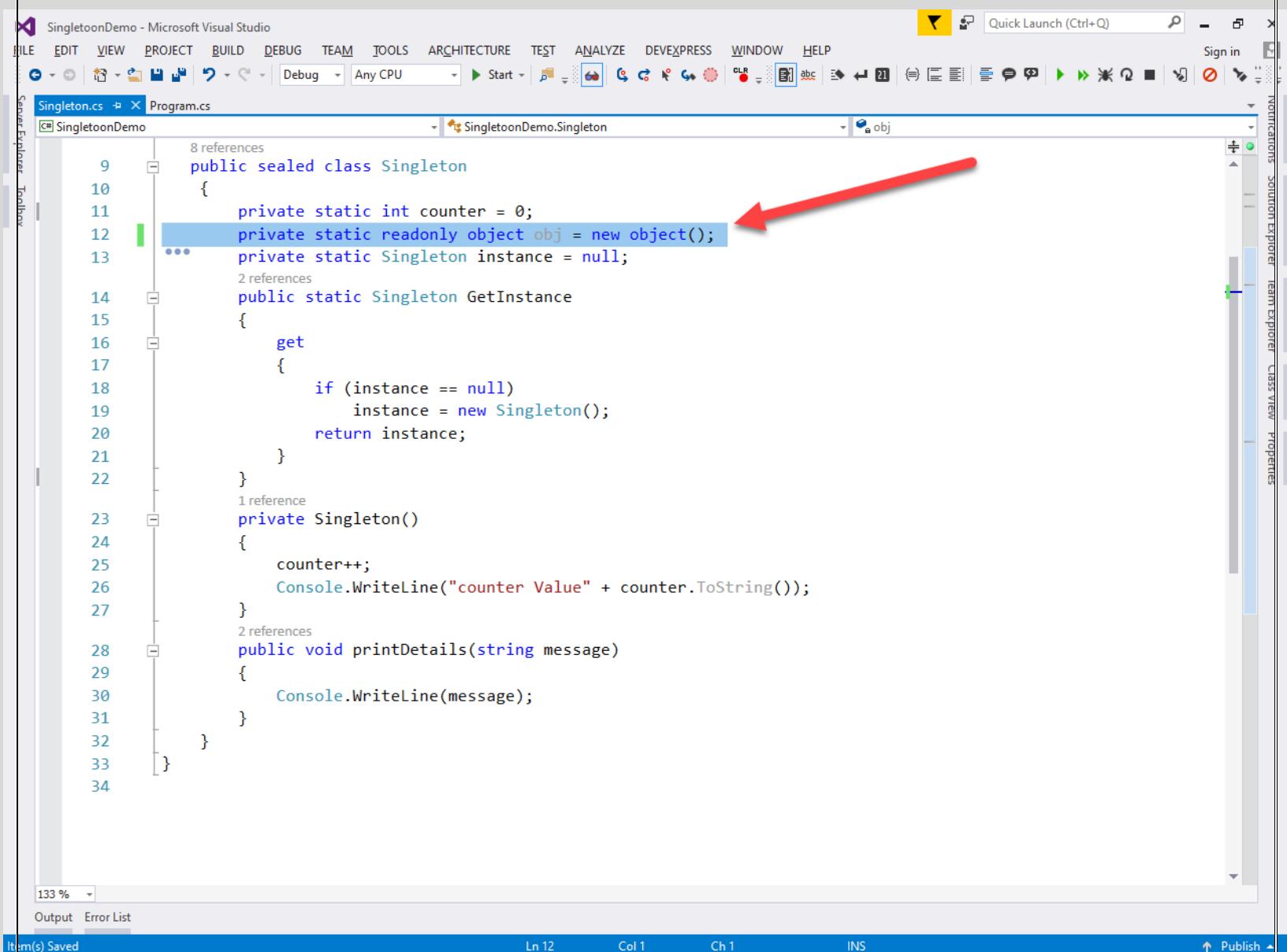
يقولك افضل حل انه نعمله

Lock

وهو نعمل

Lock

للمشود بتاعتنا بحيث



```
SingleDemo - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
SingleDemo.cs Program.cs
SingleDemo
8 references
9     public sealed class Singleton
10    {
11        private static int counter = 0;
12        private static readonly object obj = new object(); ← Red arrow here
13        ...
14        public static Singleton GetInstance
15        {
16            get
17            {
18                if (instance == null)
19                    instance = new Singleton();
20                return instance;
21            }
22        }
23        private Singleton()
24        {
25            counter++;
26            Console.WriteLine("counter Value" + counter.ToString());
27        }
28        public void printDetails(string message)
29        {
30            Console.WriteLine(message);
31        }
32    }
33
34
```

بنعمل متغير اسمه اوبيجكت

The screenshot shows the Microsoft Visual Studio interface with the project "SingletonDemo" open. The code editor displays the file "Singleton.cs" containing the implementation of the Singleton pattern. Red arrows point to the lock statement and the if condition within the GetInstance method.

```
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
```

133 %

Output Error List

Item(s) Saved Ln 19 Col 28 Ch 28 INS ↑ Publish

بص كده مفيش غير ميثود واحدة هيا اللي هتوصل للعنصر ده

او كده مفيش

غير

Thread

واحدة فقط تقدر تدخل لل코드 ده يعني لو اوبجيكت

Student

و

Employee

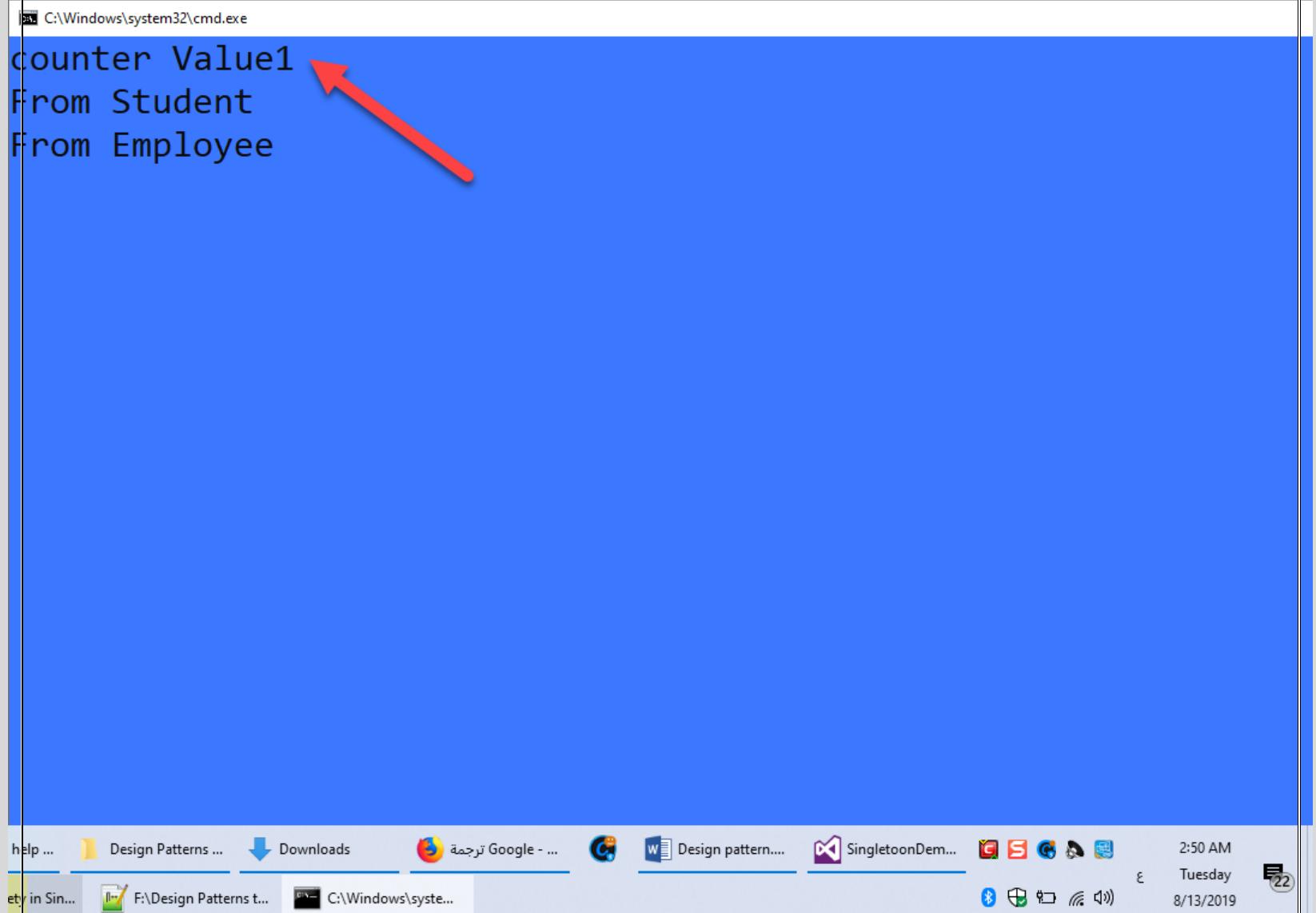
مش هيقدر و الاتنين يدخلو للكود بتابعى وهنوقف الـ

Multi thread

ده وهنخللى او بجيكت واحد بس الللى يدخل

لية لانه بالكود ده واحدة منهم فقط الللى هتنجح للوصول للبلوك او الكود ده فقط وهتتقفل على الثانية

وكمان تانى ميتوعد هتسننى لما الاول تخلص تعالى نشغل



```
C:\Windows\system32\cmd.exe
counter Value1
From Student
From Employee
```

The screenshot shows a Windows Command Prompt window with the title bar "C:\Windows\system32\cmd.exe". Inside the window, there are two lines of text: "counter Value1" and "From Student". Below the window, the taskbar is visible with several icons and open application windows. A red arrow points from the text "From Student" towards the top-left corner of the slide content.

بص مفيش غير 1

معنی كده انى قدرت احدد ميكنش فيه

Multi thread

واوبجيكت واحد فقط نجح فى الوصول كده ال

Lock

شغال بنجاح

بس فيه مشكلة ال

Lock

بيستهلك كتير من موارد الجهاز والحل كالاتى عشان نقلل من استخدامه الكبير

The screenshot shows the Microsoft Visual Studio IDE with the 'SingletonDemo' project open. The 'Singleton.cs' file is the active code editor. The code implements the Singleton pattern with a static constructor and a GetInstance method. A red arrow points to the 'if (instance == null)' check in the GetInstance method. Another red arrow points to the 'lock (obj)' statement. A green arrow points to the 'return instance;' statement. The code is as follows:

```
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletoonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13         private static Singleton instance = null;
14
15         public static Singleton GetInstance
16         {
17             get
18             {
19                 if (instance == null)
20                 {
21                     lock (obj)
22                     {
23                         if (instance == null)
24                             instance = new Singleton();
25                     }
26                 }
27             }
28
29             private Singleton()
30             {
31                 counter++;
32                 Console.WriteLine("counter Value" + counter.ToString());
33             }
34
35             public void printDetails(string message)
36         }
37     }
38 }
```

بنعمل شرط تحقق الاول وبس

C:\Windows\system32\cmd.exe

```
counter Value1
From Employee
From Student
```



نجح

وعادى نكمل

هنكل نفس الكورس بالعربي المحاضر عربي ومتدرجة

Lazy loading vs Eager initialization

The screenshot shows a PowerPoint slide with a dark blue background. At the top, the title 'In this session we will learn' is displayed in large yellow font. Below the title is a bulleted list of three items, also in yellow. Further down the slide, there is a section titled 'Suggested Videos:' in yellow, followed by four video links in white text. At the bottom of the slide, there is a link to a YouTube playlist in white text. The slide has a navigation bar at the bottom with icons for back, forward, and search, and a page number '2' in a circle.

PowerPoint Slide Show - [Part 5 - Lazy vs Eager initialization [Compatibility Mode]] - PowerPoint

In this session we will learn

- Lazy Initialization vs Eager Initialization
- Singleton with eager loading
- Singleton with Lazy keyword (.NET 4.0).

Suggested Videos :

Part 4– Thread Safety in Singleton
Part 3 – Why is Singleton class sealed
Part 2 – Singleton Design Pattern
Part 1 – Design Patterns Tutorial

Link to Dot Net Basics, C#, LINQ,SQL Server, ASP.NET, ADO.NET and EF video series
<https://www.youtube.com/c/kudvenkatarabic/playlists>

2

Lazy vs Eager Initialization

Lazy Loading

- Improves the performance.
- Avoids unnecessary load till the point object is accessed.
- Reduces the memory for print on the start-up of the program.
- Faster application load.

Non-Lazy or Eager Loading

- Initialize the required object before it's being accessed.
- The common Language Runtime (CLR) takes care of the variable initialization and its thread safety so lower memory used for prints.

4

Slide 4 of 8

الفرق بينهم ومهم جداً اولاً

Lazy loading

الاداء افضل

وبقوم بتحميل الداتا المتعلقة بالاوبيجيكت لما اعمل

اكسيس على الاوبيجيكت

بتحسن كمية البيانات اللي هتخزن داخل الميموري

او بتؤمن انى احمل البيانات الخاصة بالاوبيجيكت لما اعمل

اكسيس على الاوبيجيكت

فبالتالي هو بيحمل الداتا وده يحسن بشكل كبير الاداء

وكمان يخفف مقدار البيانات المخزنة داخل الميموري

وممكن مستخدمش في بعض الاحيان واستخدم

ال
Eager

نروح لـ
Eager

يعمل ايه بتحمل كل البيانات مرة واحدة قبل حتى ما ادخل او اعمل اكسيس لاي
اوبيجيك

وبالتالي حجم البيانات هيكون اكبر
وده هيأثر بشكل عام على اداء البرنامج
وبيقولك الـ

BISHOYNABI Clr
هو اللي بيهمت بالمتغيرات وبيعمل لها

Initialization

وبيعمل

Thread safty

وكده هيستهلك من الرامات قليل جدا

Singleton with Eager Loading

How to Convert the Singleton from Lazy to Eager Initialization ?

We need to follow these steps :

Step 1 : Change null initialization in static instance field to new singleton instance.

Step 2 : Change static instance field to read-only.

Step 3 : Remove the double check locking code and return instance and static object.

Note : After all these changes the thread is safe because the CLR takes care of the variable initialization and it's thread safety.

5

تعالی نشووف هنحول ازای من

Lazy

الى

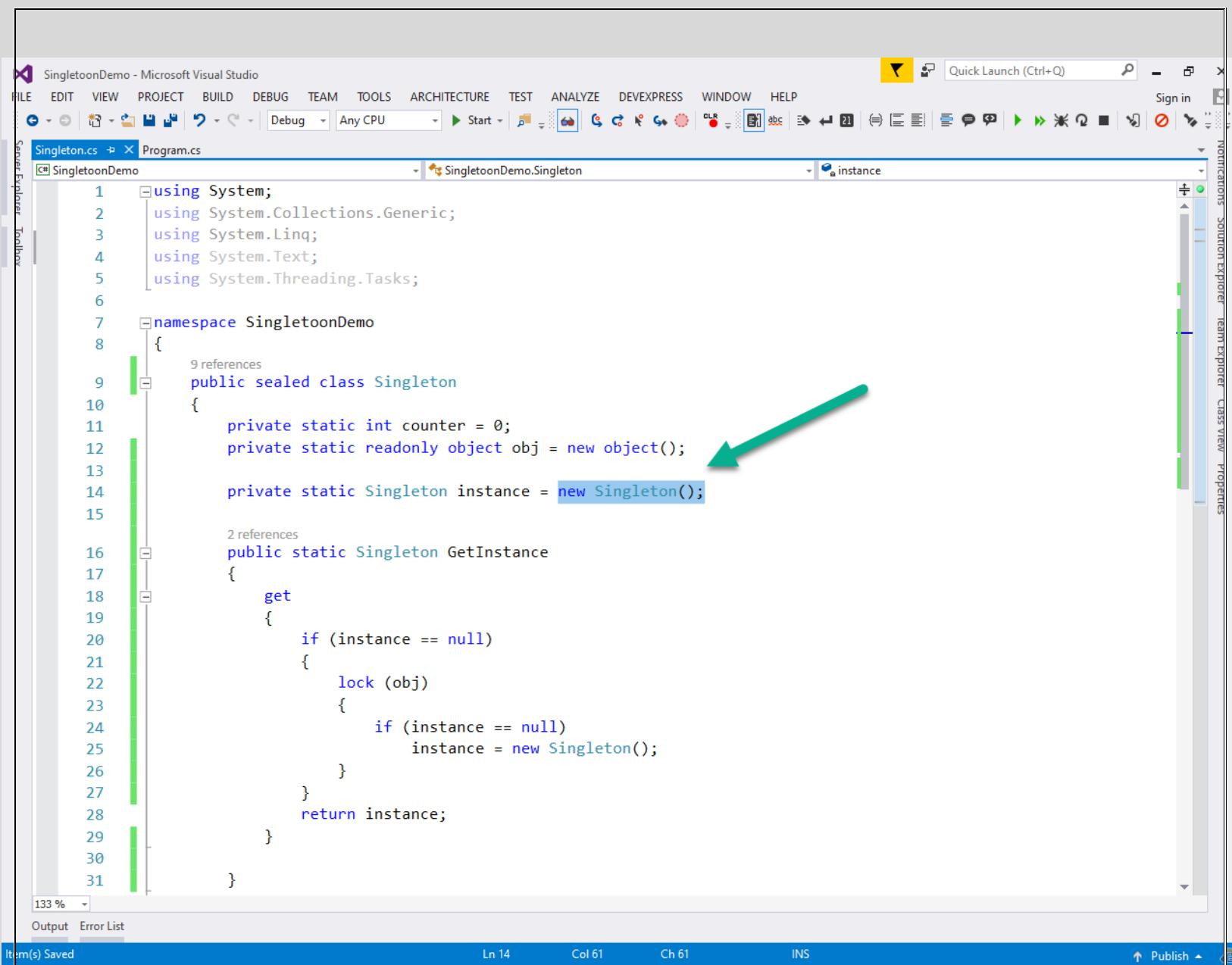
Eager

كما موضح في ثلاث خطوات

The screenshot shows the Microsoft Visual Studio interface with the title bar "SingletonDemo - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, DEVEEXPRESS, WINDOW, and HELP. The toolbar contains various icons for file operations like Open, Save, Print, and Build. The solution explorer on the left shows "Singleton.cs" and "Program.cs" under "SingletonDemo". The code editor displays the following C# code for Singleton.cs:

```
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13         private static Singleton instance = null;
14
15         public static Singleton GetInstance
16         {
17             get
18             {
19                 if (instance == null)
20                 {
21                     lock (obj) {
22                         if (instance == null)
23                             instance = new Singleton();
24                     }
25                     return instance;
26                 }
27             }
28         }
29
30         private Singleton()
31         {
32             counter++;
33             Console.WriteLine("counter Value" + counter.ToString());
34         }
35
36         public void printDetails(string message)
37     }
```

هنغيرها الى



Singleton.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletoonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13
14         private static Singleton instance = new Singleton(); new Singleton()
15
16         public static Singleton GetInstance
17         {
18             get
19             {
20                 if (instance == null)
21                 {
22                     lock (obj)
23                     {
24                         if (instance == null)
25                             instance = new Singleton();
26                     }
27                 }
28             return instance;
29         }
30     }
31 }
```

نکمل

The screenshot shows the Microsoft Visual Studio IDE interface with the title bar "SingletonDemo - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, DEVEEXPRESS, WINDOW, and HELP. The toolbar has various icons for file operations like Open, Save, Print, and Build. The status bar at the bottom shows "Item(s) Saved", "Ln 14", "Col 32", "Ch 32", "INS", and "↑ Publish".

The code editor displays the file "Singleton.cs" under the project "SingletonDemo". The code implements the Singleton pattern:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13
14         private static readonly Singleton instance = new Singleton(); // Line 14
15
16         public static Singleton GetInstance
17         {
18             get
19             {
20                 if (instance == null)
21                 {
22                     lock (obj)
23                     {
24                         if (instance == null)
25                             instance = new Singleton(); // Line 25
26                     }
27                 }
28                 return instance;
29             }
30         }
31     }
}
```

A green vertical bar highlights the entire code block. A red arrow points to the line "private static readonly Singleton instance = new Singleton();". A callout bubble with the text "خطوة بنصييف" (Step by step) is positioned above the line.

خطوة بنصييف

Read only

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Title Bar:** SingletonDemo - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Code Editor:** Displays the `Singleton.cs` file under the `SingletonDemo` project. The code implements the Singleton pattern with a private static counter and a lock object.
- Status Bar:** Shows "Item(s) Saved", "Ln 20", "Col 9", "Ch 9", and "INS".

```
public sealed class Singleton
{
    private static int counter = 0;
    private static readonly object obj = new object();

    private static readonly Singleton instance = new Singleton();

    public static Singleton GetInstance
    {
        get
        {
            if (instance == null)
            {
                lock (obj)
                {
                    if (instance == null)
                        instance = new Singleton();
                }
            }
            return instance;
        }
    }

    private Singleton()
    {
        counter++;
        Console.WriteLine("counter Value" + counter.ToString());
    }

    public void printDetails(string message)
    {
        Console.WriteLine(message);
    }
}
```

ثالث خطوة هنمسح الكود الموضح

```
C:\Windows\system32\cmd.exe
```

```
counter Value1  
From Employee  
From Student
```



بعض عمل اوجيكت واحد فقط بالرغم من انا بعمل

Multi thread

`() => PrintStudentDetails(),
() => printEmployeeDetails());`

هنا

انا عملت كده

Eager loading

وعملی هنا

```
private static readonly Singleton instance = new Singleton();
```

اوبيكت جديد قبل ما ادخل هنا

```
public static Singleton GetInstance  
{  
    get  
    {  
        return instance;  
    }  
}
```

}

ده كده ال
Eager
وخلی بالک هنا ال
Eager
هو المسؤول عن انشاء المتغيرات وال
Multi thread

The slide title is "Singleton with eager loading". It shows two side-by-side code snippets for a Singleton class.

Left Code Snippet:

```
public sealed class Singleton
{
    private static int counter = 0;
    private static Singleton instance = null;
    private static readonly object obj = new object();
    public static Singleton GetInstance
    {
        get
        {
            if (instance == null)
            {
                lock (obj)
                {
                    if (instance == null)
                    {
                        instance = new Singleton();
                    }
                }
            }
            return instance;
        }
    }
    private Singleton()
    {
        counter++;
        Console.WriteLine("Counter Value = " + counter.ToString());
    }
    public void PrintDetails(string meassage)
    {
        Console.WriteLine(meassage);
    }
}
```

Right Code Snippet:

```
public sealed class Singleton
{
    private static int counter = 0;
    private static readonly Singleton instance = new Singleton();
    public static Singleton GetInstance
    {
        get
        {
            return instance;
        }
    }
    private Singleton()
    {
        counter++;
        Console.WriteLine("Counter Value = " + counter.ToString());
    }
    public void PrintDetails(string meassage)
    {
        Console.WriteLine(meassage);
    }
}
```

6

قبل وبعد التعديل
قبل التعديل شمال

بعد التعديل يمين

PowerPoint Slide Show - [Part 5 - Lazy vs Eager initialization [Compatibility Mode]] - PowerPoint

Singleton with Lazy Loading

How to Convert the Singleton from Eager to Lazy Initialization using Lazy keyword (.NET Framework 4.0) ?

We need to follow these steps :

Step 1 : Wrap the singleton object in a Lazy manner.

```
private static Singleton instance = null;
```

↓

```
private static Lazy<Singleton> instance = new Lazy<Singleton>();
```

Step 2 : Create a new singleton instance using lambda expression as a delegate.

```
private static Lazy<Singleton> instance = new Lazy<Singleton>(()=> new Singleton());
```

Step 3 : Access to the singleton object using value property.

```
return instance.Value;
```

Note : By default lazy objects are correct safe in multi-thread.

7

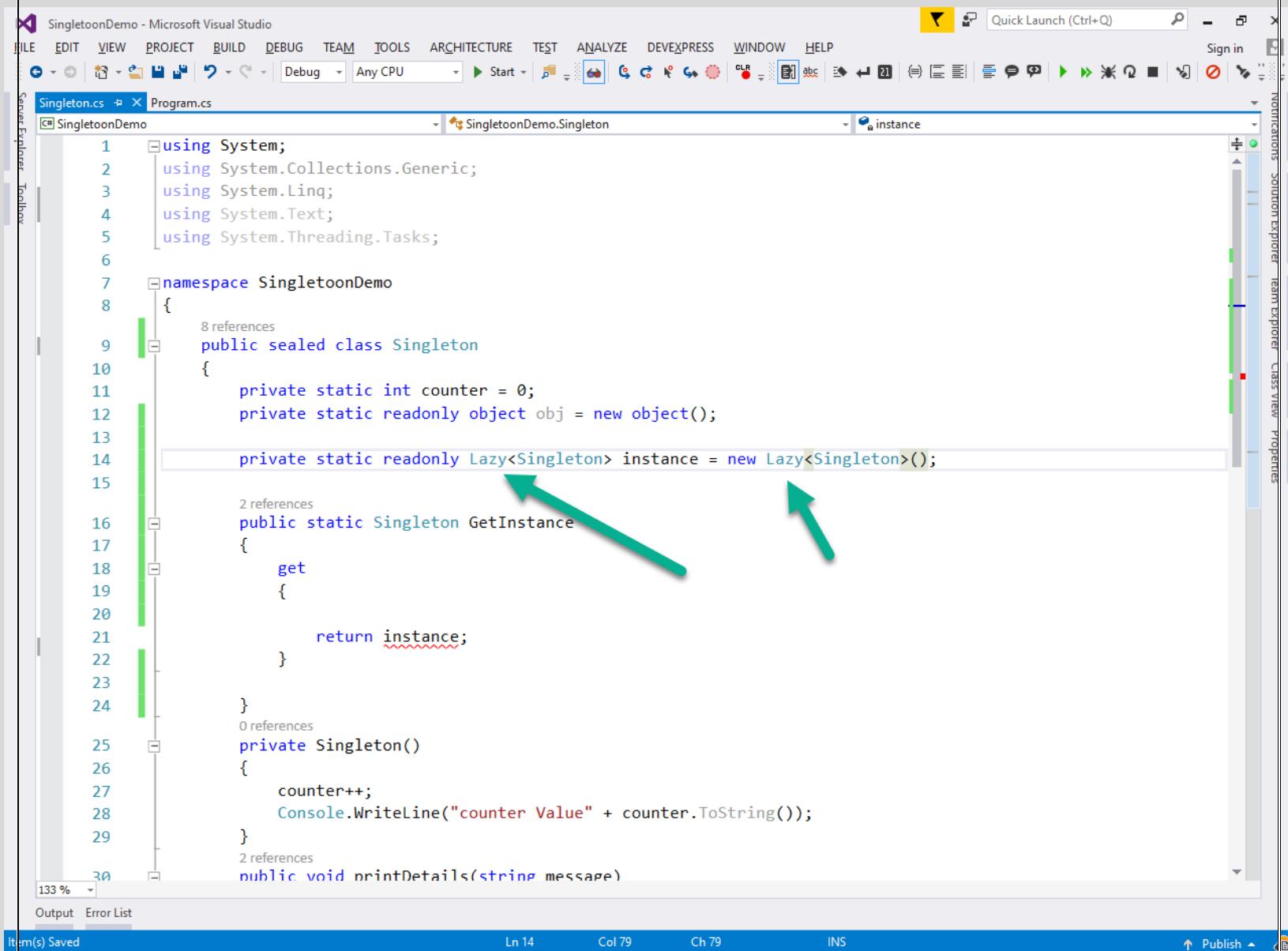
تعالی نشووف ازای نحوں من ال

Eager

الی ال

Lay loading

اول خطوة



```
Singleton.cs - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
Singleton.cs Program.cs
C# SingletonDemo
Singleton.cs -> SingletonDemo.Singleton instance
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13
14         private static readonly Lazy<Singleton> instance = new Lazy<Singleton>();
15
16         public static Singleton GetInstance
17         {
18             get
19             {
20
21                 return instance;
22             }
23
24         }
25         private Singleton()
26         {
27             counter++;
28             Console.WriteLine("counter Value" + counter.ToString());
29         }
30         public void printDetails(string message)
31     }
32 }
```

اول خطوة بنسخدم كلمة

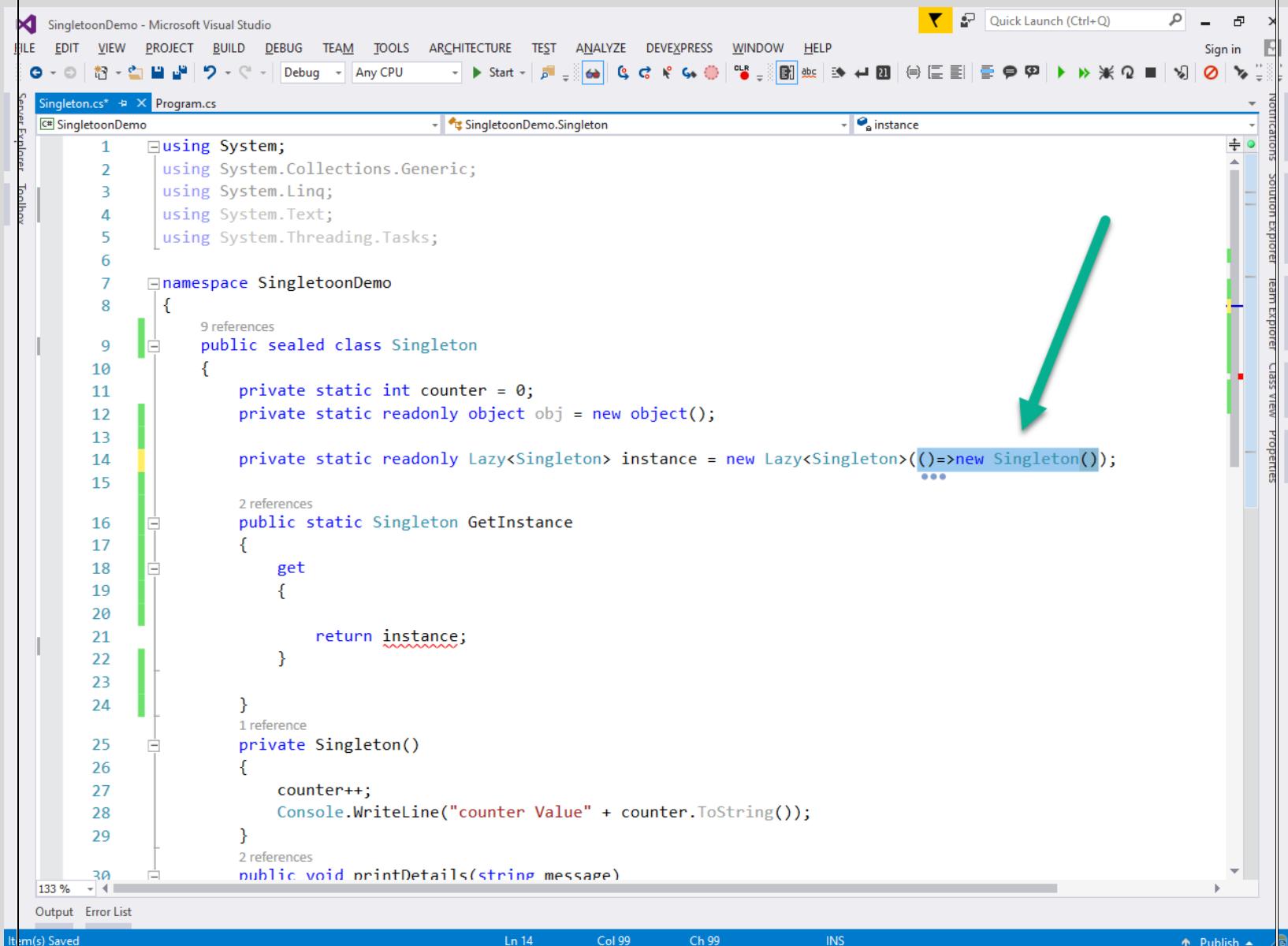
Lazy<>

تاني خطوة هنعمل

Delegate

باستخدام الـ

Lambda



The screenshot shows the code editor in Microsoft Visual Studio with the file `Singleton.cs` open. The code implements the Singleton pattern using a `Lazy<Singleton>` lambda expression. A large green arrow points from the explanatory text below to the line of code where the lambda expression is used.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SingletonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13
14         private static readonly Lazy<Singleton> instance = new Lazy<Singleton>(()=>new Singleton());
15
16         public static Singleton GetInstance
17         {
18             get
19             {
20                 return instance;
21             }
22         }
23
24     }
25     private Singleton()
26     {
27         counter++;
28         Console.WriteLine("counter Value" + counter.ToString());
29     }
30     public void printDetails(string message)
31     {
32     }
33 }
```

ثالث خطوة

SingletonDemo - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Singleton.cs* X Program.cs

SingletonDemo

SingletonDemo.Singleton

GetInstance

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;

6
7  namespace SingletonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13
14         private static readonly Lazy<Singleton> instance = new Lazy<Singleton>(()=>new Singleton());
15
16         public static Singleton GetInstance
17         {
18             get
19             {
20                 return instance.Value;
21             }
22         }
23     }
24 }
25 private Singleton()
26 {
27     counter++;
28     Console.WriteLine("counter Value" + counter.ToString());
29 }
30 public void printDetails(string message)
```

133 %

Output Error List

Ready

Ln 21 Col 33 Ch 33 INS

Quick Launch (Ctrl+Q) Sign in

Server Explorer Toolbox Notifications Solution Explorer Team Explorer Class View Properties

SingletonDemo - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Singleton.cs Program.cs

SingletonDemo.Singleton

GetInstance

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;

6
7  namespace SingletonDemo
8  {
9      public sealed class Singleton
10     {
11         private static int counter = 0;
12         private static readonly object obj = new object();
13
14         private static readonly Lazy<Singleton> instance = new Lazy<Singleton>(()=>new Singleton());
15
16         public static Singleton GetInstance
17         {
18             get
19             {
20                 return instance.Value; // Line 21
21             }
22         }
23
24     }
25     private Singleton()
26     {
27         counter++;
28         Console.WriteLine("counter Value" + counter.ToString());
29     }
30     public void printDetails(string message)
31     {
32     }
33 }
```

133 %

Output Error List

Item(s) Saved

Ln 21 Col 38 Ch 38 INS

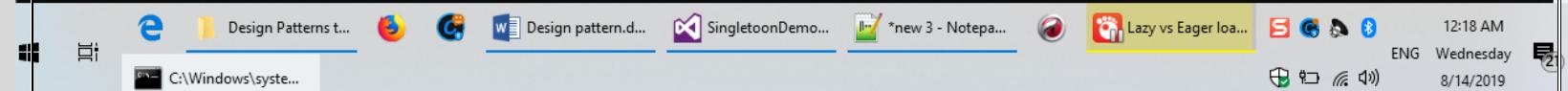
Quick Launch (Ctrl+Q)

Sign in

Notifications Solution Explorer Team Explorer Class View Properties



```
C:\Windows\system32\cmd.exe
counter Value1
From Employee
From Student
```



نجح
وتحولنا من ال
Eager initialization
إلى
Lazy Initialization

اللى كان فى الاصل خالص فى الاول قبل ما نحوال كان
Lazy initialization

In this session we will learn

- Differences between Singleton and static classes
- Static class example
- Real world usage of Singleton

Suggested Videos :

- Part 5 – Lazy vs Eager Loading in Singleton
- Part 4 – Thread Safety in Singleton
- Part 3 – Why is Singleton class sealed
- Part 2 – Singleton Design Pattern

Link to Dot Net Basics, C#, LINQ,SQL Server, ASP.NET, ADO.NET and EF video series
<https://www.youtube.com/c/kudvenkatarabic/playlists>

Recap

Versions of Singleton Design Pattern

- First version of Singleton.
- Multithread support in Singleton.
- Eager loading.
- Singleton with Lazy (.NET 4.0).

3

Slide 3 of 6



هنا فى الصورة دى فى الاول اخدنا ال

الاصدار الاول من ال

ولما لقيت اكتر من ثريد بتحصل مشكلة

ثانى حاجة ازاي نخلية يدعم المالتى ثردىنج

ثالث حاجة حول الى

Eager

بعدين حول الى

Lazy

Static vs Singleton

- **Static is a keyword and Singleton is a design pattern.**
- **Static classes can contain only static members.**
- **Singleton is an object creational pattern with one instance of the class.**
- **Singleton can implement interfaces, inherit from other classes and it aligns with the OOPS concepts.**
- **Singleton object can be passed as a reference where as Static object can't be passed as reference to other methods or objects.**
- **Singleton supports object disposal.**
- **Singleton object is stored on heap where as static classes are stored on stacks.**
- **Singleton objects can be cloned.**

4

Slide 4 of 6

BISHOY NABIL
الفرق بين الـ

Static

و

Singleton

وخلی بالک فیه لغبطة بینهم

هنشوف الفرق

الـ

Static

هیا کلمة موجودة فی الدوت نت لكن

الـ

Singleton

ھی

Design pattern

ثانياً الـ

Static class

تحتوي علىـ

Statics Member

زى الـ

Method

وغيرها

BISHOYNABIL
ثالث الـ
بيقولك الـ

Singleton

يضمـلك انه يـكون عندكـ

Instance

واحدة من الكلاس فقطـ

ويـقولك الـ

Singleton

بـمكانـه يـعملـ

Implement interfaces

وكـمان يـقدر يـعملـ

Inherit

من كـلاسـات اـخـرى وـكمـان تـقدـر تـستـخدـم معـاه مـفـاهـيم الـ

Oop

Object oriented programing

وكمان ال

Singleton

تقدر تبعته ك

Refrence

الى ال

STATIC OBJECT

فى حين انك متقدرش تعيبت ال

STATIC

الى اوجيكت اخر

BISHOYNABIL
وبقولك ال

SINGLETON

بتدعم مفهوم ال

Dispose

وكمان ال

Singleton

بتحزن فى ال

Heap

بينما ال

Static class

بتحزن فى ال

Stack

وكمان ال Singleton

بتدعم ال استنساخ

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the project "SingletonDemo" with files "Converter.cs" and "Program.cs".
- Code Editor:** Displays the "Converter.cs" file content:

```
1  using System.Collections.Generic;
2  using System.Linq;
3  using System.Text;
4  using System.Threading.Tasks;
5
6
7  namespace StaticDemo
8  {
9      static class Converter
10    {
11        public static double ToFahrenheit(double celcius)
12        {
13            return (celcius * 9 / 5) + 32;
14        }
15
16        public static double ToCelsius(double Fahrenheit)
17        {
18            return (Fahrenheit - 32) * 5 / 9;
19        }
20    }
21
22
23 }
```

The code uses a static class named "Converter" with two static methods: "ToFahrenheit" and "ToCelsius".- Toolbars and Menus:** Standard Visual Studio menus like FILE, EDIT, PROJECT, and DEBUG are visible at the top.
- Status Bar:** Shows "Item(s) Saved", "Ln 19", "Col 13", "Ch 13", and "INS".

عملت بروجيكت جديد و عملت كلاس هيكون محول درجة حرارة
عملت 2 ميثود فيه واحد بيحول الى سيليوز والثانى فهرنهايت
نكمـل

```
SingleDemo - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP
Debug Any CPU StaticDemo Start CLR abc
Server Explorer Toolbox
Converter.cs Program.cs Singleton.cs Program.cs
StaticDemo
+ using System;
5 using System.Threading.Tasks;
6
7 namespace StaticDemo
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            double celcius = 37;
14            double fahrenheit = 98.6;
15
16            Console.WriteLine("Value of {0} celcius to fahrenheit is {1}", celcius, Converter.ToFahrenheit(celcius));
17
18            Console.WriteLine("Value of {0} Fahrenheit to celcius is {1}", fahrenheit, Converter.Tocelcius(fahrenheit));
19
20            Console.ReadLine();
21        }
22    }
23
24
25
Main(string[] args)
```

تشغيل

0x Select C:\Windows\system32\cmd.exe

```
Value of 37 celcius to fahrenheit is 98.6  
Value of 98.6 Fahrenheit to celcius is 37
```

شغاله

نكمـل

طب انا ليه استخدم

Singleton

بدلا من الـ

Static

لانه هيدينى مرونة فى كتابة الكود اكتر

وامـتا استخدم الـ

Singleton

PowerPoint Slide Show - [Part 6 - Static Class vs Singleton [Compatibility Mode]] - PowerPoint

Singleton Real World Usages

- Exception/Information logging.
- Managing a connection or a pool of connections to Database.
- Device management such as printer spooling.
- File Management.
- Application Configuration management.
- Cache management.
- Session based Shopping cart.

Slide 5 of 6

في الحياة العملية في
الاكسبيشن ومعلومات تسجيل الدخول
ادارة الكونكشن مع الداتا بيز
ادارة الملفات
وكمان تعريفات الاجهزه البرنتر
وبنستخدمها مع الحاجات اللي في الصورة فوق دى كلها
:D

In this session we will learn

- Employee Portal MVC Application
- Logger library using Singleton Design Pattern
- Log exceptions using the Logger library in the MVC Portal.

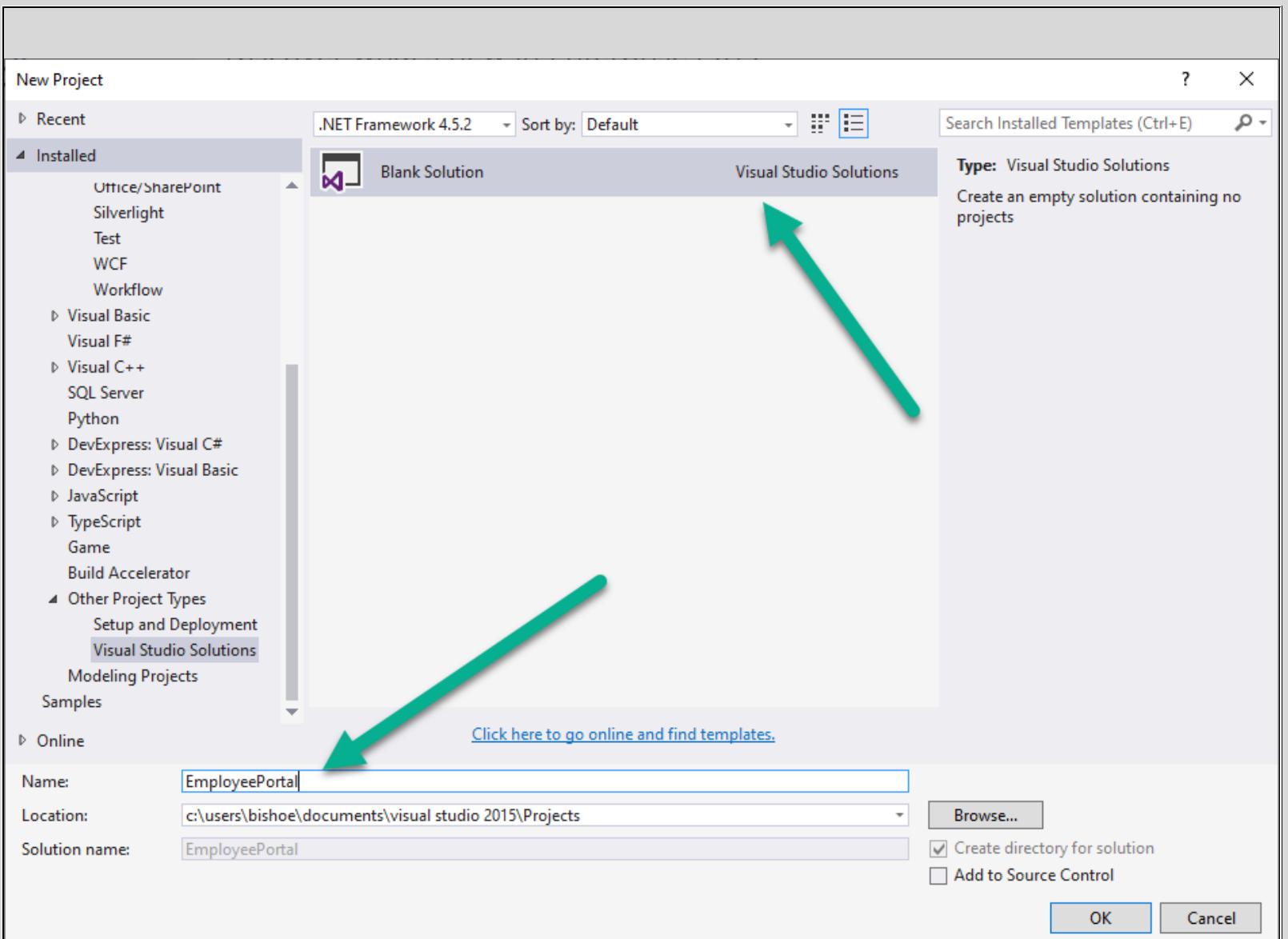
Suggested Videos :

Part 1 to 6 Design Patterns Tutorial

Link to Dot Net Basics, C#, LINQ,SQL Server, ASP.NET, ADO.NET and EF video series
<https://www.youtube.com/c/kudvenkatarabic/playlists>

2

BISHOYNABIL



هنعمل مثال لتسجيل الدخول باستخدام ال
MVC
And
Singleton

Add New Project

.NET Framework 4.5.2 Sort by: Default

Search Installed Templates (Ctrl+E)

Type: Visual C#
A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Application Insights
 Add Application Insights to Project
Help you understand and optimize your application.
[Learn more](#) [Privacy statement](#)

Recent

Installed

Visual C# Windows Web Android Cloud DXCore Extensibility iOS LightSwitch Office/SharePoint Silverlight Test WCF Workflow

Visual Basic Visual F#

Visual C++ SQL Server Python

DevExpress: Visual C# DevExpress: Visual Basic JavaScript TypeScript

Online

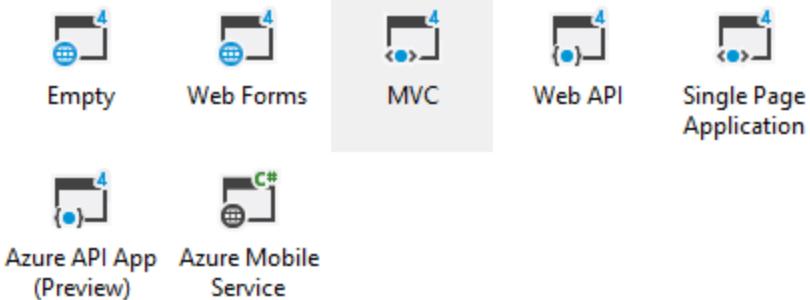
Click here to go online and find templates.

Name:

Location:

OK Cancel

Select a template:

ASP.NET 4.5.2 Templates**ASP.NET 5 Templates**

Add folders and core references for:

 Web Forms MVC Web API Add unit testsTest project name:

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

[Learn more](#)[Change Authentication](#)Authentication: **Individual User Accounts**

Microsoft Azure

 Host in the cloud

Add New Item - Web

?

X

Installed

Sort by: Default



Search Installed Templates (Ctrl+E)



Visual C#

Code

Data

General

Web

General

Markup

MVC

Razor

Scripts

SignalR

Web API

Web Forms

Windows Forms

WPF

DevExpress

DXCore

Reporting

Silverlight

SQL Server

Workflow

Online



DevExpress v15.2 ORM Persistent Object

Visual C#



ADO.NET Entity Data Model

Visual C#



DataSet

Visual C#



EF 5.x DbContext Generator

Visual C#



EF 6.x DbContext Generator

Visual C#



LINQ to SQL Classes

Visual C#



SQL Server Database

Visual C#



XML File

Visual C#



XML Schema

Visual C#



XSLT File

Visual C#



DevExpress ORM Data Model Wizard

Visual C#

[Click here to go online and find templates.](#)

Name:

EmployeePortal

Add

Cancel

The screenshot shows the Microsoft Visual Studio interface with the 'EmployeePortal - Microsoft Visual Studio' project open. In the center, the 'Table Designer' is displayed for a table named 'Table'. The 'Properties' window on the right shows the 'Id' column configuration, where the 'Is Identity' property is highlighted with a green arrow. A red arrow points to the 'Update' button in the toolbar at the top left.

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>	
JobDescription	varchar(50)	<input checked="" type="checkbox"/>	
Number	varchar(50)	<input checked="" type="checkbox"/>	
Department	varchar(50)	<input checked="" type="checkbox"/>	

```
1 CREATE TABLE [dbo].[Table]
2 (
3     [Id] INT NOT NULL PRIMARY KEY IDENTITY,
4     [Name] VARCHAR(50) NULL,
5     [JobDescription] VARCHAR(50) NULL,
6     [Number] VARCHAR(50) NULL,
7     [Department] VARCHAR(50) NULL
8 )
9
```

Properties

Id Column
(Name) Id
Allow Nulls False
Collation
Computed Column Specification
Data Type int
Default Value or Binding
Description
Full Text Specification
Identity Specification
(Is Identity) True
Identity Increment 1
Identity Seed 1
Is Column Set
Is File Stream
Is ROWGUID Column
Is Sparse
Not For Replication
Primary Key

(Is Identity)
Specifies whether the column is the identity column for the table.

بنعمل جدول ونخلص وندوس ابديت خلى بالك هنغير اسم الجدول من

Table

إلى

Employee

Add New Item - Web

Sort by: Default Search Installed Templates (Ctrl+E)

Type: Visual C#
A project item for creating an ADO.NET Entity Data Model.

	DevExpress v15.2 ORM Persistent Object	Visual C#
	ADO.NET Entity Data Model	Visual C# (highlighted)
	DataSet	Visual C#
	EF 5.x DbContext Generator	Visual C#
	EF 6.x DbContext Generator	Visual C#
	LINQ to SQL Classes	Visual C#
	SQL Server Database	Visual C#
	XML File	Visual C#
	XML Schema	Visual C#
	XSLT File	Visual C#
	DevExpress ORM Data Model Wizard	Visual C#

[Click here to go online and find templates.](#)

Name:

بنصيفها داخل ال
Models
وبنصيف الجدول بتاعنا ونكمel

Add Scaffold

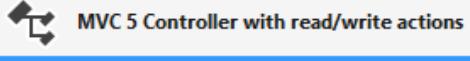
X

Installed

Common Controller



MVC 5 Controller - Empty



MVC 5 Controller with read/write actions



MVC 5 Controller with views, using Entity Framework



Web API 2 Controller – Empty



Web API 2 Controller with actions, using Entity Framework



Web API 2 Controller with read/write actions



Web API 2 OData v3 Controller with actions, using Entity Framework



Web API 2 OData v3 Controller with read/write actions

MVC 5 Controller with views, using Entity Framework
by Microsoft
v5.0.0.0

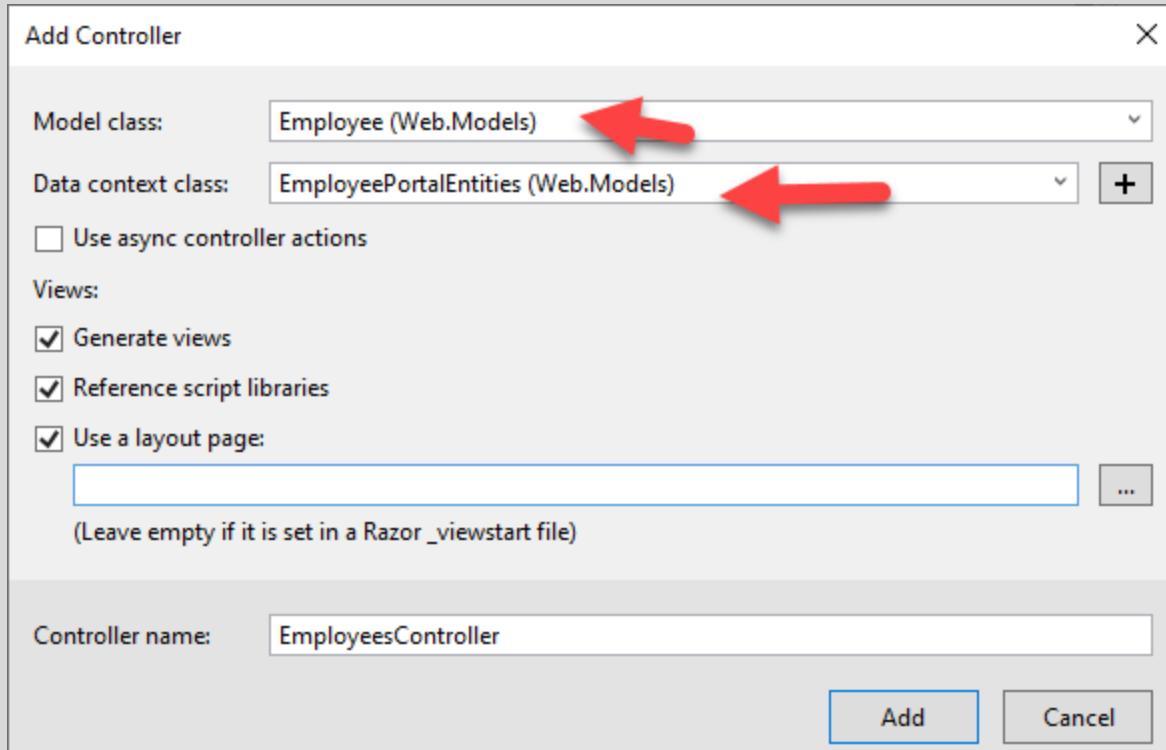
An MVC controller with actions and Razor views to create, read, update, delete, and list entities from an Entity Framework data context.

Id: MvcControllerWithContextScaffolder

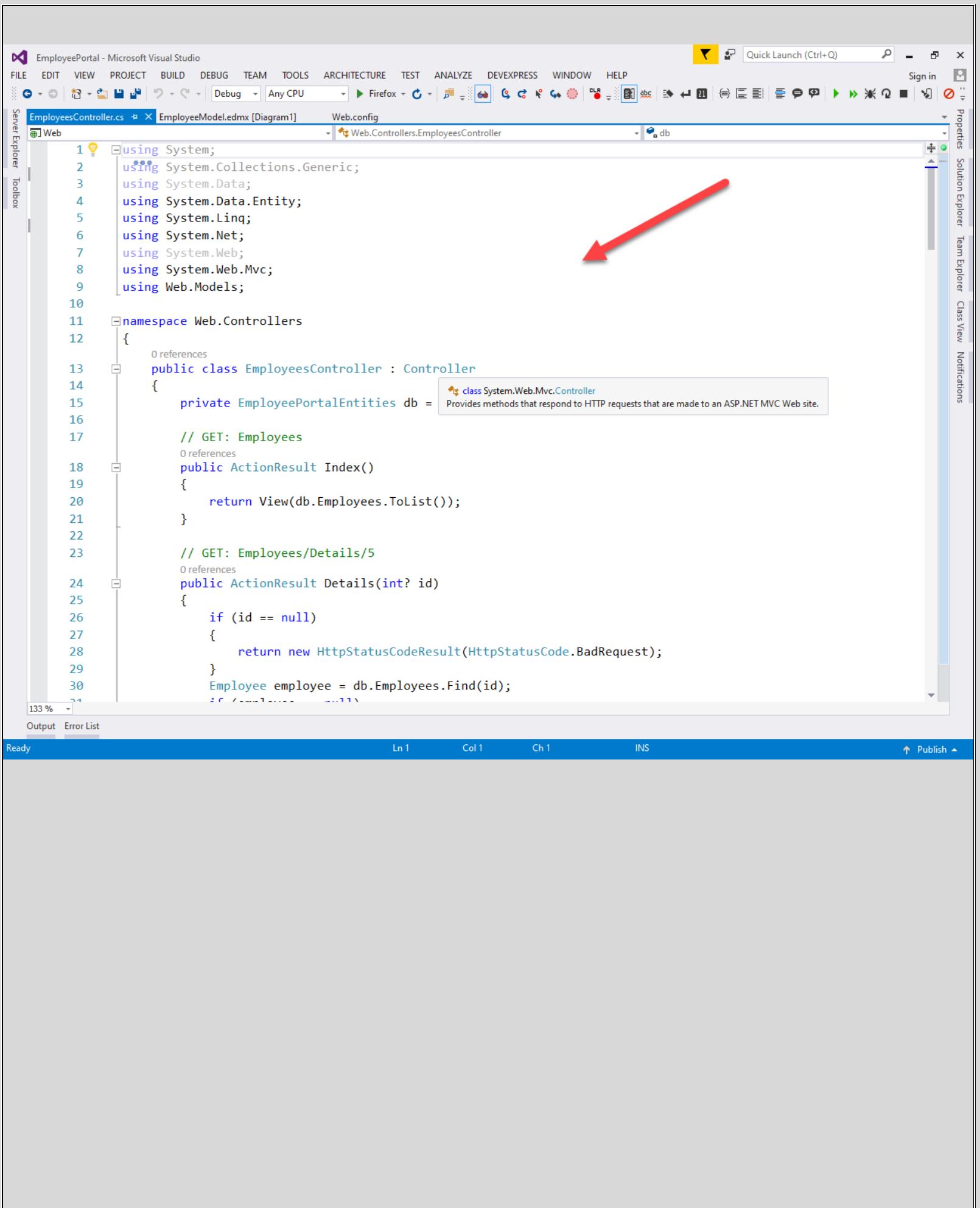
[Click here to go online and find more scaffolding extensions.](#)

Add

Cancel



BISHOYNABIL



EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Debug Any CPU Firefox CLR abc

EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config Web.Controllers.EmployeesController db

Properties Solution Explorer Team Explorer Class View Notifications

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.Linq;
6 using System.Net;
7 using System.Web;
8 using System.Web.Mvc;
9 using Web.Models;
10
11 namespace Web.Controllers
12 {
13     public class EmployeesController : Controller
14     {
15         private EmployeePortalEntities db = new EmployeePortalEntities();
16
17         // GET: Employees
18         public ActionResult Index()
19         {
20             return View(db.Employees.ToList());
21         }
22
23         // GET: Employees/Details/5
24         public ActionResult Details(int? id)
25         {
26             if (id == null)
27             {
28                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
29             }
30             Employee employee = db.Employees.Find(id);
31         }
32     }
33 }
```

133 %

Output Error List

Ready Ln 1 Col 1 Ch 1 INS ↑ Publish ▾

File Edit View History Bookmarks Tools Help

Top 10 C# Design | Design Patterns tu | Introduction to De | DownSub.com | project c# design | Learn C# Design P | Learn C# Design P | WhatsApp | Index - My ASP X

localhost:49936/Employees ... Search

Most Visited Getting Started Design Patterns in C# ...

Application name Home About Contact Register Log in

Index

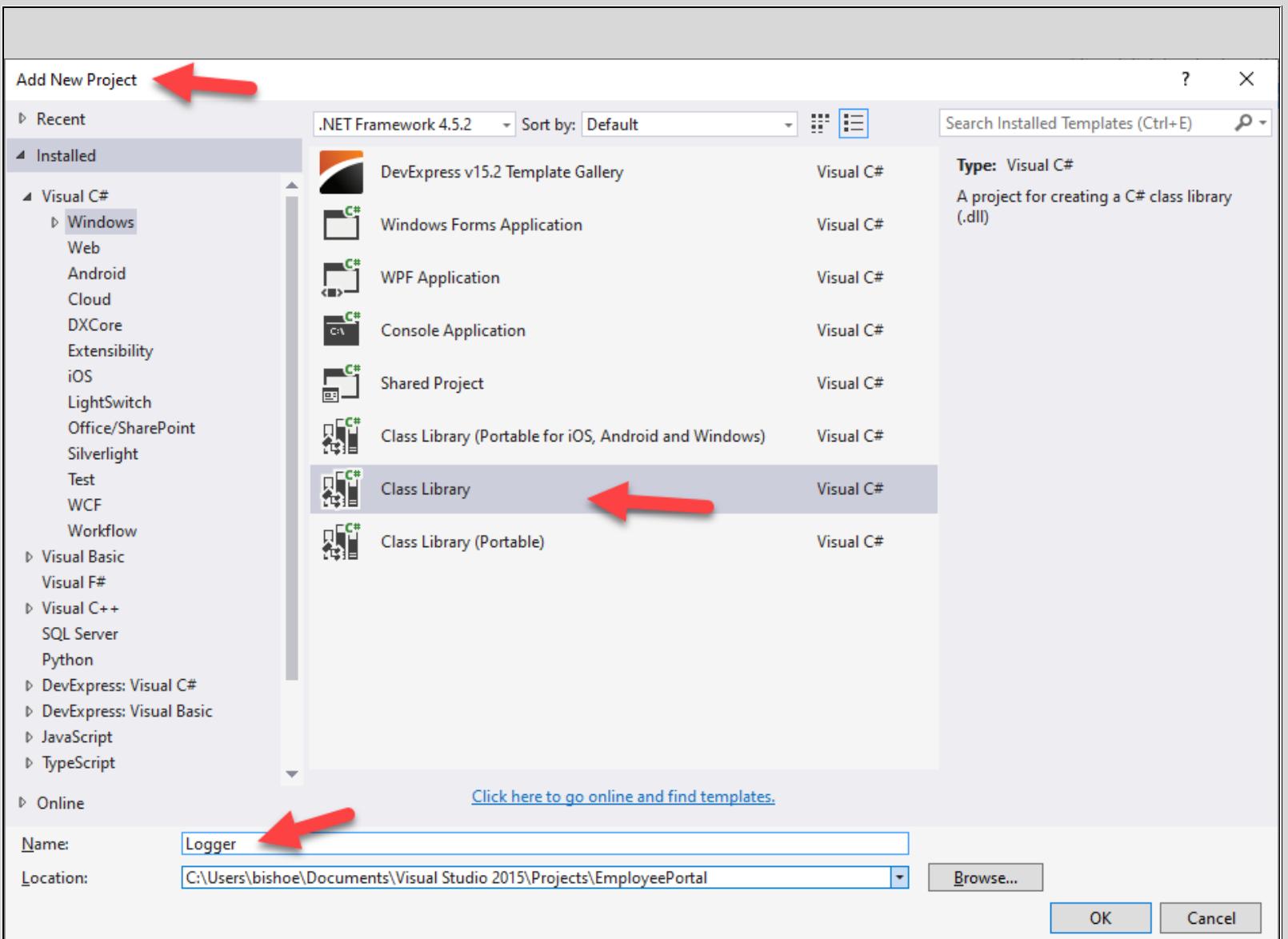
Create New

Name	JobDescription	Number	Department
bishoe	.net	1	.net

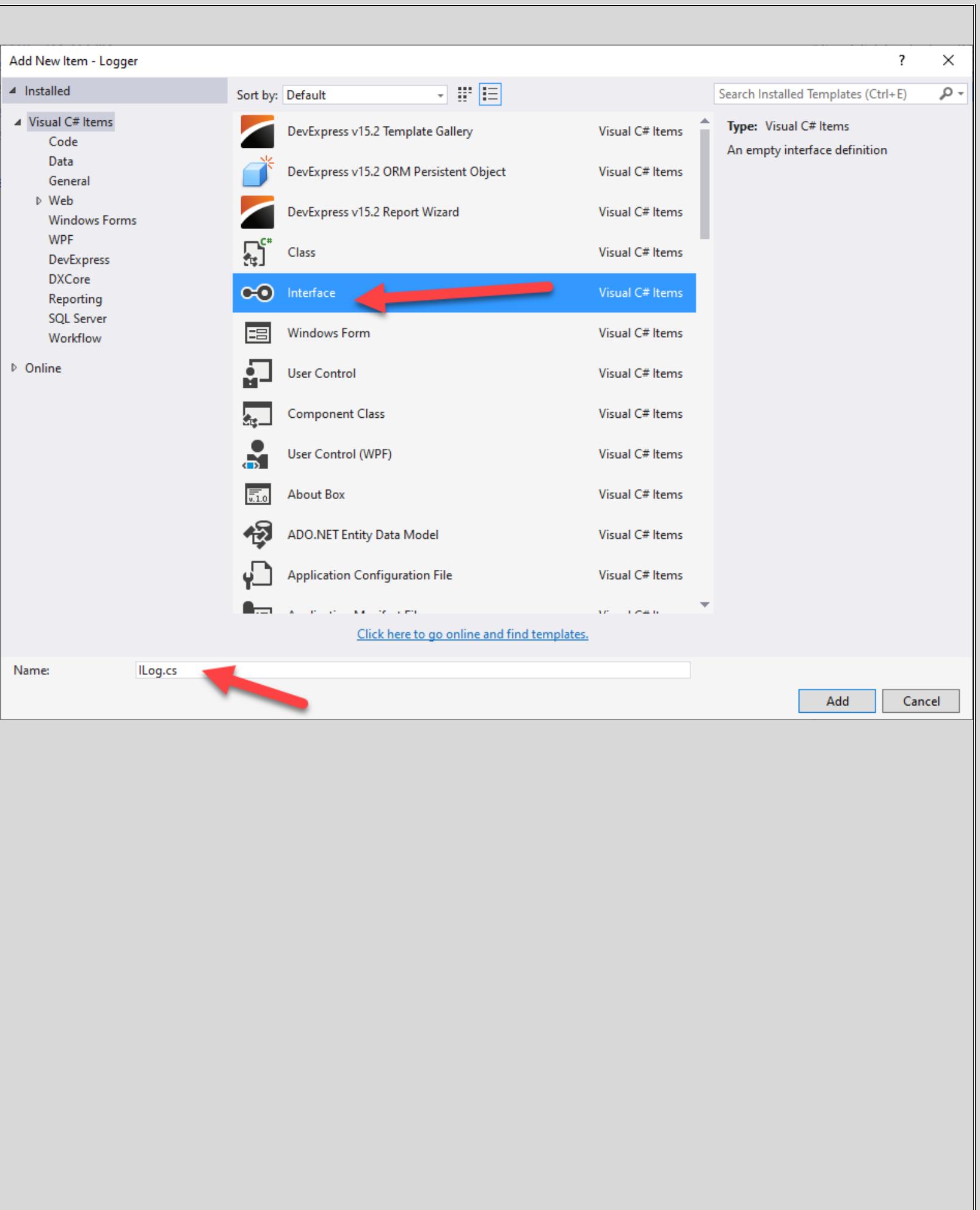
Edit | Details | Delete

© 2019 - My ASP.NET Application

بضيف موظف جديد
ونكمل



بضم الف م مشروع جديد
عشان نعمل
تسجيل الدخول بال
Singleton



EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP

Debug Any CPU Firefox CLR abc

ILog.cs Class1.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config

Logger

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Logger
8  {
9      interface ILog
10     {
11     }
12 }
13
```

Properties Solution Explorer Team Explorer Class View Notifications

133 %

Output Error List

Item(s) Saved

Ln 13 Col 1 Ch 1 INS ↑ Publish ▾

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Logger
8  {
9      interface ILog
10     {
11     }
12 }
13
```

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP

Debug Any CPU Firefox

ILog.cs Class1.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config

Server Explorer Toolbox

Logger

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Logger
8  {
9      //public class Class1
10     //{
11     //}
12 }
13
14
```

Solution Explorer

Search Solution Explorer (Ctrl+.)

Solution 'EmployeePortal' (2 projects)

- Logger
 - Properties
 - References
 - Class1.cs
 - ILog.cs
- Web

133 %

Output Error List

Ready

Ln 12 Col 5 Ch 5 INS ↑ Publish ▾

The screenshot shows the Microsoft Visual Studio interface. In the center-left, there is a code editor window displaying a C# file named 'Class1.cs'. The code contains several using statements at the top, followed by a namespace declaration 'namespace Logger'. Inside the namespace, there is a single line of code starting with 'public class Class1' which is immediately followed by a brace '}'. Below this, there is another brace '}' on its own line. A red arrow points to this second brace. In the bottom right corner of the code editor, there are status indicators: 'Ln 12', 'Col 5', 'Ch 5', 'INS', and an upward arrow followed by 'Publish ▾'. To the right of the code editor is the 'Solution Explorer' window, which lists two projects: 'Logger' and 'Web'. Under the 'Logger' project, 'Properties', 'References', and 'Class1.cs' are visible. Another red arrow points to 'Class1.cs'. At the very bottom of the screen, there is a toolbar with icons for 'Output' and 'Error List', and a status bar indicating 'Ready'.

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP
- Toolbars:** Standard, Debug, Task List, Solution Explorer, Properties, Task List, Status Bar.
- Solution Explorer:** Shows 'Solution 'EmployeePortal' (2 projects)' with 'Logger' selected, containing 'Properties', 'References', 'Class1.cs', 'ILog.cs', and 'Web'.
- Code Editor:** Displays the 'Class1.cs' file with the following code:

```
1 using System.Linq;
2 using System.Text;
3 using System.Threading.Tasks;
4
5 namespace Logger
6 {
7     //public class Class1
8     //{
9     //}
10    5 references
11    public sealed class Log
12    {
13        private static int counter = 0;
14        private static readonly object obj = new object();
15
16        private static readonly Lazy<Log> instance = new Lazy<Log>(() => new Log());
17
18        0 references
19        public static Log GetInstance
20        {
21            get
22            {
23
24                return instance.Value;
25            }
26        }
27        1 reference
28        private Log()
29        {
30            counter++;
31            Console.WriteLine("counter Value" + counter.ToString());
32        }
33    }
34}
```
- Annotations:** Two red arrows point to the 'Log' class definition and the 'GetInstance' property.
- Status Bar:** Item(s) Saved, Ln 13, Col 6, Ch 6, INS.

ده كان كلاس قديم اللي استخدمناه في الشرح قبل كده جبناه وغيرت اسمه

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Solution Explorer:** Shows the solution 'EmployeePortal' with two projects: 'Logger' and 'Web'. 'Logger' contains files 'Properties', 'References', 'Class1.cs', and 'ILog.cs'. 'Web' is also listed.
- Properties Window:** Located on the right side of the interface.
- Code Editor:** Displays the 'ILog.cs' file with the following C# code:

```
using System.Text;
using System.Threading.Tasks;

namespace Logger
{
    //public class Class1
    //{
    //}

    5 references
    public sealed class Log
    {
        //private static int counter = 0;
        //private static readonly object obj = new object();

        private static readonly Lazy<Log> instance = new Lazy<Log>(() => new Log());

        0 references
        public static Log GetInstance
        {
            get
            {
                return instance.Value;
            }
        }

        1 reference
        private Log()
        {
            //counter++;
            //Console.WriteLine("counter Value" + counter.ToString());
        }

        //public void printDetails(string message)
        //{
        //    Console.WriteLine(message);
        //}
    }
}
```
- Status Bar:** Shows '110 %' zoom level, 'Output' tab selected, 'Ready' status, and file navigation information (Ln 24, Col 39, Ch 39, INS).
- Bottom Status:** Includes 'Publish' and other system icons.

Text Overlay: عطلت منه كل الاكواب اللى مش محتاجها

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays a C# code editor with the file `ILog.cs` open. The code defines an interface `ILog` with a single method `LogException(string message)`. The code is as follows:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Logger
8  {
9      interface ILog
10     {
11         void LogException(string message);
12     }
13 }
14
```

A red arrow points from the line `interface ILog` to the line `void LogException(string message);`, indicating a reference or relationship between the interface and its implementation.

The Solution Explorer on the right shows two projects: `Logger` and `Web`. The `Logger` project contains `Class1.cs` and `ILog.cs`.

بنعمل ميثود في الانترفيس وبتاخد رسالى الايرور

The screenshot shows the Microsoft Visual Studio interface with the code editor open. The code editor displays a C# file named `ILog.cs`. A red arrow points to a context menu that has appeared over the line of code `public sealed class Log : ILog`. The context menu is titled "Refactor" and includes options such as "Rename", "Code", "Create Implementer (implicit)", "Create Implementer (explicit)", and "Implement IDisposable". The "Implement IDisposable" option is highlighted with a blue background. The Solution Explorer and Properties windows are visible on the right side of the IDE.

```
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Logger
8  {
9      5 references
10     public sealed class Log : ILog
11     {
12         {
13             private static readonly Lazy<Log> () => new
14
15             0 references
16             public static Log GetI
17             {
18                 get
19                 {
20                     return instance.Value;
21                 }
22             }
23         }
24         1 reference
25         private Log()
26         {
27     }
28
29 }
```

بنعمل ايمپلمنت للانترفيس

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP

Debug Any CPU Firefox

Server Explorer Toolbox

ILog.cs Class1.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config

Logger

```
10
11
12
13
14
15
16
17     get
18     {
19
20         return instance.Value;
21     }
22
23 }
24 private Log()
25 {
26
27 }
28
29 void ILog.LogException(string message)
30 {
31     throw new NotImplementedException();
32 }
33 ...
34 }
```

1 reference

1 reference

Solution Explorer Properties References Class1.cs ILog.cs Web

146 %

Output Error List

Ready Ln 32 Col 10 Ch 10 INS Publish

هنسخ الله بداخله

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEPRESS WINDOW HELP
Debug Any CPU Firefox
Server Explorer Toolbox
ILog.cs Class1.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config
Logger.cs
24 }
25     1 reference
26     private Log()
27     {
28
29     }
30     1 reference
31     void ILog.LogException(string message)
32     {
33         string fileName = string.Format("{0}_{1}.log", "Exception",
34             DateTime.Now.ToString("yyyy-MM-dd"));
35
36         string logFilePath = string.Format(@"{0}\{1}", AppDomain.CurrentDomain.BaseDirectory,
37             fileName);
38
39         StringBuilder sb = new StringBuilder();
40         sb.AppendLine("-----");
41         sb.AppendLine(DateTime.Now.ToString());
42
43         sb.AppendLine(message);
44
45         using (StreamWriter writer = new StreamWriter(logFilePath))
46         {
47             writer.WriteLine(sb.ToString());
48             writer.Flush();
49         }
    }

Properties Solution Explorer Team Explorer Class View Notifications
```

The screenshot shows a Microsoft Visual Studio interface with the code editor open. The code is a C# class named 'Logger' that implements the 'ILog' interface. Red arrows point to two specific lines of code: one pointing to the start of the 'LogException' method body and another pointing to the closing brace of the entire class definition.

كود مبسط بنكريت ملف

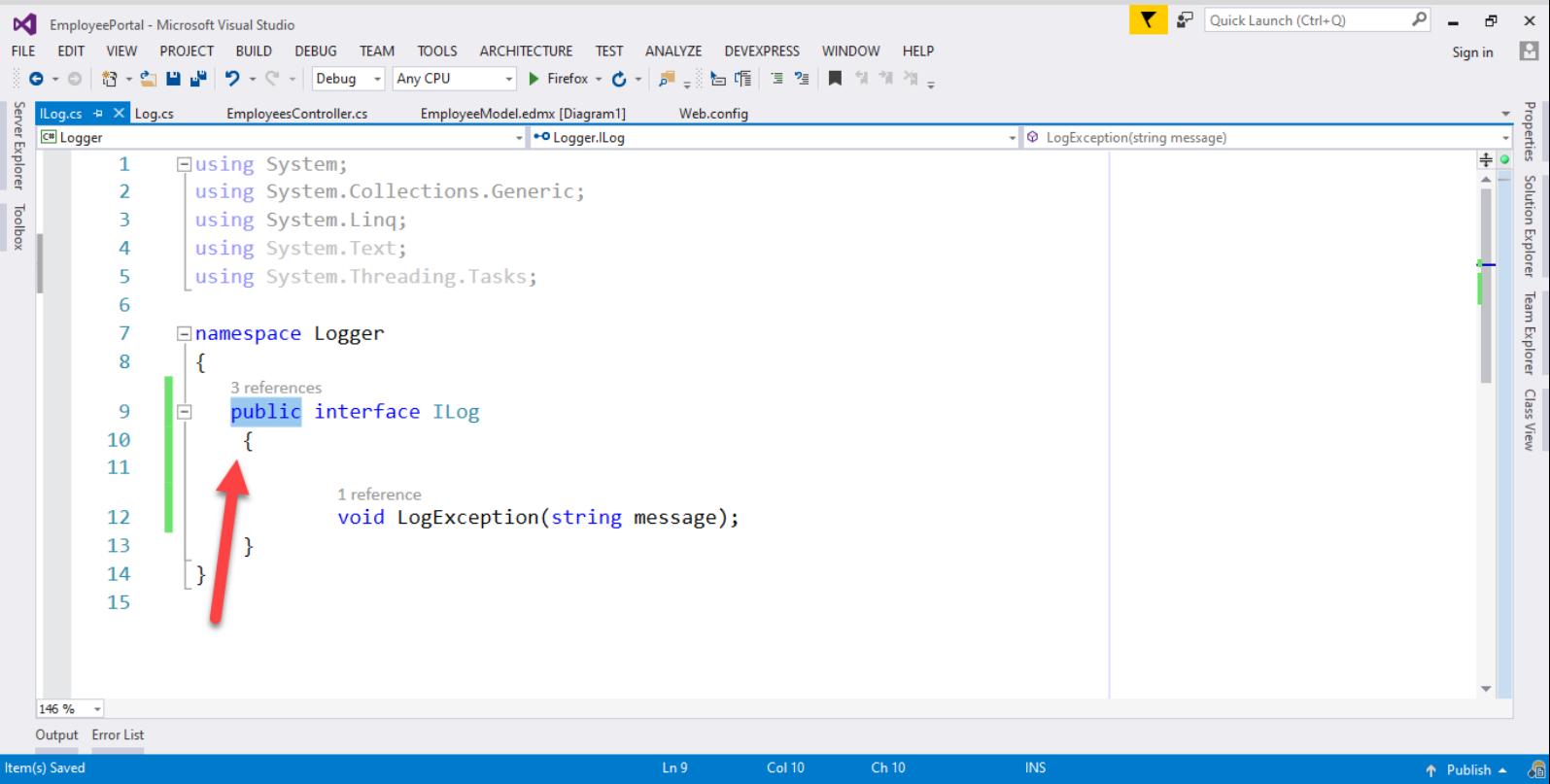
Log

بحث فيه كل الایرور

نكملي بغير اسم الكلاس ال

Log

ونروح على الانترفيس



The screenshot shows the Microsoft Visual Studio IDE interface. The title bar reads "EmployeePortal - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, DEVEXPRESS, WINDOW, and HELP. The toolbar has various icons for file operations like Open, Save, and Print. The status bar at the bottom shows "Item(s) Saved", "Ln 9", "Col 10", "Ch 10", and "INS". The code editor displays the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Logger
8  {
9      public interface ILog
10     {
11         void LogException(string message);
12     }
13 }
14
15
```

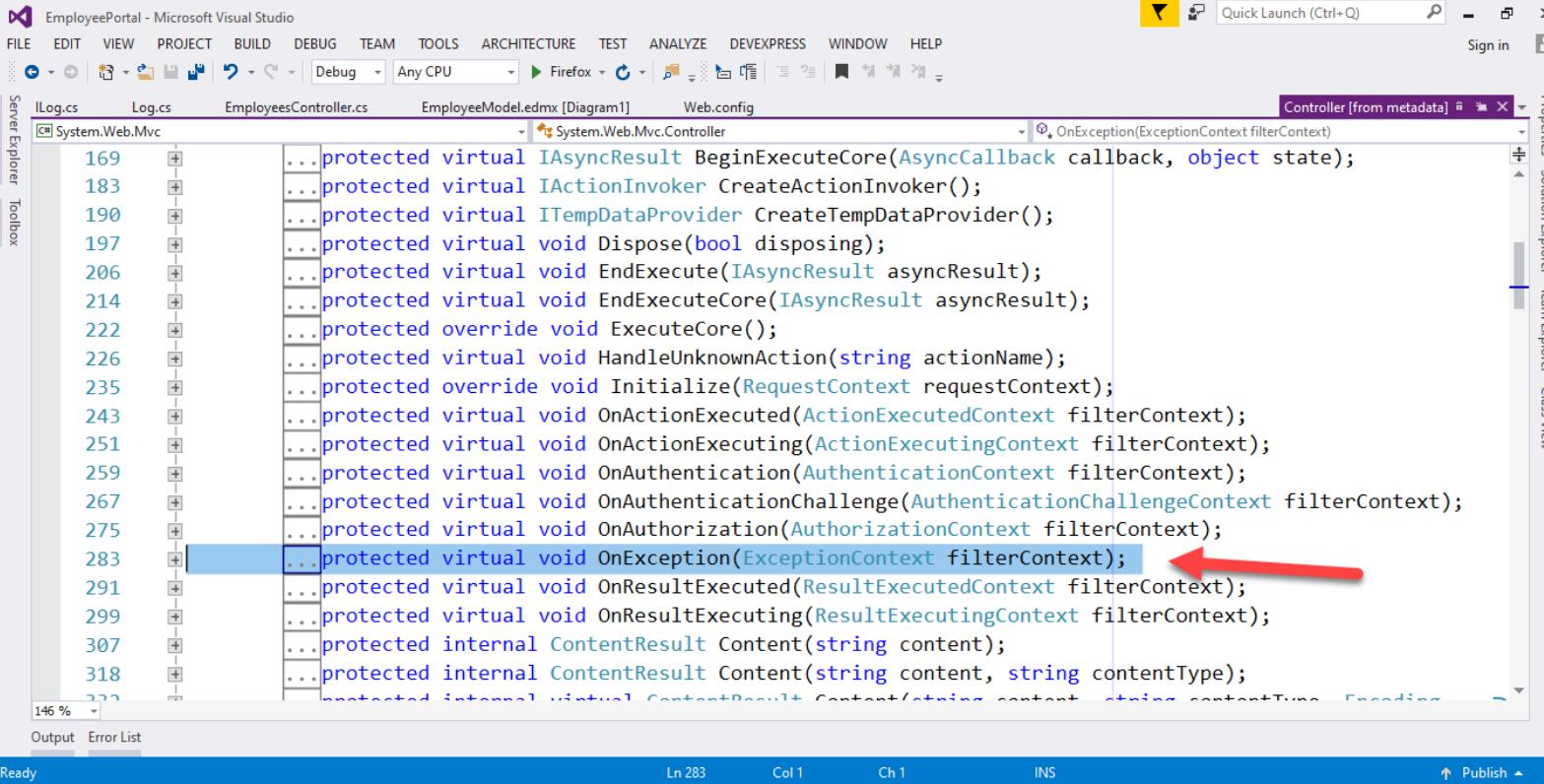
A red arrow points to the line "public interface ILog" in the code editor. The code editor also highlights "Logger" in the namespace and "ILog" in the interface definition. The status bar indicates the current line is 9, column 10.

بنعملها

Public

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
Quick Launch (Ctrl+Q) Sign in
ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config
J Web Web.Controllers.EmployeesController EmployeesController()
10  using Web.Models;
11
12  namespace Web.Controllers
13  {
14      public class EmployeesController : Controller
15      {
16          private ILog _ILog;
17
18          private EmployeePortalEntities db = new EmployeePortalEntities();
19
20          public EmployeesController()
21          {
22              _ILog = Log.GetInstance;
23          }
24
25
26          // GET: Employees
146 %
Output Error List
Ready Ln 23 Col 10 Ch 10 INS Publish
```

BISHOY NABIL



The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, DEVEXPRESS, WINDOW, HELP
- Toolbar:** Standard icons for file operations.
- Quick Launch:** Quick Launch (Ctrl+Q) button.
- User Account:** Sign in button.
- Code Editor:** The main window displays the source code for `System.Web.Mvc.Controller`. The code includes various protected virtual methods such as `BeginExecuteCore`, `CreateActionInvoker`, `CreateTempDataProvider`, `Dispose`, `EndExecute`, `EndExecuteCore`, `ExecuteCore`, `HandleUnknownAction`, `Initialize`, `OnActionExecuted`, `OnActionExecuting`, `OnAuthentication`, `OnAuthenticationChallenge`, `OnAuthorization`, `OnException`, `OnResultExecuted`, and `OnResultExecuting`. The line `...protected virtual void OnException(ExceptionContext filterContext);` is highlighted with a blue selection bar and has a red arrow pointing to it from the right.
- Toolbars:** Server Explorer, Toolbox, Properties, Solution Explorer, Team Explorer, Class View.
- Status Bar:** Shows 146 % zoom, Ln 283, Col 1, Ch 1, INS.
- Output Window:** Ready.

هنا دا الكترولار فى ميثود فيه خاصة بالاكسيشن تعالى نعملها Override

ونستخدمها فى الكترولار بتاعنا

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEPRESS WINDOW HELP
ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config
Sign in
Controller [from metadata]
OnException(ExceptionContext filterContext)
Web.Controllers.EmployeesController
17     private ILog _ILog;
18
19     private EmployeePortalEntities db = new EmployeePortalEntities();
20     public EmployeesController()
21     {
22         ...
23     }
24
25     protected override void OnException(ExceptionContext filterContext)
26     {
27         _ILog.LogError(filterContext.Exception.ToString());
28     }
29
30
31
32     // GET: Employees
33     public ActionResult Index()
34     {
35         return View(db.Employees.ToList());
36     }

```

هنا بعمل اوفر رايد للميثود
وبنزل تحت بقوله هات الايرور اللى هيطلع وهىكون متخزن فى
Filtercontext
بحببه واحطه داخل الميثود بتاعتي
ونكمل

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEXPRESS WINDOW HELP
Debug Any CPU Firefox Web.config
ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Controller [from metadata]
OnException(ExceptionContext filterContext)

17     private ILog _ILog;
18
19     private EmployeePortalEntities db = new EmployeePortalEntities();
20
21     public EmployeesController()
22     {
23
24         protected override void OnException(ExceptionContext filterContext)
25         {
26             _ILog.LogException(filterContext.Exception.ToString());
27             filterContext.ExceptionHandled = true;
28             this.View("Error").ExecuteResult(this.ControllerContext);
29
30         }
31
32
33
34     // GET: Employees
35     public ActionResult Index()
36     {
```

بنزل تحت افعلن انه يهندل الايرور
وتحته بقوله لما يطلع ايرور حوله على الفيو اللي اسمه
Error

Screenshot of a web browser showing an ASP.NET application's 'Create' page for an 'Employee'. The page has fields for Name (MAnb), JobDescription (Hr), Number (01), and Department. A green arrow points to the 'Department' field. Below the form is a 'Create' button and a 'Back to List' link.

هنا لو سیته فاصلی هیطلع اکسپلورر تعالی ندوس ونجرب ونشوف هیحصل ایه

Screenshot of SQL Server Management Studio (SSMS) showing the 'EMPLOYEE' table design. The 'Department' column is highlighted with a blue selection bar. A green arrow points from the 'Department' field in the previous screenshot to this selection bar. The table structure includes columns: Id, Name, JobDescription, Number, and Department. The 'Department' column is defined as varchar(50) and allows nulls. The table also has a primary key constraint named <unnamed>. The T-SQL code for creating the table is visible at the bottom.

```

CREATE TABLE [dbo].[EMPLOYEE] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Name] VARCHAR (50) NOT NULL,
    [JobDescription] VARCHAR (50) NOT NULL,
    [Number] VARCHAR (50) NOT NULL,
    [Department] VARCHAR (50) NOT NULL,
)

```

بنعدل هنا مش بنسمح انه يقبل نصوص فارغه
وتعالی نجرب نضيف موظف تانى

File Edit View History Bookmarks Tools Help

Top 10 C# Design | Design Patterns | Introduction to D | DownSub.com | project c# design | Learn C# Design | Learn C# Design | WhatsApp

localhost:49936/Employees/Create

Most Visited Getting Started Design Patterns in C# ...

Application name Home About Contact Register Log in

Create

Employee

Name	MAnb
JobDescription	Hr
Number	AssF001
Department	<input type="text"/>

[Create](#)

[Back to List](#)

© 2019 - My ASP.NET Application

File Edit View History Bookmarks Tools Help

Top 10 C# Design | Design Patterns | Introduction to D | DownSub.com | project c# design | Learn C# Design | Learn C# Design | WhatsApp

localhost:49936/Employees/Create

Most Visited Getting Started Design Patterns in C# ...

Server Error in '/' Application.

Could not find a part of the path 'C:\Users\bishoe\Documents\Visual Studio 2015\Projects\EmployeePortal\Web\Exception_8\16\2019.log'.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.IO.DirectoryNotFoundException: Could not find a part of the path 'C:\Users\bishoe\Documents\Visual Studio 2015\Projects\EmployeePortal\Web\Exception_8\16\2019.log'.

Source Error:

```

Line 41:         sb.AppendLine(message);
Line 42:
Line 43:         using (StreamWriter writer = new StreamWriter(logFilePath))
Line 44:         {
Line 45:             writer.WriteLine(sb.ToString());

```

Source File: C:\Users\bishoe\Documents\Visual Studio 2015\Projects\EmployeePortal\Logger\Log.cs **Line:** 43

Stack Trace:

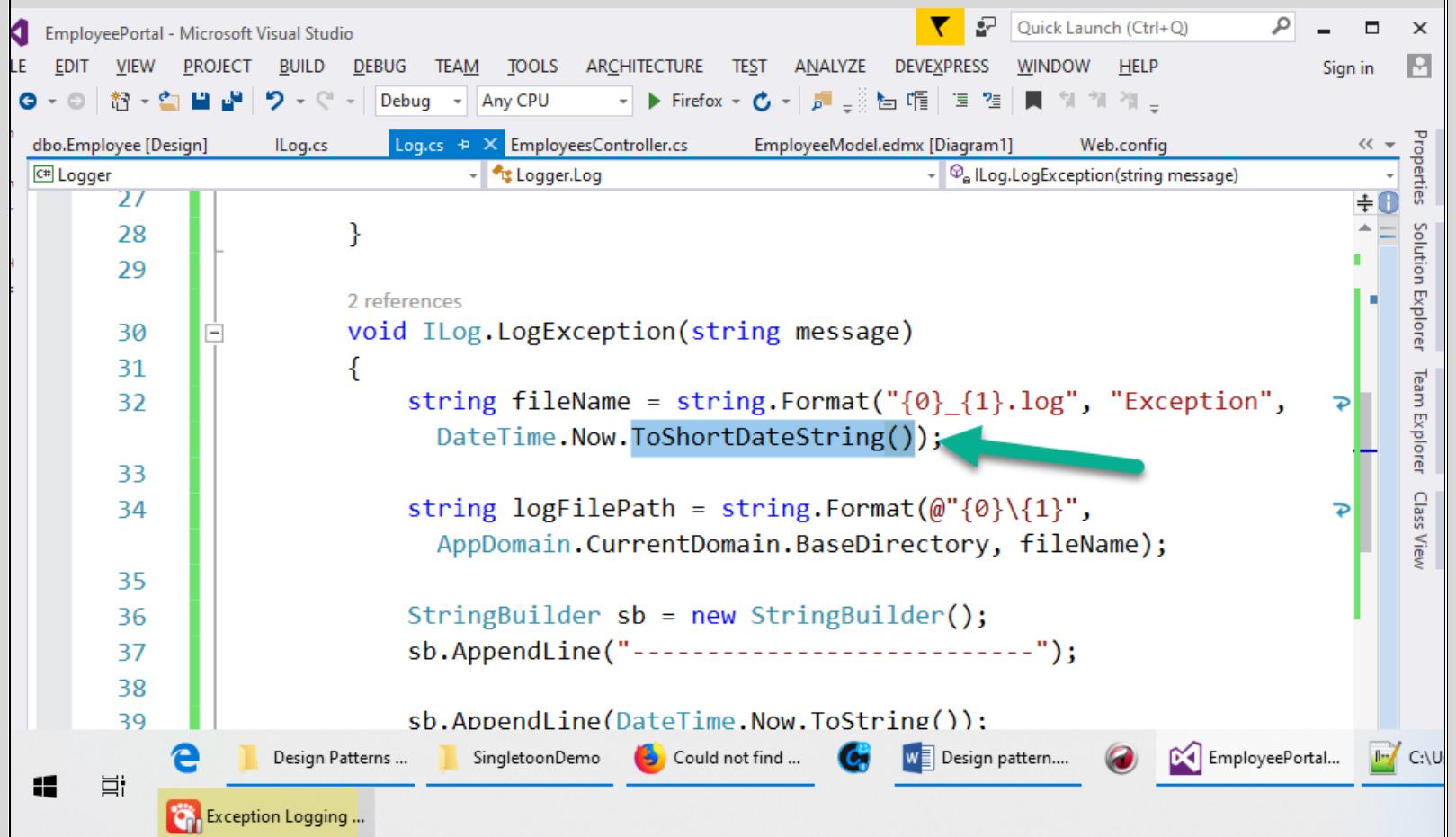
```

[FileNotFoundException: Could not find a part of the path 'C:\Users\bishoe\Documents\Visual Studio 2015\Projects\EmployeePortal\Web\Exception_8\16\2019.log'.]
System.IO._Error.WinIOError(Int32 errorCode, String maybeFullPath) +373
System.IO.FileStream.Init(String path, FileMode mode, FileAccess access, Int32 rights, Boolean useRights, FileShare share, Int32 bufferSize, FileOptions options, S
System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, FileOptions options, String msgPath, Boolean bFromProxy)
System.IO.StreamWriter.Createfile(String path, Boolean append, Boolean checkHost) +76
System.IO.StreamWriter..ctor(String path, Boolean append, Encoding encoding, Int32 bufferSize, Boolean checkHost) +75
System.IO.StreamWriter..ctor(String path) +36

```

مشكلة في البروجيكت هنعالجها ونكملي
 المشكلة في اسم الملف عملناه وبالشكل ده غلط مينفعش تحط

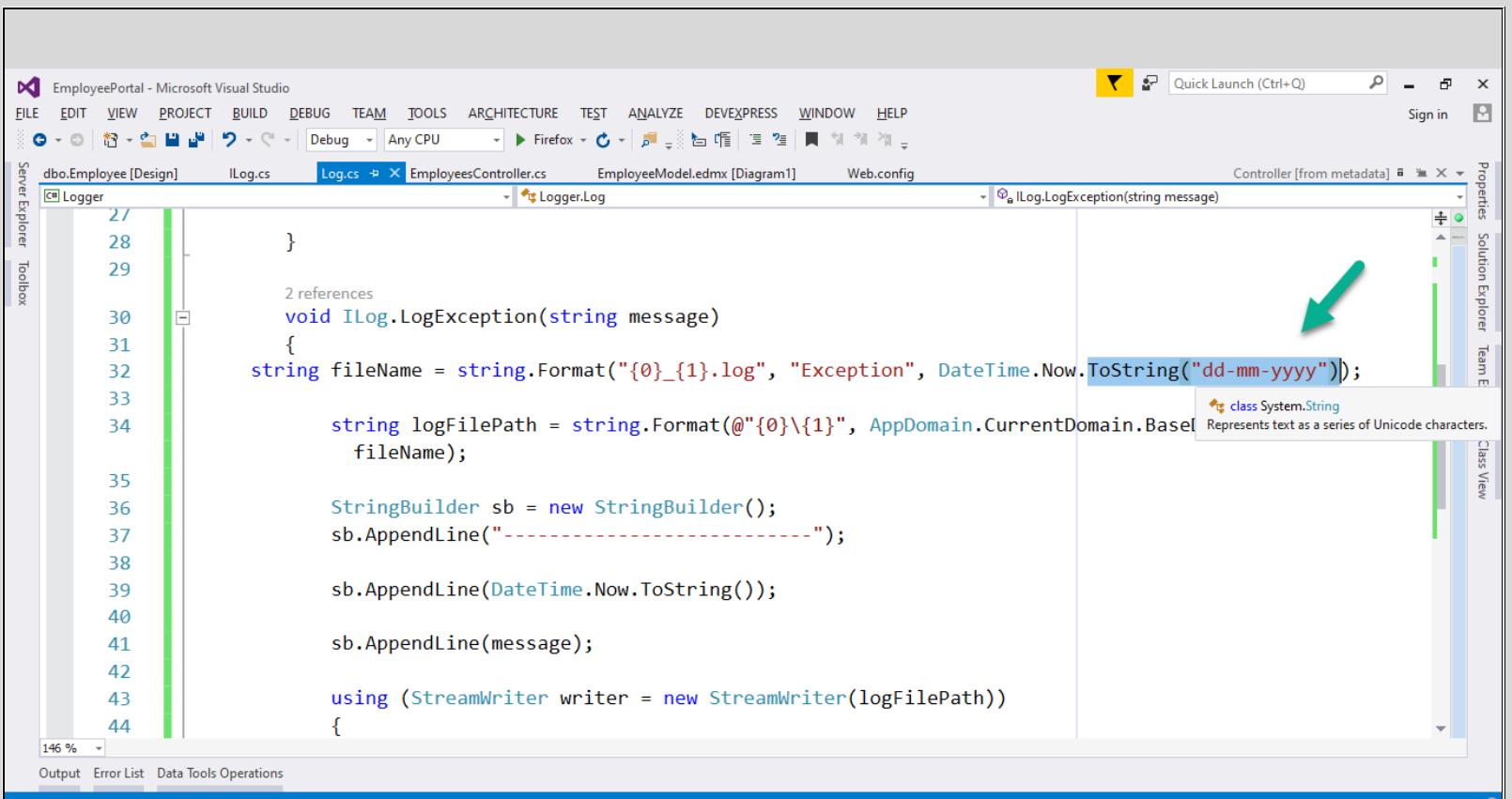
/
في اسم الملف فطلع ايور لازم نغير الاسم تعالى بص



The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEXPRESS WINDOW HELP
- Toolbar:** Standard toolbar with icons for file operations.
- Toolbox:** Standard toolbox with icons for various development tools.
- Code Editor:** Displays the `Logger.cs` file. The code defines a method `ILog.LogException(string message)` which formats a log file name using `DateTime.Now.ToShortDateString()`. A large green arrow points to this line of code.
- Solution Explorer:** Shows the project structure with files like `dbo.Employee [Design]`, `ILog.cs`, `Log.cs`, `EmployeesController.cs`, `EmployeeModel.edmx [Diagram1]`, and `Web.config`.
- Properties Window:** Standard properties window for the selected item.
- Task List:** Shows references and tasks.
- Status Bar:** Shows the current file as `Exception Logging ...`.

هنجيرها ب



EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Server Explorer Toolbox

dbo.Employee [Design] ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config Controller [from metadata]

Logger

```
27
28 }
29
30 void ILog.LogException(string message)
31 {
32     string fileName = string.Format("{0}_{1}.log", "Exception", DateTime.Now.ToString("dd-mm-yyyy"));
33
34     string filePath = string.Format(@"{0}\{1}", AppDomain.CurrentDomain.BaseName, fileName);
35
36     StringBuilder sb = new StringBuilder();
37     sb.AppendLine("-----");
38
39     sb.AppendLine(DateTime.Now.ToString());
40
41     sb.AppendLine(message);
42
43     using (StreamWriter writer = new StreamWriter(filePath))
44 {
```

2 references

System.String

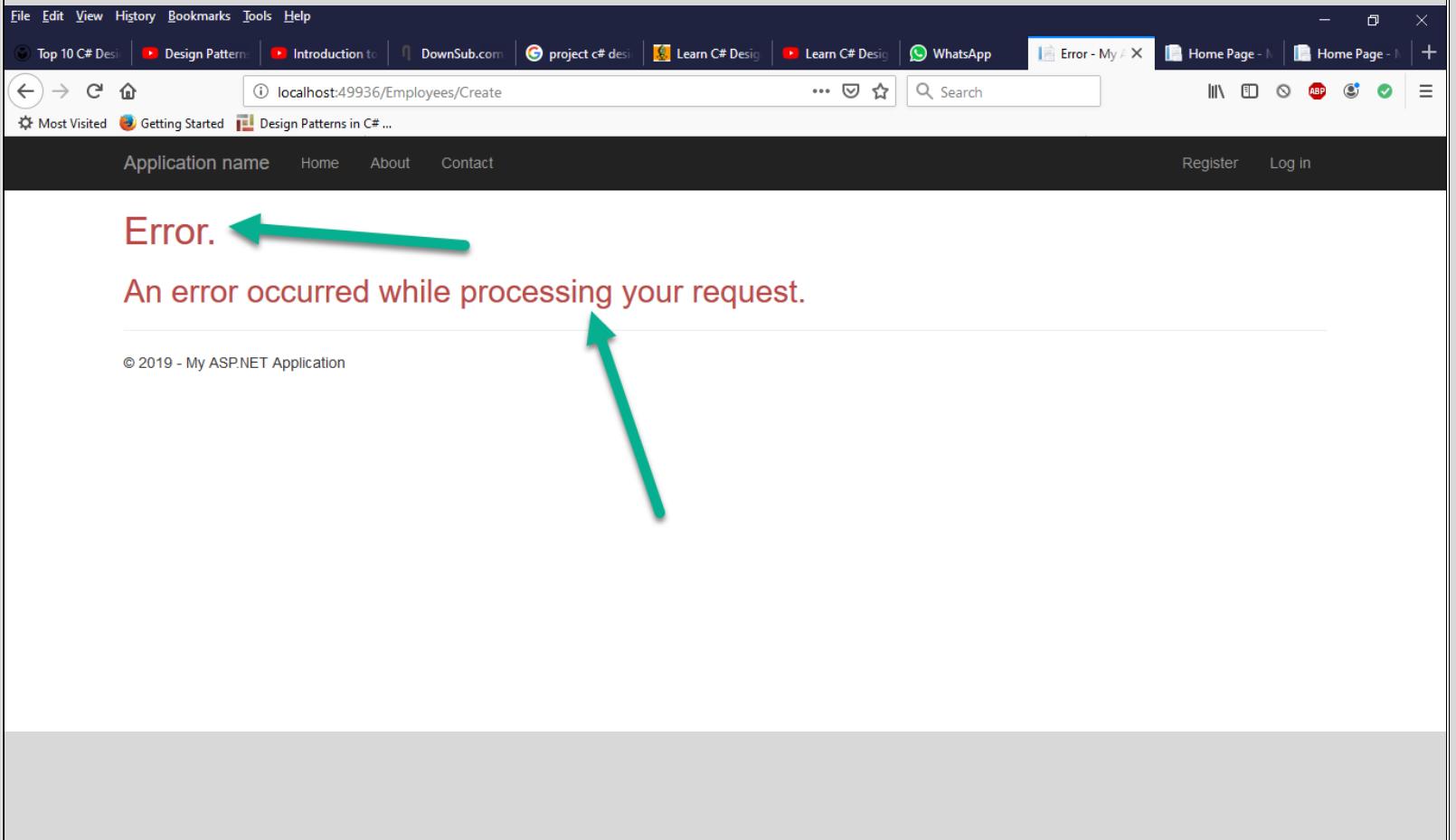
146 %

Output Error List Data Tools Operations

Item(s) Saved

Ln 32 Col 101 Ch 101 INS

BISHOYNABIL نجرب تانی



File Edit View History Bookmarks Tools Help

Top 10 C# Design Patterns Introduction to DownSub.com project c# design Learn C# Design Learn C# Design WhatsApp Error - My Application Home Page - Home Page -

localhost:49936/Employees/Create

Most Visited Getting Started Design Patterns in C# ...

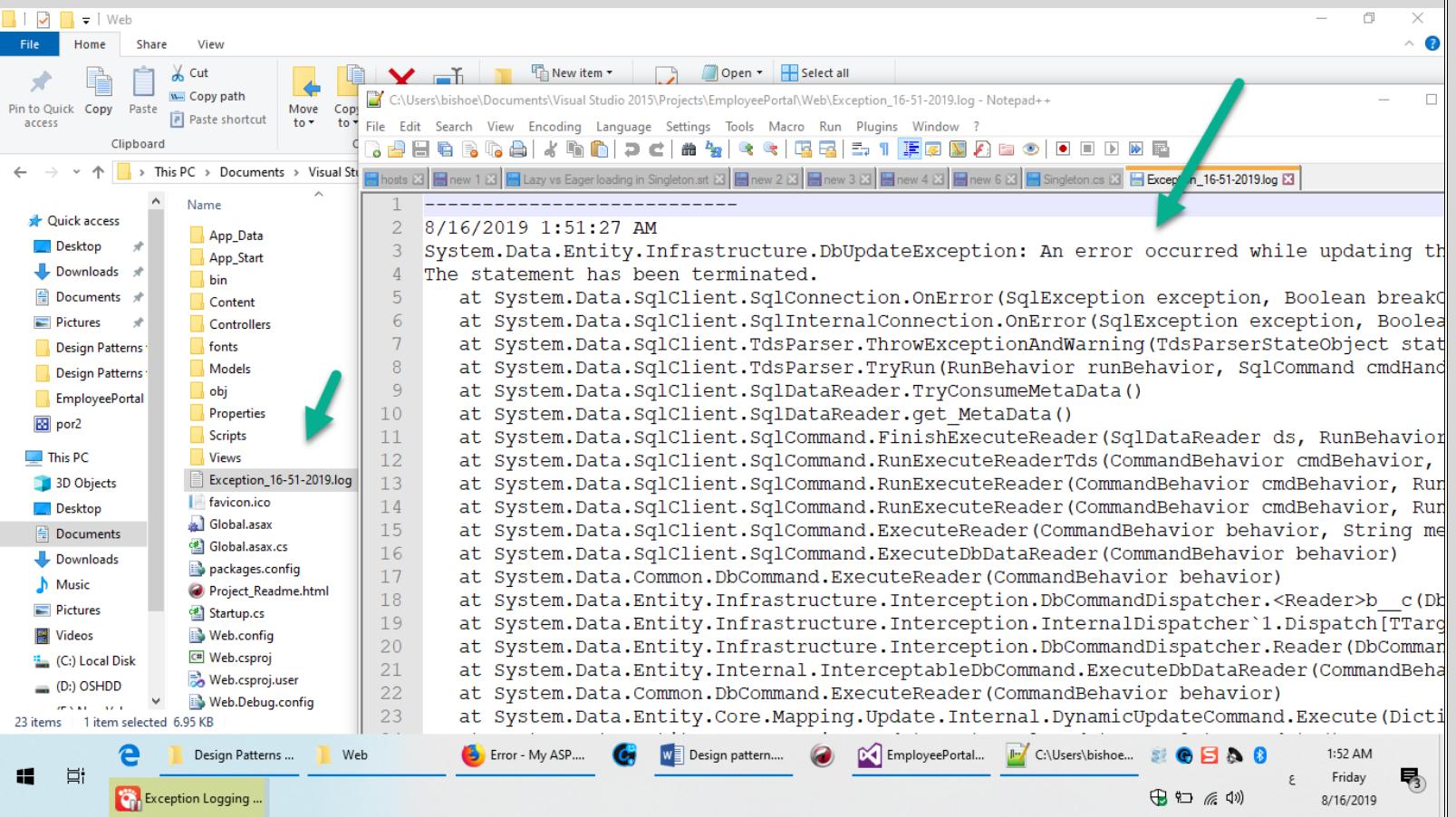
Application name Home About Contact Register Log in

Error.

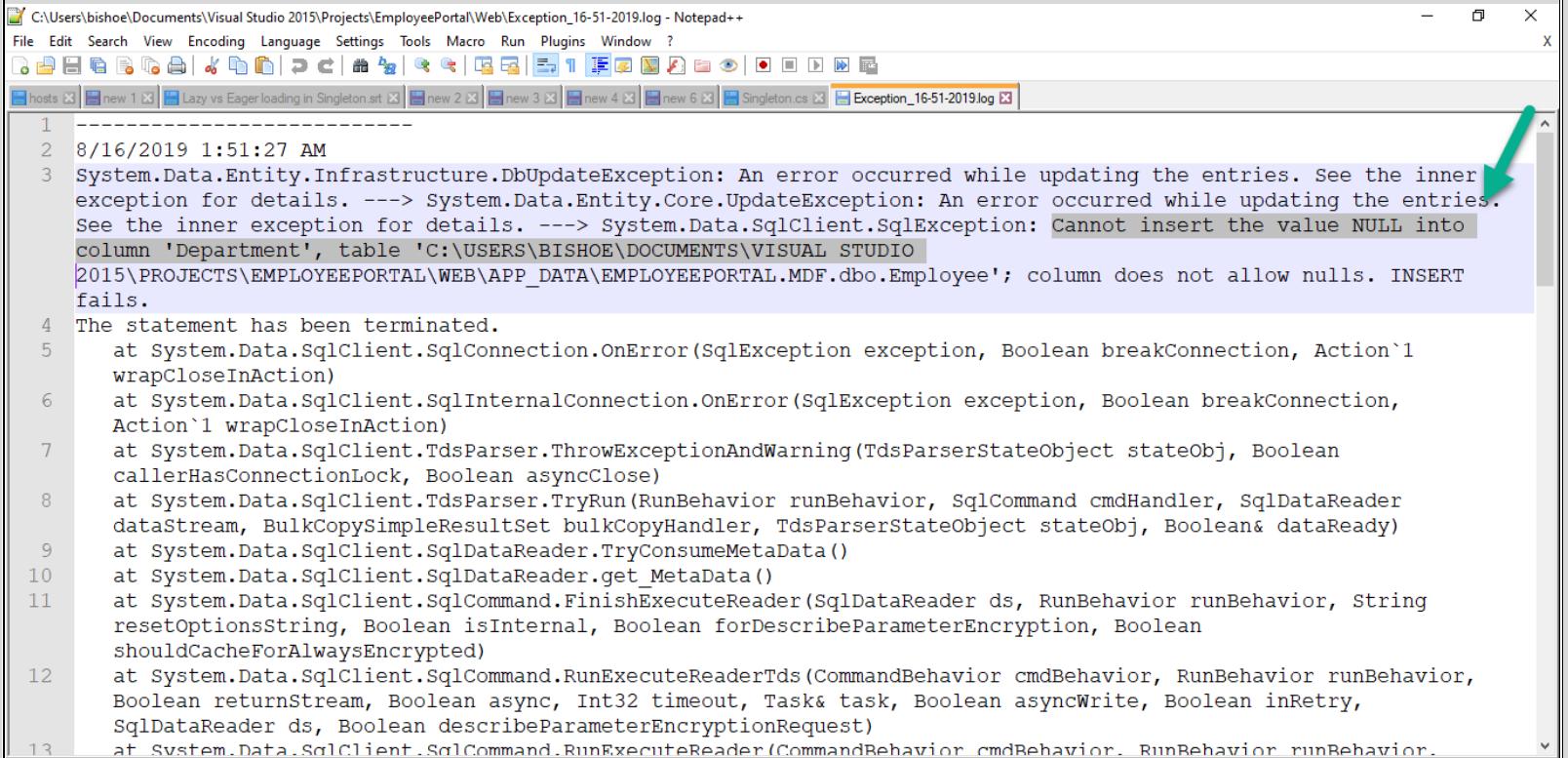
An error occurred while processing your request.

© 2019 - My ASP.NET Application

بص نجح وطلع اكسبيشين



بالفعل عملی ملف وسجل فيه الايرور



C:\Users\bishoe\Documents\Visual Studio 2015\Projects\EmployeePortal\Web\Exception_16-51-2019.log - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

hosts new 1 Lazy vs Eager loading in Singleton str new 2 new 3 new 4 new 6 Singleton.cs Exception_16-51-2019.log

```
1 -----
2 8/16/2019 1:51:27 AM
3 System.Data.Entity.Infrastructure.DbUpdateException: An error occurred while updating the entries. See the inner exception for details. ---> System.Data.Entity.Core.UpdateException: An error occurred while updating the entries. See the inner exception for details. ---> System.Data.SqlClient.SqlException: Cannot insert the value NULL into column 'Department', table 'C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL.MDF.dbo.Employee'; column does not allow nulls. INSERT fails.
4 The statement has been terminated.
5     at System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)
6     at System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)
7     at System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose)
8     at System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean& dataReady)
9     at System.Data.SqlClient.SqlDataReader.TryConsumeMetaData()
10    at System.Data.SqlClient.SqlDataReader.get_MetaData()
11    at System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString, Boolean isInternal, Boolean forDescribeParameterEncryption, Boolean shouldCacheForAlwaysEncrypted)
12    at System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Boolean async, Int32 timeout, Task& task, Boolean asyncWrite, Boolean inRetry, SqlDataReader ds, Boolean describeParameterEncryptionRequest)
13    at System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior,
```

BISHOYNABIL شوف كاتب الایرور بنجاح

Factory Design Pattern in Arabic

In this session we will learn

- What is Factory Design Pattern
- Implementation Guidelines
- Simple factory implementation

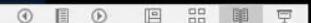
Suggested Videos :

Part 1 to 7 Design Patterns Tutorial

Link to Dot Net Basics, C#, LINQ,SQL Server, ASP.NET, ADO.NET and EF video series
<https://www.youtube.com/c/kudvenkatarabic/playlists>

2

Slide 2 of 15



What is Singleton Design Pattern

Creational design patterns type in Gang of four category as follows :

Singleton

Factory

Abstract Factory

Builder

Prototype patterns.

3

Slide 3 of 15



Factory Design Pattern

Gang Of Four Definition

"Define an interface for creating an object, but let subclasses decide which class to instantiate. The Factory method lets a class defer instantiation it uses to subclasses"

Factory pattern is one of the most used design patterns in real world applications

Factory pattern creates object without exposing the creation logic to the client and refer to newly created object using a common interface



4

Slide 4 of 15

BISHOY NABIL
ما هو ال

Factory design pattern

انتا بتعرف انترفيس والانترفيس دى بتكون مسؤولة عن انشاء الاوبيجيكت بقى التعريف
المحاضر مقدرش يترجمه صح

ال

Factory pattern

يتسمح للكلاس انه يقوم بتأجيل عملية
برضوا فشل فى ترجمة الشرح فى النقطة دى

بيقولك ال

Factory

من اشهر الباترن اللي بيتم تقديمها على ارض الواقع

وبقولك ال

Factory design

بتس محلك انه تعمل

اوبيكت

بدون انه ما يكون فيه توقع عندك انه يحصل

Creation logic from client

والامر ده بيعد الى انه الاوبيكت لما بيتعمل بيتعمل عن طريق

ال

Common interface

هنشرح اللي جايينه بعددين في الشرح

كما في الدايرام فوق بيقولك مينفعش اخل الكلاينت يوصل لل

Product

بدون الوصول الى اولا الى

Factory

وفكتوري تقوم بانشاء البرودكت

Implementation Guidelines

Choose Factory Pattern when

- The Object needs to be extended to subclasses
- The Classes doesn't know what exact sub-classes it has to create
- The Product implementation tend to change over time and the Client remains unchanged

5

Slide 5 of 15

BISHOYNABIL
امتا اختيار
ال

Design factory pattern

لما يكون عندك اوبجيكت عايز تكبره او تعمله

Extended

لـ

Subclasses

تاني حاجة

لما يكون عدك كلاسات فعليا انتا مش عارف الصب كلاسيز اللي تتبعها او المرتبطة بيهـا
حتى ان يتم انشائه فى اللحظة دى نستخدم الـ

Factory

Simple Factory Example

Business Requirement

Differentiate employees as permanent and contract and segregate their pay scales as well as bonus based on their employee types.

We can address the above requirement with the below implementations :

- Implement without Factory Pattern.
- Use a Simple Factory.
- Enhance Simple factory to Factory Method Pattern.

Note : We will be working on the Employee Portal that we used in the Singleton tutorials. Please refer to them before proceeding.

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Toolbox:** Standard Visual Studio toolbox.
- Solution Explorer:** Shows the Solution 'EmployeePortal' (2 projects) with the following structure:
 - Logger
 - Web
 - Properties
 - References
 - ILog.cs
 - Log.cs
 - Factory
 - AbstractFactory
 - fonts
 - Models
 - Scripts
 - Views
 - favicon.ico
 - Global.asax
 - packages.config
 - Project_Readme.html
 - Startup.cs
 - Web.config
- SQL Query Editor:** SQLQuery2.sql* window showing a CREATE TABLE statement for 'Employee_Type'. A green arrow points from the code editor area to the message bar below.
- Message Bar:** Command(s) completed successfully.
- Status Bar:** Shows the current state as Ready, with Ln 8, Col 1, Ch 1, and INS indicators.
- Bottom Navigation:** Output, Error List, Data Tools Operations, Solution Explorer, Team Explorer, Class View.

Text at the bottom center: بنعمل جدول جديد

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, SQL, TOOLS, ARCHITECTURE, TEST, ANALYZE, DEVEEXPRESS, WINDOW, HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, and others.
- Toolbox:** Standard Visual Studio toolbox.
- Solution Explorer:** Shows the Solution 'EmployeePortal' (2 projects) with two projects: Logger and Web. The Web project contains files like ILog.cs, Log.cs, App_Data, App_Start, Content, Controllers, Factory, AbstractFactory, fonts, Models, Scripts, Views, favicon.ico, Global.asax, packages.config, Project_Readme.html, Startup.cs, and Web.config.
- Code Editor:** A T-SQL script for creating the Employee table is open. The script includes columns for Id, Name, JobDescription, Number, Department, HourlyPay, Bonus, and EmployeeTypeID, with a primary key clustered on Id and a foreign key constraint linking EmployeeTypeID to Employee_Type.Id.
- Output Window:** Shows the message "Command(s) completed successfully."
- Status Bar:** Displays "146 %", "Query executed successfully at 6:31...", "Ln 15", "Col 1", "Ch 1", "INS", and "Ready".

بمسح جداول الموظفين واصيغه من جديد مع اضافة بعض الحقول الجديدة

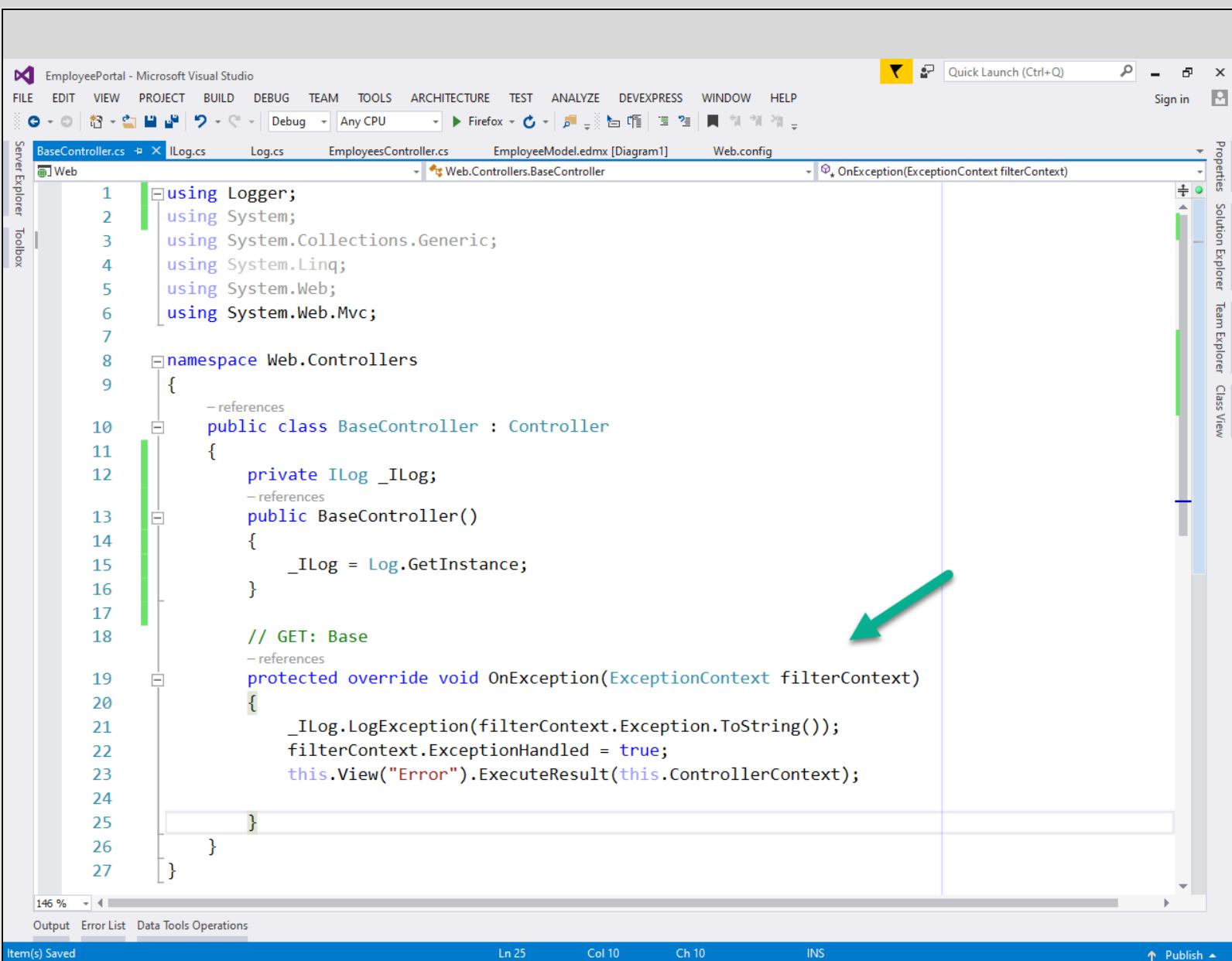
وبصيغه في جدول

Employee type

حقل 2

Permanent

Contract



```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP
Debug Any CPU Firefox
BaseController.cs ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config
Web.Controllers.BaseController
@_* OnException(ExceptionContext filterContext)

1 using Logger;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Web;
6 using System.Web.Mvc;
7
8 namespace Web.Controllers
9 {
10     public class BaseController : Controller
11     {
12         private ILog _ILog;
13         public BaseController()
14         {
15             _ILog = Log.GetInstance();
16         }
17
18         // GET: Base
19         protected override void OnException(ExceptionContext filterContext)
20         {
21             _ILog.LogError(filterContext.Exception.ToString());
22             filterContext.ExceptionHandled = true;
23             this.View("Error").ExecuteResult(this.ControllerContext);
24
25         }
26     }
27 }
```

بنصيف كنترولار جديد فارع وبحط فيه الميثود دى اللي استخدامناها فى الدروس السابقة
زى ما هيا

وبطلع فوق بعرف نسخه من الانترفيس وبعمل كونستركتور بعد كده ونكمel

انا ليه عملت كده عشان اخلى الكنترولار ده المسئول عن الاكسيشين كلها فى المشروع
وكل كنترول يحصل فيه اضافة او تعديل ميأثرش على كود الاكسيشين وهنا كده هقدر
اعمل

HANDLE

لای اکسپشنین بظہر

بعض النظر عن حدوث اى تغییلات فى اى کنترولار اخر

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP
BaseController.cs ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config
Sign in
Server Explorer Toolbox Properties Solution Explorer Team Explorer Class View
Web
10  using System;
11
12  namespace Web.Controllers
13  {
14      public class EmployeesController : Controller
15      {
16          //private ILog _ILog; ←
17
18          private EmployeePortalEntities db = new EmployeePortalEntities();
19
20          public EmployeesController()
21          {
22              ***←
23              //_ILog = Log.GetInstance; ←
24
25              //protected override void OnException(ExceptionContext filterContext)
26              //{
27              //    _ILog.LogError(filterContext.Exception.ToString());
28              //    filterContext.ExceptionHandled = true;
29              //    this.View("Error").ExecuteResult(this.ControllerContext);
30
31          }
32
33
34          // GET: Employees
35          public ActionResult Index()
36          {
37              return View(db.Employees.ToList());
```

هنا بمسحهم خلاص مش عايز الكود ده غير في ال

BASE

فقط

Add Scaffold

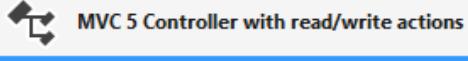
X

Installed

Common Controller



MVC 5 Controller - Empty



MVC 5 Controller with read/write actions



MVC 5 Controller with views, using Entity Framework



Web API 2 Controller – Empty



Web API 2 Controller with actions, using Entity Framework



Web API 2 Controller with read/write actions



Web API 2 OData v3 Controller with actions, using Entity Framework



Web API 2 OData v3 Controller with read/write actions

MVC 5 Controller with views, using Entity Framework
by Microsoft
v5.0.0.0

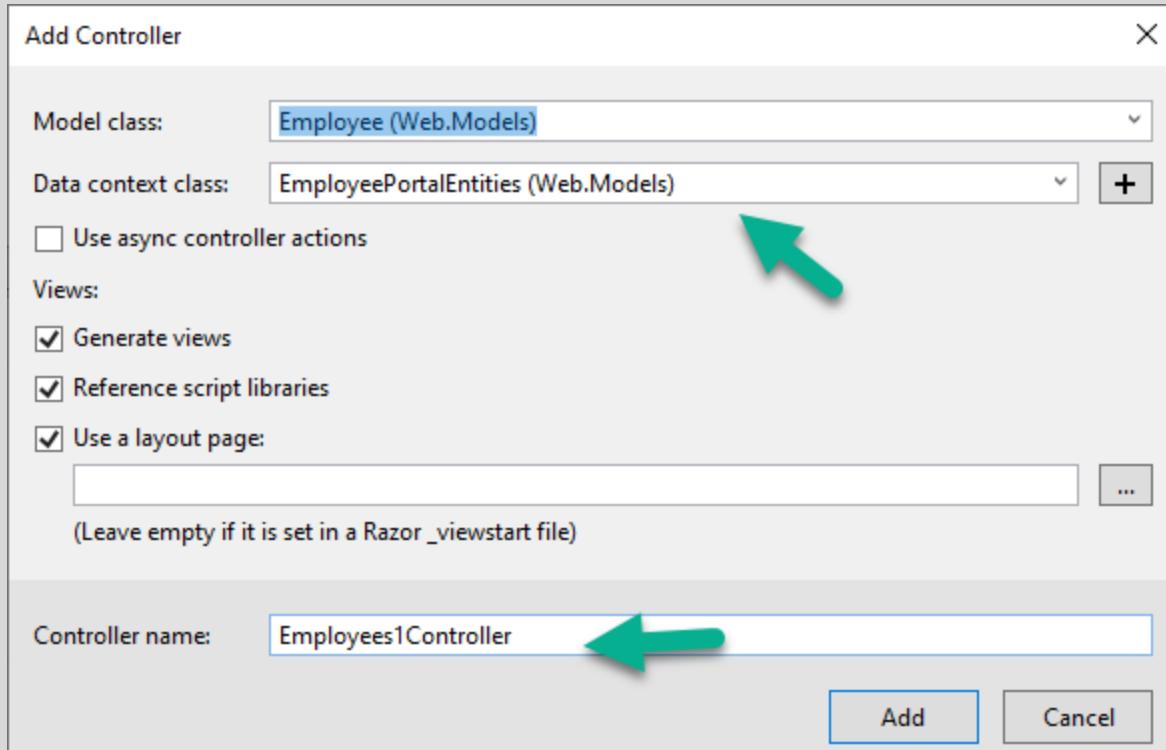
An MVC controller with actions and Razor views to create, read, update, delete, and list entities from an Entity Framework data context.

Id: MvcControllerWithContextScaffolder

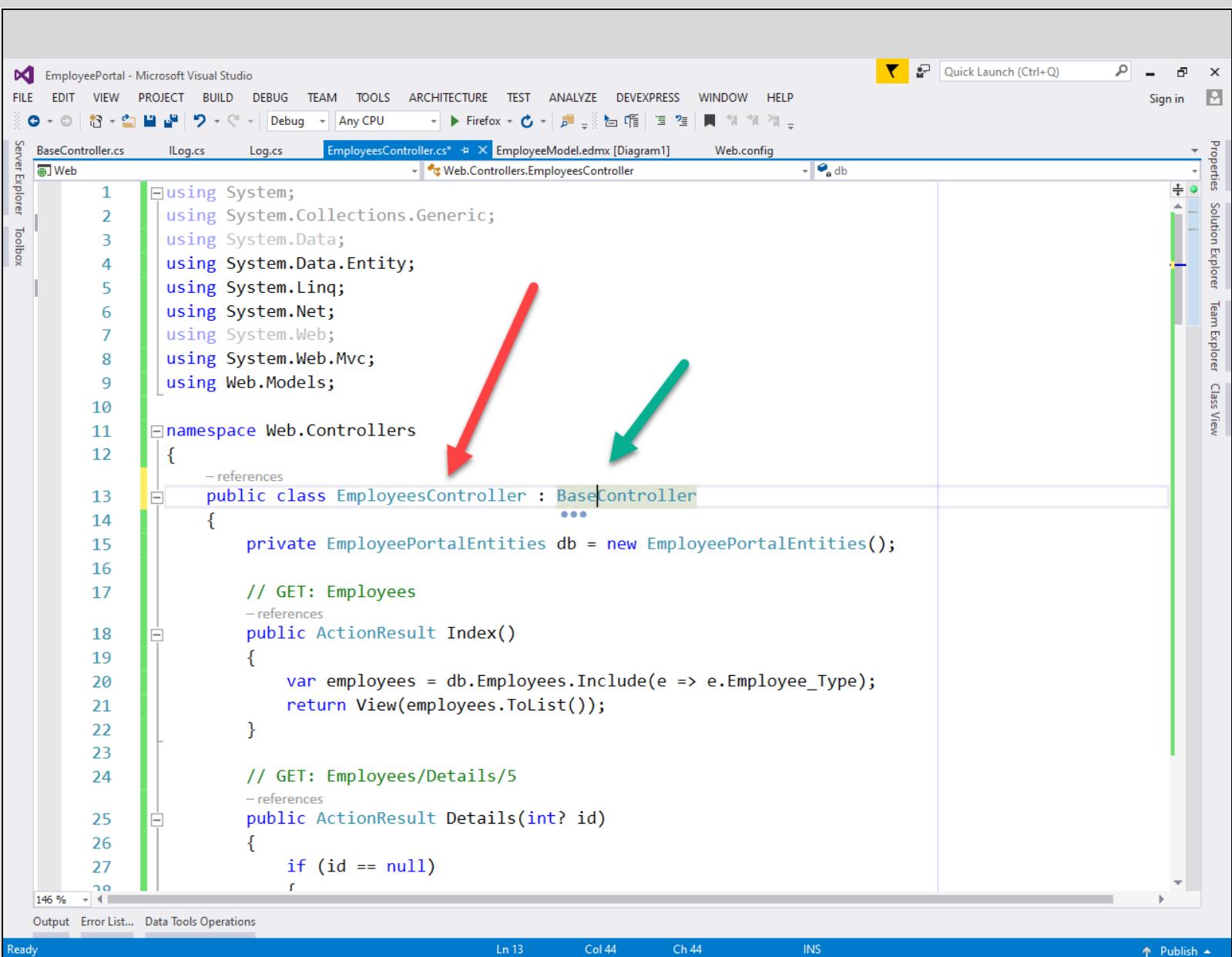
[Click here to go online and find more scaffolding extensions.](#)

Add

Cancel



هغير الاسم واشيل الـ 1 اللى فى الاسم
وادوس حفظ هيفولى انه موجود قبل كده هقوله اعمله
OVERWRITE



```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP
Debug Any CPU Firefox
BaseController.cs ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config
Web.Controllers.EmployeesController db

1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.Linq;
6 using System.Net;
7 using System.Web;
8 using System.Web.Mvc;
9 using Web.Models;
10
11 namespace Web.Controllers
12 {
13     public class EmployeesController : BaseController
14     {
15         private EmployeePortalEntities db = new EmployeePortalEntities();
16
17         // GET: Employees
18         public ActionResult Index()
19         {
20             var employees = db.Employees.Include(e => e.Employee_Type);
21             return View(employees.ToList());
22         }
23
24         // GET: Employees/Details/5
25         public ActionResult Details(int? id)
26         {
27             if (id == null)
28             {
29                 return HttpNotFound();
30             }
31             var employee = db.Employees.Find(id);
32             if (employee == null)
33             {
34                 return HttpNotFound();
35             }
36             return View(employee);
37         }
38
39         // POST: Employees/Create
40         [HttpPost]
41         public ActionResult Create(Employees employee)
42         {
43             if (ModelState.IsValid)
44             {
45                 db.Employees.Add(employee);
46                 db.SaveChanges();
47                 return RedirectToAction("Index");
48             }
49             return View(employee);
50         }
51
52         // GET: Employees/Edit/5
53         public ActionResult Edit(int? id)
54         {
55             if (id == null)
56             {
57                 return HttpNotFound();
58             }
59             var employee = db.Employees.Find(id);
60             if (employee == null)
61             {
62                 return HttpNotFound();
63             }
64             return View(employee);
65         }
66
67         // POST: Employees/Edit/5
68         [HttpPost]
69         public ActionResult Edit(Employees employee)
70         {
71             if (ModelState.IsValid)
72             {
73                 db.Entry(employee).State = EntityState.Modified;
74                 db.SaveChanges();
75                 return RedirectToAction("Index");
76             }
77             return View(employee);
78         }
79
80         // GET: Employees/Delete/5
81         public ActionResult Delete(int? id)
82         {
83             if (id == null)
84             {
85                 return HttpNotFound();
86             }
87             var employee = db.Employees.Find(id);
88             if (employee == null)
89             {
90                 return HttpNotFound();
91             }
92             db.Employees.Remove(employee);
93             db.SaveChanges();
94             return RedirectToAction("Index");
95         }
96
97         protected override void Dispose(bool disposing)
98         {
99             if (disposing)
100             {
101                 db.Dispose();
102             }
103             base.Dispose(disposing);
104         }
105     }
106 }
```

هنا خلیته ي

عمل

Inherit

من كنترول ال

Base

يعنى اى ايور هيحصل هنا كنترولار

Base

هيقوم بمعالجه فورا

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEPRESS WINDOW HELP
Debug Any CPU Firefox
BaseController.cs ILog.cs Log.cs EmployeesController.cs EmployeeModel.edmx [Diagram1] Web.config
Properties Solution Explorer Team Explorer Class View
Server Explorer Toolbox
51 public ActionResult Create([Bind(Include = "Id,Name,JobDescription,Number,Department,HourlyPay")]
52 {
53     if (ModelState.IsValid)
54     {
55         if(employee.EmployeeTypeID == 1)
56         {
57             employee.HourlyPay = 8;
58             employee.Bonus = 10;
59         }
56         else if(employee.EmployeeTypeID == 2)
57         {
58             employee.HourlyPay = 12;
59             employee.Bonus = 15;
60         }
61     }
62     db.Employees.Add(employee);
63     db.SaveChanges();
64     return RedirectToAction("Index");
65 }
66
67 ViewBag.EmployeeTypeID = new SelectList(db.Employee_Type, "Id", "EmployeeType", employee.EmployeeTypeID);
68 return View(employee);
69 }
70
71 // GET: Employees/Edit/5
72 // references
73 public ActionResult Edit(int? id)
74 {
75     if (id == null)
76     {
77         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
78     }
79 }
```

ينحسب له البونس بناء الموظف الدائم والموظف اللي بعقد زي المثال راجع المثال

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Quick Launch:** Quick Launch (Ctrl+Q) with a search bar.
- User Account:** Sign in button.
- Code Editor:** The main window displays the `Create.cshtml` file. The code uses Razor syntax (`@Html.EditorFor`, `@Html.LabelFor`) to generate HTML forms. Specific sections of the code are highlighted with blue boxes:
 - Line 37: `@Html.EditorFor(model => model.Number, new { htmlAttributes = new { @class = "form-control" })`
 - Line 43: `@Html.LabelFor(model => model.Department, htmlAttributes: new { @class = "control-label col-md-2" })`
 - Line 51: `@Html.LabelFor(model => model.HourlyPay, htmlAttributes: new { @class = "control-label col-md-2" })`
 - Line 59: `@Html.LabelFor(model => model.Bonus, htmlAttributes: new { @class = "control-label col-md-2" })`
 - Line 66: `@Html.LabelFor(model => model.EmployeeTypeID, "EmployeeTypeID", htmlAttributes: new { @class = "control-label col-md-2" })`
- Solution Explorer:** Shows files like BaseController.cs, ILog.cs, Log.cs, EmployeesController.cs, EmployeeModel.edmx [Diagram1], and Web.config.
- Properties:** A sidebar tab under the Solution Explorer.
- Toolbox:** A sidebar tab under the Solution Explorer.
- Output:** Shows Output, Error List, Data Tools Operations.
- Status Bar:** Ready, Ln 50, Col 2, Ch 2, INS, Publish.

هنسح دول عشان نمنعه انه يدخلهم للاهه غلط نمسح ونشغل الابلكشن

Design X Sql serv X c# inte X chrome X DownS X Dot Ne X Design X Design X G New T X Create X Create X

localhost:49936/Employees/Create

Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Application name Home About Contact Register Log in

Create

Employee

Name Jhon

JobDescription Manager

Number A123

Department It

EmployeeTypeID Permanent

Create

Back to List

© 2019 - My ASP.NET Application

بضیف یوزر

localhost:49936/Employees/Create

Application name Home About Contact Register Log in

Create

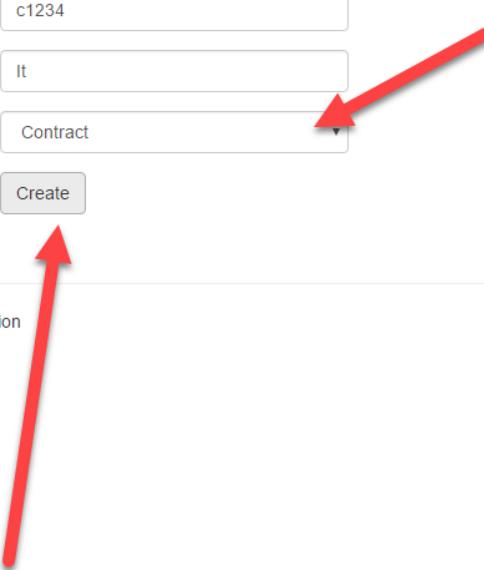
Employee

Name	Smith
JobDescription	Contract Dev
Number	c1234
Department	It
EmployeeTypeID	Contract

[Create](#)

[Back to List](#)

© 2019 - My ASP.NET Application



بضيف موظف بس

Contract

Name	JobDescription	Number	Department	HourlyPay	Bonus	EmployeeType
Jhon	Manager	A123	It	8.00	10.00	Permanent
Smith	Contract Dev	c1234	It			Contract

© 2019 - My ASP.NET Application

```
if (employee.EmployeeTypeID == 1)
{
    employee.HourlyPay = 8;
    employee.Bonus = 10;
}
else if (employee.EmployeeTypeID == 2)
{
    employee.HourlyPay = 12;
    employee.Bonus = 5;
}
```

الكود بالشكل ده غلط لانه كل ما اضيف نوعه لازم اجي اعدل هنا وده موضوع متعب جدا
وحتى في صيانته الكود هبيقى صعب

نكمـل هـنضـيف فـولـدر جـديـد
Manager

ونـحـط بـداـخلـه الـاتـى انـتـرـفـيـس
BISHOYNABIL

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEPRESS WINDOW HELP
Server Explorer > EmployeeManager.cs > Create.cshtml EmployeesController.cs Error.cshtml EmployeePortal.edmx [Diagram1] BaseController.cs Web.config
Web
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Web.Manager
{
    interface IEmployeeManager
    {
        decimal GetBonus();
        decimal GetPay();
    }
}
```

بنصييف 2 ميثود واحدة للبونص والثانية بتجبيال

Pay

نكمـل

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DDEXPRESS WINDOW HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Toolbox:** Standard Visual Studio toolbox.
- Server Explorer:** Shows Data Connections, EmployeePortalEntities, and Servers.
- Solution Explorer:** Shows files like PermanentEmployeeManager.cs, IEmployeeManager.cs, Create.cshtml, EmployeesController.cs, Error.cshtml, and EmployeePortal.edmx [Diagram1].
- Properties:** Standard Visual Studio properties panel.
- Code Editor:** Displays the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace Web.Manager
7  {
8      public class PermanentEmployeeManager : IEmployeeManager
9      {
10         public decimal GetBonus()
11         {
12             throw new NotImplementedException();
13         }
14
15         public decimal GetPay()
16         {
17             throw new NotImplementedException();
18         }
19     }
20 }
```

Annotations in the code editor:

- A red arrow points to the line `public class PermanentEmployeeManager : IEmployeeManager`.
- A purple arrow points to the method `GetBonus()`.
- A purple arrow points to the method `GetPay()`.

Bottom Status Bar: Item(s) Saved, Ln 8, Col 61, Ch 61, INS, Publish.

يُعمل كلاس جديدة ببيورث الانترفيس بناعاتى ثم بعمل

Implement

للميثود من الانترفيس

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;

5
6  namespace Web.Manager
7  {
8      public class PermanentEmployeeManager : IEmployeeManager
9      {
10         public decimal GetBonus()
11         {
12             return 10;
13         }
14
15         public decimal GetPay()
16         {
17             return 8;
18         }
19     }
20 }
```

بقوله زى المثال فى الشرح الاول البونص 10 وال

Pay

8

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace Web.Manager
7  {
8      public class ContractEmployeeManager : IEmployeeManager
9      {
10         public decimal GetBonus()
11         {
12             return 5;
13         }
14
15         public decimal GetPay()
16         {
17             return 12;
18         }
19     }
20 }
```

بنعمل كلاس جديد حديد كما موضح

ودى

طريقة البونص للموظف اللي بعقد

In this session we will learn

- Simple Factory
- Factory Method Pattern Implementation

Suggested Videos :

Part 1 to 8 Design Patterns Tutorial

Link to Dot Net Basics, C#, LINQ, SQL Server, ASP.NET, ADO.NET and EF video series
<https://www.youtube.com/c/kudvenkatarabic/playlists>

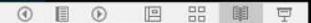
Factory Method Pattern Example

Business Requirement

- Differentiate employees as permanent and contract and segregate their pay scales as well as bonus based on their employee types. (Achieved using simple factory).
- Calculate Permanent employee house rent allowance.
- Calculate Contract employee medical allowance.

5

Slide 5 of 15



هنا لو عندي عايز اضيف مميزات زي مخصصات بيت وومخصصات تأمين صحي (")
ومخصصات البيت للموظفين الـ

Permanent

وبتوع التأمين الصحي خاصة بالموظفين اللي بعقد

```
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Web.Manager
8  {
9      public interface IEmployeeManager
10     {
11         decimal GetBonus();
12         decimal GetPay();
13     }
14 }
15
16
```

انا هنا مقدرش اضيف المميزات بتاعتي هنا مينفععش يتحطوا هنا

BISHOYNABIL

ومينفععش ال

PermanentEmployeeManager

و

ContractEmployeeManager

مينفععش الاثنين دول يعملوا

Implement

للانترفيس بتاعتي اللي اسمها

IEmployeeManager

ولانه مش كل المميزات موجودة في كل ال

types

اللي بيسرح حمار

وفي الانترفيس عندى الاتنين مشتركين في نفس

ال

Types

اللى هما

Permenet

و

Contract

وكده محتاج اعمل تعديلات

من هنا بتيجي فكرة الـ

Factory method pattern

BISHOYNABIL
Factory

الكود هيبقى معقد بدرجة كبيرة

تعالي نشوف شكل

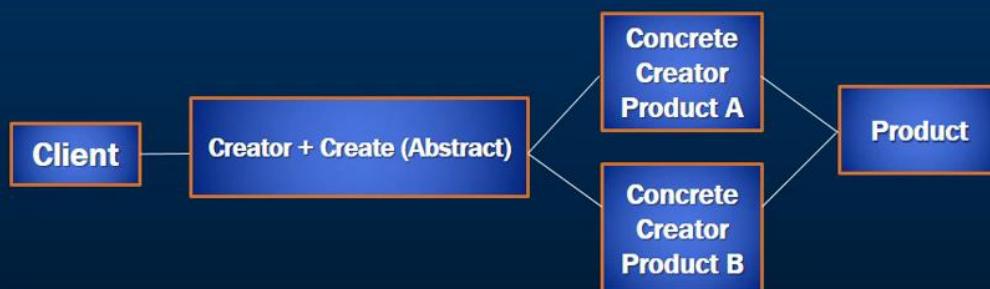
Factory

Factory Representation

Simple Factory



Factory Method Pattern



6

BISHOY NABIL
دھ ال

Simple factory

راجعنناه فى الدروس السابق

خلينا فى ال

Factory method pattern

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEXPRESS WINDOW HELP

Debug Any CPU Comodo Dragon

PermanentEmployeeManager.cs EmployeeManagerFactory.cs EmployeePortal.edmx dbo.Employee [Design] IEmployeeManager.cs

Quick Launch (Ctrl+Q) Sign in

Server Explorer Toolbox Properties Solution Explorer Team Explorer Class View

Update Script File: dbo.Employee.sql

	Name	Data Type	Allow Nulls	Default
Id	int		<input type="checkbox"/>	
Name	varchar(50)		<input type="checkbox"/>	
JobDescription	varchar(50)		<input type="checkbox"/>	
Number	varchar(50)		<input type="checkbox"/>	
Department	varchar(50)		<input type="checkbox"/>	
HourlyPay	decimal(18,0)		<input checked="" type="checkbox"/>	
Bonus	decimal(18,0)		<input checked="" type="checkbox"/>	
EmployeeTypeID	int		<input checked="" type="checkbox"/>	
HouseAllowance	decimal(18,0)		<input checked="" type="checkbox"/>	
MedicalAllowance	decimal(18,0)		<input checked="" type="checkbox"/>	

Keys (1)
<unnamed> (Primary Key, Clustered: Id)

Check Constraints (0)

Indexes (0)

Foreign Keys (1)
FK_Employee_EmployeeType (Id)

Triggers (0)

Design T-SQL

```
8     [Bonus]      DECIMAL (18) NULL,
9     [EmployeeTypeID] INT      NULL,
10    [HouseAllowance] DECIMAL NULL,
11    [MedicalAllowance] DECIMAL NULL,
12    PRIMARY KEY CLUSTERED ([Id] ASC),
```

146 % Connection Ready

Output Error List Data Tools Operations

Ready

RISHONIAR بيعمل تعديل للبيانات

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEXPRESS WINDOW HELP

Debug Any CPU Comodo Dragon

Employee.cs EmployeesController.cs PermanentEmployeeManager.cs EmployeeManagerFactory.cs EmployeePortal.edmx [Diagram1]

Quick Launch (Ctrl+Q) Sign in

Server Explorer Toolbox Properties Solution Explorer Team Explorer Class View

Employee.cs

```
19     public string JobDescription { get; set; }
20     public string Number { get; set; }
21     public string Department { get; set; }
22     public Nullable<decimal> HourlyPay { get; set; }
23     public Nullable<decimal> Bonus { get; set; }
24     public Nullable<int> EmployeeTypeID { get; set; } *
25     public Nullable<decimal> HouseAllowance { get; set; }
26     public Nullable<decimal> MedicalAllowance { get; set; }
27
28     public virtual Employee_Type Employee_Type { get; set; }
29 }
30 }
```

146 % Output Error List... Data Tools Operations

Ln 24 Col 25 Ch 25 INS

Ready

هتمسح الجزء ده ونكمel

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

Employee.cs EmployeesController.cs PermanentEmployeeManager.cs EmployeeManagerFactory.cs EmployeePortal.edmx [Diagram1]

Web Manager.PermanentEmployeeManager

```
12     return 10;
13 }
14
15     3 references
16     public decimal GetPay()
17     {
18         return 8;
19     }
20
21     0 references
22     public decimal GetHouseAllowance()
23     {
24         return 150;
25     }

```

146 %

Output Error List Data Tools Operations

Item(s) Saved

Ln 22 Col 24 Ch 24 INS

↑ Publish

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEEXPRESS WINDOW HELP

ContractEmployeeManager.cs Employee.cs EmployeesController.cs PermanentEmployeeManager.cs EmployeeManagerFactory.cs EmployeePortal.edmx [Diagram1]

Web Manager.ContractEmployeeManager

```
12     return 5;
13 }
14
15     3 references
16     public decimal GetPay()
17     {
18         return 12;
19     }
20
21     0 references
22     public decimal GetMedicalAllowance()
23     {
24         return 100;
25     }

```

146 %

Output Error List Data Tools Operations

Item(s) Saved

Ln 21 Col 23 Ch 23 INS

↑ Publish

بعمل ميثود في الكونتركت

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5
6 namespace Web.Factory.FactoryMethod
7 {
8     public class BaseEmployeeFactory
9     {
10 }
11 }
```

بعمل فولدر جديد داخل فولدر ال
BISHOYNABIL
Factory

وبنزل تحت اعمل كلاس
وهيكون مسؤل عن انشء الانترفييس
Iemployeemanagertype
ويتعامل كمان مع
Employee object model

```
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using Web.Manager;
6
7  namespace Web.Factory.FactoryMethod
8  {
9      public abstract class BaseEmployeeFactory
10     {
11         public abstract IEmployeeManager Create();
12     }
13 }
14
```

ودى هتكون مسؤولة عن انشاء التایب
BISHOYNABIL
IEmployeemangaer

ولازم نفتكر عندى نوعين من المديرين

Perm

And

Contract

ويعملوا اميلمېنت ل

IEmployee

واللى تحتوى بداخلها على ال

Getbounce

And

Houerpay

نكمـل هـخلـى الـكـلاـس دـه يـعـالـج الـ Employee object model

```
using Web.Models;
namespace Web.Factory.FactoryMethod
{
    public abstract class BaseEmployeeFactory
    {
        protected Employee _emp;
    }
    public partial class Employee
    {
        public int Id { get; set; }
    }
    public BaseEmployeeFactory(Employee emp)
    {
        _emp = emp;
    }
}
```

هـنا بـعـمل اوـبـجيـكت مـن الجـدول بـتـاعـى وـانـزل تـحـت اـعـمل الكـونـسـتـركـتـور وـاعـمل مـتـغـير نـوعـه
هـيـكون الجـدول بـتـاعـى وـانـزل تـحـت اـعـمل

Initialize

للـ

_emp

```
4  using System.Web;
5  using Web.Manager;
6  using Web.Models;
7
8  namespace Web.Factory.FactoryMethod
9  {
10    public class PermanentEmployeeFactory : BaseEmployeeFactory
11    {
12      public PermanentEmployeeFactory(Employee emp) : base(emp)
13      {
14      }
15
16      public override IEmployeeManager Create()
17      {
18        //throw new NotImplementedException();
19      }
20    }
}
```

بضميف كلاس جديد
الميثود

Create

هيا اللي هتعملني تايب من نوع
Permenet Employee Manager

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEXPRESS WINDOW HELP

Debug Any CPU Comodo Dragon

PermanentEmployeeFactory.cs BaseEmployeeFactory.cs ContractEmployeeManager.cs Employee.cs EmployeesController.cs PermanentEmployeeManager.cs

Properties Solution Explorer Team Explorer Class View

Server Explorer Toolbox

```
1 reference
10 public class PermanentEmployeeFactory : BaseEmployeeFactory
11 {
12     0 references
13     public PermanentEmployeeFactory(Employee emp) : base(emp)
14     {
15     }
16     1 reference
17     public override IEmployeeManager Create()
18     {
19         PermanentEmployeeManager manager = new PermanentEmployeeManager();
20         _emp.HouseAllowance = manager.GetHouseAllowance(); ← Red arrow
21         return manager; ← Red arrow
22     }
23 }
24 }
```

146 %

Output Error List Data Tools Operations

Item(s) Saved

Ln 20 Col 28 Ch 28 INS

BISHOYNARI

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE DEVEXPRESS WINDOW HELP

Debug Any CPU Comodo Dragon

ContractEmployeeFactory.cs PermanentEmployeeFactory.cs BaseEmployeeFactory.cs ContractEmployeeManager.cs Employee.cs EmployeesController.cs

Properties Solution Explorer Team Explorer Class View

Server Explorer Toolbox

```
7
8 namespace Web.Factory.FactoryMethod
9 {
10    1 reference
11    public class ContractEmployeeFactory : BaseEmployeeFactory
12    {
13        0 references
14        public ContractEmployeeFactory(Employee emp) : base(emp)
15    }
16    2 references
17    public override IEmployeeManager Create()
18    {
19        ContractEmployeeManager manager = new ContractEmployeeManager();
20        _emp.MedicalAllowance = manager.GetMedicalAllowance(); ← Red arrow
21        return manager; ← Red arrow
22    }
23 }
```

146 %

Output Error List Data Tools Operations

Item(s) Saved

Ln 21 Col 1 Ch 1 INS

BISHOYNARI

بعمل کلاس جديد عشان يرجع

Medical allowance

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Log.cs HomeController.cs PermanentEmployeeFactory.cs ContractEmployeeFactory.cs EmployeesController.cs EmployeeManagerFactory.cs BaseEmployeeFactory.cs
Sign in
Server Explorer Toolbox Properties Solution Explorer Class View
Web
8  namespace Web.Factory.FactoryMethod
9  {
10     public abstract class BaseEmployeeFactory
11     {
12         protected Employee _emp;
13
14         public BaseEmployeeFactory(Employee emp)
15         {
16             _emp = emp;
17         }
18
19         public Employee ApplySalary()
20         {
21             IEmployeeManager manager = this.Create();
22             _emp.Bonus = manager.GetBonus();
23             _emp.HourlyPay = manager.GetPay();
24
25             return _emp;
26         }
27
28     }
}
121 %
Output Error List
Ready Ln 23 Col 25 Ch 25 INS Publish
```

بضميف الميثود دى عشان ترجعلى البونص وساعات العمل

فى الاول بقوله اعملى متغير مانجر

This.create

هنا المانجر بتحتوى على الى

GetBonus

And

GetPay

فبستخدمها بعض النظر عن النوع اللي هيرجع سواء كان

Perm

Or

Contract

والنوعين دول بيعملوا امبليمنت لكل من البونص وساعات العمل

BISHOYNABIL

هنعممل كلاس جديد اسمه

Employee

Manager

هينادى على كلاس

Baseemployefactory

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
- Toolbars:** Standard, Debug, Start, Stop, Build, Analyze, Window, Help.
- Solution Explorer:** Shows the project structure with files like AbstractFactory.cs, FactoryMethod.cs, BaseEmployeeFactory.cs, ContractEmployeeFactory.cs, EmployeeManagerFactory.cs, PermanentEmployeeFactory.cs, AccountController.cs, BaseController.cs, EmployeesController.cs, HomeController.cs, ManageController.cs, Manager.cs, PermanentEmployeeManager.cs, AccountViewModels.cs, EmployeePortal.edmx, IdentityModels.cs, ManageViewModels.cs, and Scripts.
- Code Editor:** Displays the `EmployeeManagerFactory.cs` file with the following code:

```
7  namespace Web.Factory.FactoryMethod
8  {
9      public class EmployeeManagerFactory
10     {
11         public BaseEmployeeFactory CreateFactory(Employee emp)
12         {
13             BaseEmployeeFactory returnValue = null;
14
15             if (emp.EmployeeTypeID == 1)
16             {
17                 returnValue = new PermanentEmployeeFactory(emp);
18             }
19             else if (emp.EmployeeTypeID == 2)
20             {
21                 returnValue = new ContractEmployeeFactory(emp);
22             }
23         }
24         return returnValue;
25     }
26 }
```

Two green arrows point to the class definition (`EmployeeManagerFactory`) and the method definition (`CreateFactory`).

هنا بقوله الميثود دی هتنشئى النوع معتمد على ال
Factory

فممکن يکون

PermanentEmployeeFactory

or

ContractEmployeeFactory

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Quick Launch (Ctrl+Q) Sign in
Server Explorer Toolbox
EmployeeManagerFactory.cs BaseEmployeeFactory.cs
Web Web.Factory.FactoryMethod.EmployeeManagerFactory CreateFactory(Employee emp)
7 namespace Web.Factory.FactoryMethod
8 {
9     0 references
10    public class EmployeeManagerFactory
11    {
12        0 references
13        public BaseEmployeeFactory CreateFactory(Employee emp)
14        {
15            BaseEmployeeFactory returnValue = null;
16
17            if (emp.EmployeeTypeID == 1)
18            {
19                returnValue = new PermanentEmployeeFactory(emp);
20            }
21            else if (emp.EmployeeTypeID == 2)
22            {
23                returnValue = new ContractEmployeeFactory(emp);
24            }
25        }
26        return returnValue;
27    }
}
Solution Explorer
Search Solution Explorer (Ctrl+F)
Controllers
    AccountController.cs
    BaseController.cs
    EmployeesController.cs
    HomeController.cs
    ManageController.cs
Factory
    AbstractFactory
    FactoryMethod
        BaseEmployeeFactory.cs
        ContractEmployeeFactory.cs
        EmployeeManagerFactory.cs
        PermanentEmployeeFactory.cs
        EmployeeManagerFactory.cs
    fonts
Manager
    ContractEmployeeManager.cs
    IEmployeeManager.cs
    PermanentEmployeeManager.cs
Models
    AccountViewModels.cs
    EmployeePortal.edmx
    IdentityModels.cs
    ManageViewModels.cs
Scripts
Output Error List...
Ln 24 Col 32 Ch 32 INS
Ready Publish
```

هنا بقوله شرط تحقق لو كان الايدى واحد بيقى تعاملى تحطلى داخل ال
BISHOYNABIL
Return value

نسخه جديدة من ال

Permanent

الشرط اللي تحت هغيره عشان الايدى عندي 3 مش 2 ونكمel عادى التغيير اهو

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Debug Any CPU Start Properties
EmployeeManagerFactory.cs PermanentEmployeeFactory.cs ContractEmployeeFactory.cs EmployeeManagerFactory.cs EmployeesController.cs
Web
7 namespace Web.Factory.FactoryMethod
8 {
9     public class EmployeeManagerFactory
10    {
11        public BaseEmployeeFactory CreateFactory(Employee emp)
12        {
13            BaseEmployeeFactory returnValue = null;
14
15            if (emp.EmployeeTypeID == 1)
16            {
17                returnValue = new PermanentEmployeeFactory(emp);
18            }
19            else if (emp.EmployeeTypeID == 3) // Yellow arrow points here
20            {
21                returnValue = new ContractEmployeeFactory(emp);
22            }
23        }
24        return returnValue;
}
146 %
Output Error List...
Item(s) Saved
Ln 19 Col 45 Ch 45 INS
Solution Explorer
Search Solution Explorer (Ctrl+F)
Controllers
AccountController.cs
BaseController.cs
EmployeesController.cs
HomeController.cs
ManageController.cs
Factory
AbstractFactory
FactoryMethod
BaseEmployeeFactory.cs
ContractEmployeeFactory.cs
EmployeeManagerFactory.cs
PermanentEmployeeFactory.cs
EmployeeManagerFactory.cs
fonts
Manager
ContractEmployeeManager.cs
IEmployeeManager.cs
PermanentEmployeeManager.cs
Models
AccountViewModels.cs
EmployeePortal.edmx
IdentityModels.cs
ManageViewModels.cs
Scripts
Solution Explorer Class View
Properties
```

BISHOYNABIL
نكمـل

هـنعمل التـعديـلات عـلـى الكـونـتـرـولـار عـشـان تـمـشـى مـع اـخـر حـاجـة عـامـلـينـها فـى الكـوـد

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

EmployeeManagerFactory.cs PermanentEmployeeFactory.cs ContractEmployeeFactory.cs EmployeeManagerFactory.cs EmployeesController.cs

Web

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.Linq;
6 using System.Net;
7 using System.Web;
8 using System.Web.Mvc;
9 using Web.Controllers;
10 using Web.Factory; ←
11 using Web.Manager;
12 using Web.Models;
13
14 namespace Web.Controllers
15 {
16     public class EmployeesController : BaseController
17     {
18         private EmployeePortalEntities db = new EmployeePortalEntities();
19     }
}
```

146 %

Output Error List...

Ready

Ln 10 Col 18 Ch 18 INS

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

EmployeePortal (2 projects)

- Logger
- Properties
- References
- ILog.cs
- Log.cs
- Web
- Properties
- References
- App_Data
- App_Start
- Content
- Controllers
- AccountController.cs
- BaseController.cs
- EmployeesController.cs ←
- HomeController.cs
- ManageController.cs
- Factory
- AbstractFactory
- FactoryMethod
- BaseEmployeeFactory.cs
- ContractEmployeeFactory.cs
- EmployeeManagerFactory.cs
- PermanentEmployeeFactory.cs

Solution Explorer Class View

هـنـيـغـرـ الـمـسـارـ

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

EmployeeManagerFactory.cs PermanentEmployeeFactory.cs ContractEmployeeFactory.cs EmployeeManagerFactory.cs EmployeesController.cs*

Web

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.Linq;
6 using System.Net;
7 using System.Web;
8 using System.Web.Mvc;
9 using Web.Controllers;
10 using Web.Factory.FactoryMethod; ←
11 using Web.Manager;
12 using Web.Models;
13
14 namespace Web.Controllers
15 {
16     public class EmployeesController : BaseController
17     {
18         private EmployeePortalEntities db = new EmployeePortalEntities();
19     }
}
```

146 %

Output Error List...

Ready

Ln 10 Col 32 Ch 32 INS

Solution Explorer

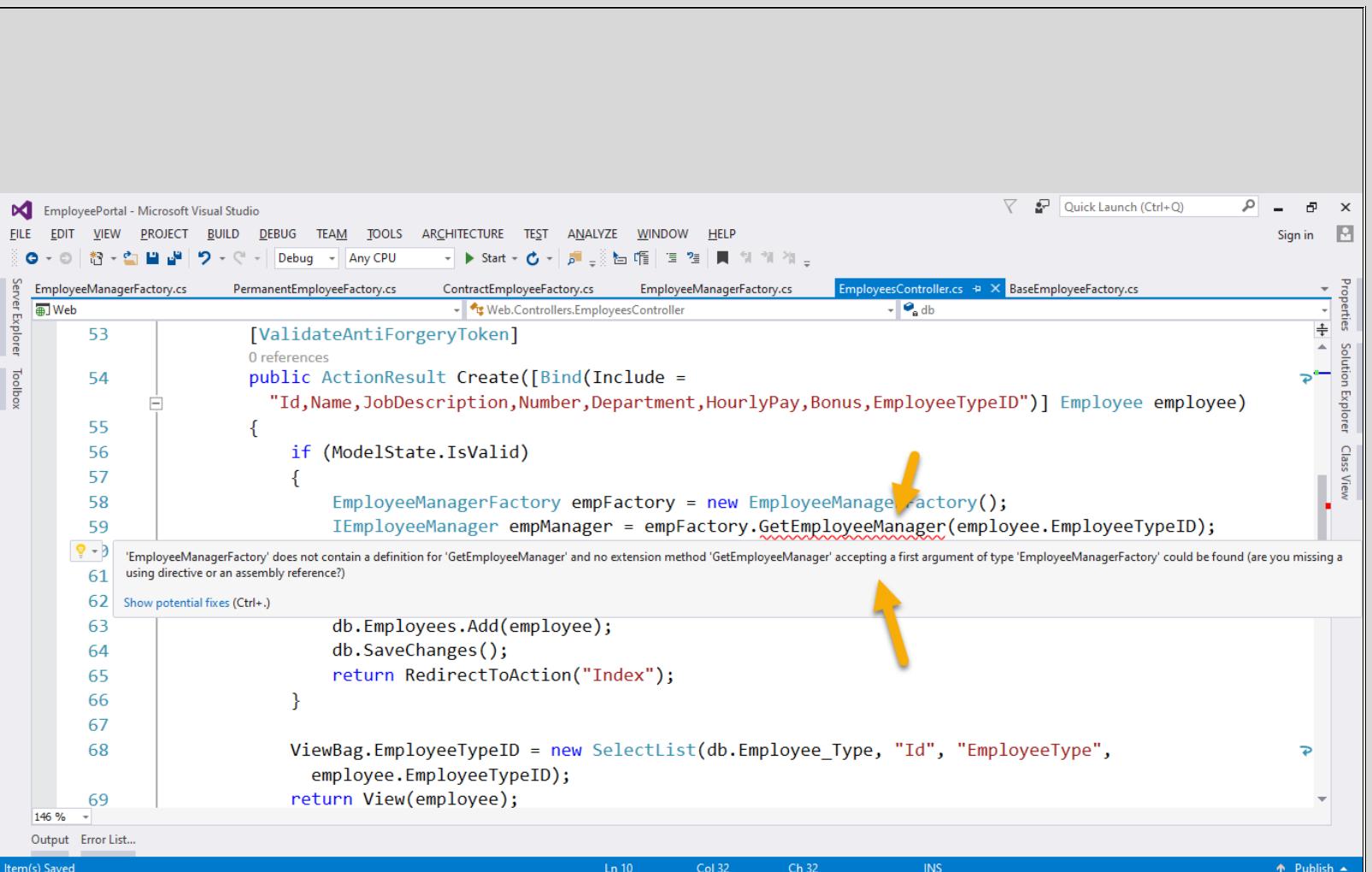
Search Solution Explorer (Ctrl+Shift+F)

EmployeePortal (2 projects)

- Logger
- Properties
- References
- ILog.cs
- Log.cs
- Web
- Properties
- References
- App_Data
- App_Start
- Content
- Controllers
- AccountController.cs
- BaseController.cs
- EmployeesController.cs ←
- HomeController.cs
- ManageController.cs
- Factory
- AbstractFactory
- FactoryMethod
- BaseEmployeeFactory.cs
- ContractEmployeeFactory.cs
- EmployeeManagerFactory.cs
- PermanentEmployeeFactory.cs

Solution Explorer Class View

نـكـمـلـ



```
53 [ValidateAntiForgeryToken]
54 public ActionResult Create([Bind(Include =
55     "Id,Name,JobDescription,Number,Department,HourlyPay,Bonus,EmployeeTypeID")] Employee employee)
56 {
57     if (ModelState.IsValid)
58     {
59         EmployeeManagerFactory empFactory = new EmployeeManagerFactory();
60         IEmployeeManager empManager = empFactory.GetEmployeeManager(employee.EmployeeTypeID);
61         'EmployeeManagerFactory' does not contain a definition for 'GetEmployeeManager' and no extension method 'GetEmployeeManager' accepting a first argument of type 'EmployeeManagerFactory' could be found (are you missing a
62         Show potential fixes (Ctrl+.)
63         db.Employees.Add(employee);
64         db.SaveChanges();
65         return RedirectToAction("Index");
66     }
67
68     ViewBag.EmployeeTypeID = new SelectList(db.Employee_Type, "Id", "EmployeeType",
69     employee.EmployeeTypeID);
       return View(employee);

```

هندل المسار
بس قبلها هندل الاتى

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

EmployeeManagerFactory.cs PermanentEmployeeFactory.cs ContractEmployeeFactory.cs EmployeeManagerFactory.cs EmployeesController.cs BaseEmployeeFactory.cs

Web Web.Controllers.EmployeesController Create(Employee employee)

```
53 [ValidateAntiForgeryToken]
54 public ActionResult Create([Bind(Include =
55     "Id,Name,JobDescription,Number,Department,HourlyPay,Bonus,EmployeeTypeID")] Employee employee)
56 {
57     if (ModelState.IsValid)
58     {
59         EmployeeManagerFactory empFactory = new EmployeeManagerFactory();
60         IEmployeeManager empManager = empFactory.GetEmployeeManager(employee.EmployeeTypeID);
61         employee.Bonus = empManager.GetBonus();
62         employee.HourlyPay = empManager.GetPay();
63
64         db.Employees.Add(employee);
65         db.SaveChanges();
66         return RedirectToAction("Index");
67     }
68
69     ViewBag.EmployeeTypeID = new SelectList(db.Employee_Type, "Id", "EmployeeType",
70         employee.EmployeeTypeID);
71     return View(employee);
72 }
```

146 % Output Error List...

Ready Ln 58 Col 1 Ch 1 INS Publish

هندل الكود
BISHOYNABIL
من اللي فوق الى

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

EmployeeManagerFactory.cs PermanentEmployeeFactory.cs ContractEmployeeFactory.cs EmployeeManagerFactory.cs EmployeesController.cs BaseEmployeeFactory.cs

Web Web.Controllers.EmployeesController Create(Employee employee)

```
53 [ValidateAntiForgeryToken]
54 public ActionResult Create([Bind(Include =
55     "Id,Name,JobDescription,Number,Department,HourlyPay,Bonus,EmployeeTypeID")] Employee employee)
56 {
57     if (ModelState.IsValid)
58     {
59         BaseEmployeeFactory empFactory = new EmployeeManagerFactory().CreateFactory(employee);
60         IEmployeeManager empManager = empFactory.GetEmployeeManager(employee.EmployeeTypeID);
61         employee.Bonus = empManager.GetBonus();
62         employee.HourlyPay = empManager.GetPay();
63
64         db.Employees.Add(employee);
65         db.SaveChanges();
66         return RedirectToAction("Index");
67     }
68
69     ViewBag.EmployeeTypeID = new SelectList(db.Employee_Type, "Id", "EmployeeType",
70         employee.EmployeeTypeID);
71     return View(employee);
72 }
```

146 % Output Error List...

Item(s) Saved Ln 58 Col 9 Ch 9 INS Publish

بغير النوع فى الاول ثم بروح انادى على ميثود ال

Create

ونكمل

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar reads "EmployeePortal - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, WINDOW, and HELP. The toolbar has various icons for file operations like Open, Save, and Build. The status bar at the bottom shows "Item(s) Saved", "Ln 61", "Col 60", "Ch 60", and "INS". The main code editor window displays the following C# code:

```
53 [ValidateAntiForgeryToken]
54 public ActionResult Create([Bind(Include =
55     "Id,Name,JobDescription,Number,Department,HourlyPay,Bonus,EmployeeTypeID")]
56     Employee employee)
57 {
58     if (ModelState.IsValid)
59     {
60         BaseEmployeeFactory empFactory = new EmployeeManagerFactory().CreateFactory(employee);
61         //IEmployeeManager empManager = empFactory.GetEmployeeManager(employee.EmployeeTypeID);
62         //employee.Bonus = empManager.GetBonus();
63         //employee.HourlyPay = empManager.GetPay();
64
65         db.Employees.Add(employee);
66         db.SaveChanges();
67         return RedirectToAction("Index");
68
69     }
70
71     ViewBag.EmployeeTypeID = new SelectList(db.Employee_Type, "Id", "EmployeeType",
72         employee.EmployeeTypeID);
73     return View(employee);
74 }
```

The code is annotated with several colored highlights: a green highlight covers the first few lines of the method body; a blue highlight covers the line "BaseEmployeeFactory empFactory = new EmployeeManagerFactory().CreateFactory(employee);"; a yellow highlight covers the three commented-out lines starting with "//". The status bar indicates the cursor is at line 61, column 60.

وقفت الكود ده لاني مش محتاجه وهستخدم

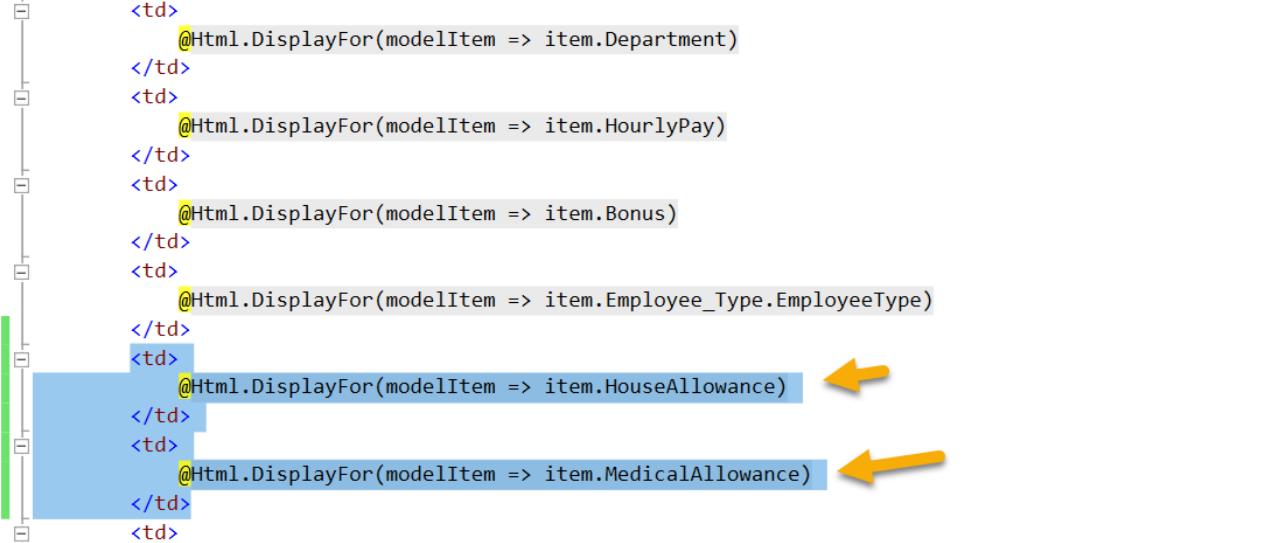
The screenshot shows the Microsoft Visual Studio interface with the title bar "EmployeePortal - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, WINDOW, and HELP. The toolbar has icons for file operations like Open, Save, and Print. The status bar at the bottom shows "Ready", "Ln 64", "Col 17", "Ch 17", and "INS".

The code editor displays the `EmployeesController.cs` file:

```
53     [ValidateAntiForgeryToken]
54     public ActionResult Create([Bind(Include =
55         "Id,Name,JobDescription,Number,Department,HourlyPay,Bonus,EmployeeTypeID")] Employee employee)
56     {
57         if (ModelState.IsValid)
58         {
59             BaseEmployeeFactory empFactory = new EmployeeManagerFactory().CreateFactory(employee);
60             //IEmployeeManager empManager = empFactory.GetEmployeeManager(employee.EmployeeTypeID);
61             //employee.Bonus = empManager.GetBonus();
62             //employee.HourlyPay = empManager.GetPay();
63             empFactory.ApplySalary();
64
65             db.Employees.Add(employee);
66             db.SaveChanges();
67             return RedirectToAction("Index");
68         }
69
70         ViewBag.EmployeeTypeID = new SelectList(db.Employee Type, "Id", "EmployeeType",
```

A yellow rectangular highlight surrounds the line `empFactory.ApplySalary();`. The status bar at the bottom indicates the cursor is in column 17, line 64.

بستخدم الميثود دى الللى بتجييلى البونص وال Hourlypay



The screenshot shows the Microsoft Visual Studio IDE interface with the title bar "EmployeePortal - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, WINDOW, and HELP. The toolbar contains icons for Back, Forward, Stop, Refresh, and Save. The status bar at the bottom shows "146 %", "1265", "Col 14", "Ch 14", and "INS".

The main code editor window displays the file "Index.cshtml" with the following content:

```
49     <td>
50         @Html.DisplayFor(modelItem => item.Department)
51     </td>
52     <td>
53         @Html.DisplayFor(modelItem => item.HourlyPay)
54     </td>
55     <td>
56         @Html.DisplayFor(modelItem => item.Bonus)
57     </td>
58     <td>
59         @Html.DisplayFor(modelItem => item.Employee_Type.EmployeeType)
60     </td>
61     <td>
62         @Html.DisplayFor(modelItem => item.HouseAllowance) ← Yellow arrow
63     </td>
64     <td>
65         @Html.DisplayFor(modelItem => item.MedicalAllowance) ← Yellow arrow
66     </td>
67     <td>
68         @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
         @Html.ActionLink("Details", "Details", new { id=item.Id }) |

```

The code uses C# syntax with Razor directives (@Html.DisplayFor and @Html.ActionLink) to generate HTML table cells (td). Lines 62 and 65 are highlighted with blue selection bars and have yellow arrows pointing to them from the left.

بنصیف الحقول دی فی الفیو

```
28     </th>
29     <th>
30         @Html.DisplayNameFor(model => model.Bonus)
31     </th>
32     <th>
33         @Html.DisplayNameFor(model => model.Employee_Type.EmployeeType)
34     </th>
35     <th>
36         @Html.DisplayNameFor(model => model.HouseAllowance) ←
37     </th>
38     <th>
39         @Html.DisplayNameFor(model => model.MedicalAllowance) ←
40     </th>
41     <th></th>
42   </tr>
43
44 @foreach (var item in Model) {
45   <tr>
46     <td>
47       @Html.DisplayFor(modelItem => item.Name)
```

وبصیف اللیبل الحاصل بهم للعرض ونکمل

http://localhost:49936/Employees/Create

Create - My ASP.NET Appli... ×

Application name Home About Contact Register Log in

Create

Employee

Name	<input type="text" value="alex"/>
JobDescription	<input type="text" value="dev"/>
Number	<input type="text" value="T1909"/>
Department	<input type="text" value="it"/> x
EmployeeTypeID	<input type="text" value="Permanent"/>

←

[Back to List](#)

© 2019 - My ASP.NET Application

http://localhost:49936/Employees/Create

هنـصـيف وـاـحـد جـديـد

http://localhost:49936/Employees/Create

Create - My ASP.NET Appli... X

Application name Home About Contact Register Log in

Create

Employee

Name	mary
JobDescription	Qa
Number	Q1040
Department	it
EmployeeTypeID	Contract

[Create](#)

[Back to List](#)

© 2019 - My ASP.NET Application



http://localhost:49936/Employees

Index - My ASP.NET Appli... X

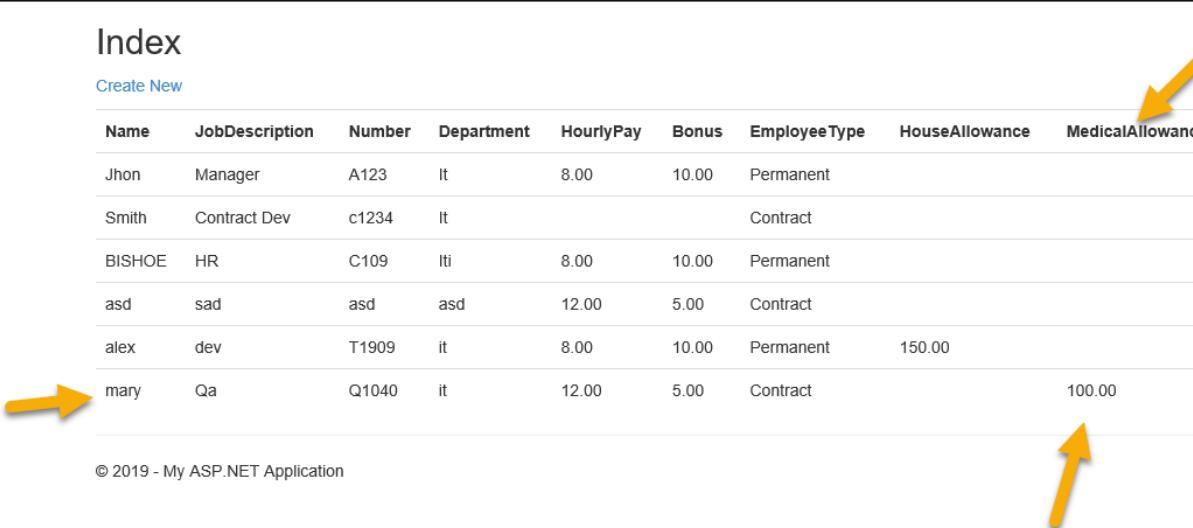
Application name Home About Contact Register Log in

Index

Create New

Name	JobDescription	Number	Department	HourlyPay	Bonus	EmployeeType	HouseAllowance	MedicalAllowance	
Jhon	Manager	A123	It	8.00	10.00	Permanent			Edit Details Delete
Smith	Contract Dev	c1234	It			Contract			Edit Details Delete
BISHOE	HR	C109	Iti	8.00	10.00	Permanent			Edit Details Delete
asd	sad	asd	asd	12.00	5.00	Contract			Edit Details Delete
alex	dev	T1909	it	8.00	10.00	Permanent	150.00		Edit Details Delete
mary	Qa	Q1040	it	12.00	5.00	Contract		100.00	Edit Details Delete

© 2019 - My ASP.NET Application

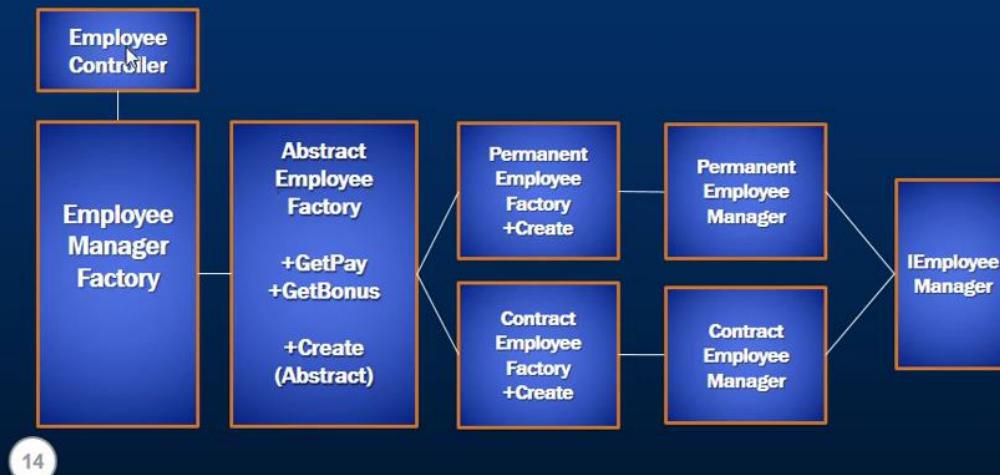


نـجـاح وـحـسـب المـطـلـوـب

Factory Method Pattern

Gang Of Four Definition

"Define an interface(abstract class) for creating an object, but let subclasses decide which class to instantiate. The Factory method lets a class defer instantiation it uses to subclasses"



14

دھ ملخص الدرس او اللي حصل فعلیاً واحدنا عملناه

BISHOYNABIL

Design Pattern

Part 10 – Abstract Factory Design Pattern

Mohammad Emad Arafah

PRAGIM Technologies

Mohammad_Arafah@hotmail.com

<http://csharp-video-tutorials.blogspot.com>

PRAGIM Technologies | 9900113931 | www.pragimtech.com | www.facebook.com/pragimtech

In this session we will learn

- What is Abstract Factory Design Pattern
- Implementation Guidelines
- Abstract Factory Implementation
- Differences between Factory Method and Abstract Factory Design Pattern

Suggested Videos :

[Part 1 to 9 Design Patterns Tutorial](#)

[Link to Dot Net Basics, C#, LINQ,SQL Server, ASP.NET, ADO.NET and EF video series](#)
<https://www.youtube.com/c/kudvenkatarabic/playlists>

2

Slide 2 of 19



What is Abstract Factory Design Pattern

"The Abstract factory pattern provides a way to encapsulate a group of individual factories that have a common theme without specifying their concrete classes"

- **The Abstract Factory Pattern provides an interface for creating families of related or dependent objects without specifying their concrete classes.**
- **Abstract Factory pattern belongs to creational patterns and is one of the most used design patterns in real world applications.**
- **Abstract factory is a super factory that creates other factories.**

3

Slide 3 of 19



تعريف

ال

Abstract

طريقة بتعمل بيها

Encapsulate

لمجموعه من

Factories

بمعنى اخر ال

Abstract

بتروduct ب بانترفيس قادره على انشاء مجموعه من الاوبيجيكس والاوبيجكتس دي ممكّن تكون بتعتمد على بعضها البعض او

او

BISHOYNABIL
Related
مع بعضها

بدون ما تحدد ال

Concrete class

Abstract factory

باترن بيتنتمي الى

Creational patterns

وهو واحد من ال

Design pattern

الاكثر استخداما عند بناء ابلكشين

Abstract factory

يعتبر

سوبر فكتوري بامكانه انه يعمل

Creation

لـ

Other factories

لو مش واضح الكلام ده على نكمل الدرس وهنفهمه

The slide has a dark blue background. At the top center, the title 'Implementation Guidelines' is written in a large, yellow, sans-serif font. Below the title, there is a section header 'Choose Abstract Factory Pattern when' in a smaller yellow font. Following this, there is a bulleted list of four items, each starting with a yellow triangle symbol. The list items are: 'The application need to create multiple families of objects or products.', 'We need to use only one of the subset of families of objects at a given point of time.', and 'We want to hide the implementations of the families of products by decoupling the implementation of each of these operations.' In the bottom left corner of the slide area, there is a small white circle containing the number '4'. The bottom right corner of the slide shows the standard Microsoft PowerPoint navigation icons.

اما بحتاج استخدم

Abstract factory pattern

لما يكون الأبلكشن بحتاج يعمل اكتر من عملية انشاء للأوبيجكتس او المنتجات

او عند انشاء مجموعة من الاوبيجكتس او مجموعة جزئية عند لحظة معينة

او

لما اقوم بعمل اخفاء

لل

Implementations

لمجموعة من المنتجات عن طريق عمل

Decoupling

لل

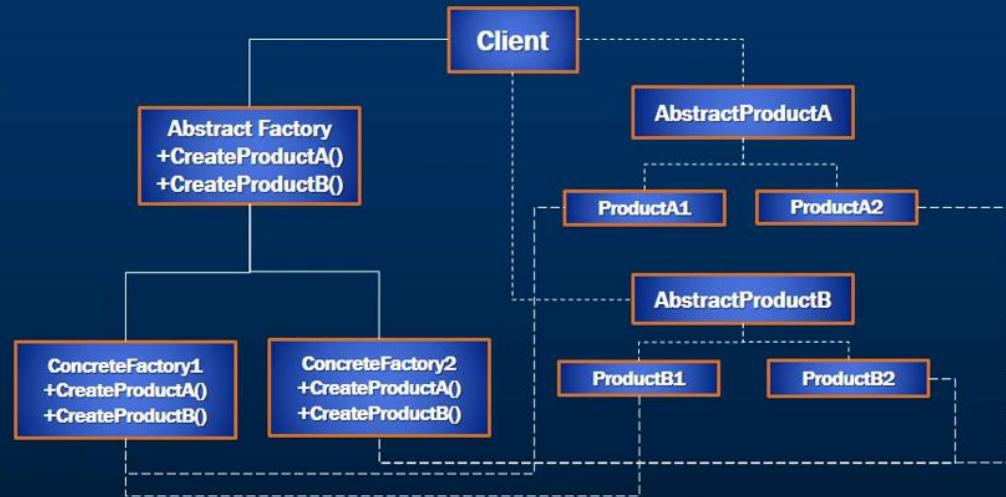
Implementations

للاوبرشين الخاصة بالاوبيجكتس

مش فاهم حاجة في العملي هنفهم كل حاجة

Abstract Factory Representation

Abstract Factory



هنا الكلاينت كلاس وال

Abstract factory

و

Abstract class

هما انترفيسيز

وشوف ال

Concrete

عامل

Implement

لـ

Abstract factory

BISHOYNABIL
وكذلك

Product a1

Product a2

يعملوا

Implement

لللانترفيس

Abstract product a

Abstract Factory Pattern Example

Business Requirement

- Handout computers to Contract and Permanent employees based on the designation and employee type with below specifications
- Permanent Employee
 - Managerial Position is eligible for Apple MAC Book Laptop
 - Non Managerial Position is eligible for Apple iMac desktop
- Contract Employee
 - Managerial Position is eligible for Dell Laptop
 - Non Managerial Position is eligible for Dell desktop

6

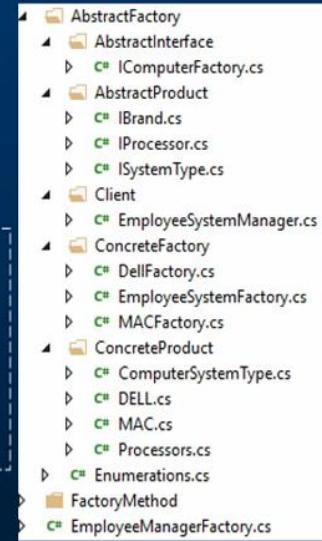
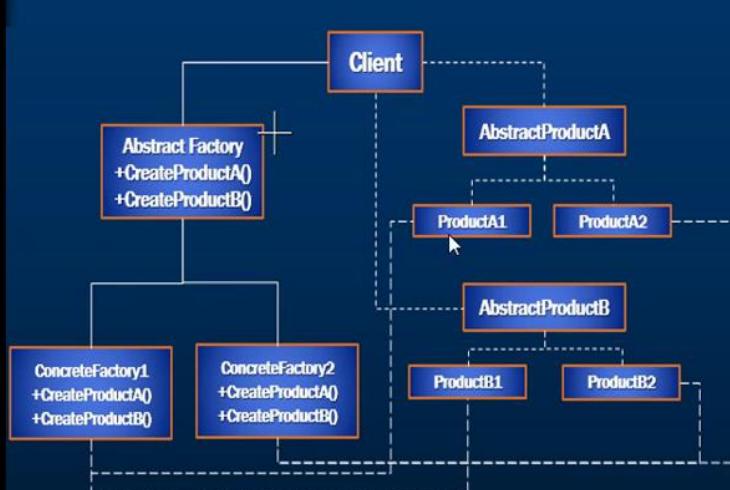
Slide 6 of 19

BISHOY NABIL
مثال ده

يقولك شركة كمبيوتر وفيه نوعين نوع مدير هيأخذ لاب توب ماك بوك واللهى مش مدير
هيأخذ ديسك توب ابل

وتحت لو مدير هيأخذ لاب ديل ولو مش مدير هيأخذ ديل ديسكتوب

Abstract Factory Implementation



7

Slide 7 of 19

BISHOY NABI

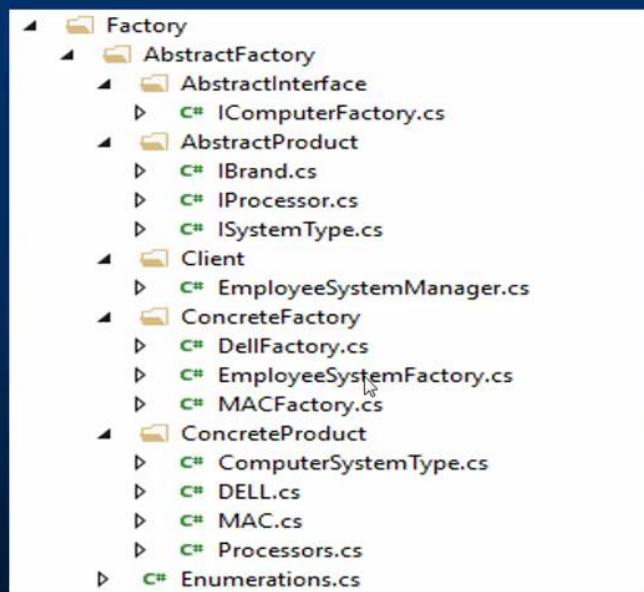
Real implement

واللى على اليمين ده المفروض شكل البروجيكت اللي هنعمله

الشرح اللي هنمشى عليه والخطوات

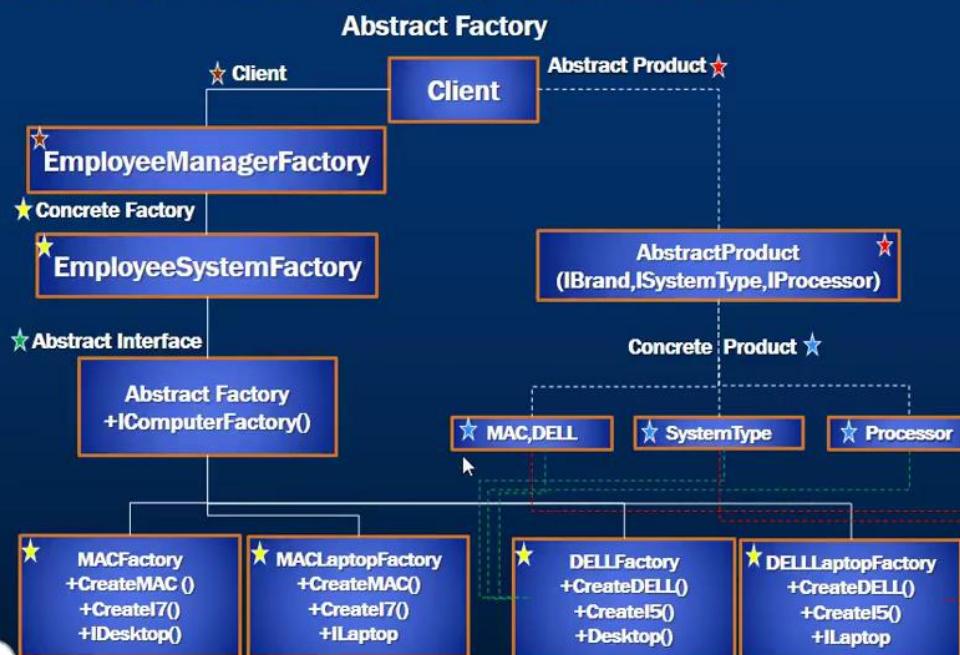
Abstract Factory Implementation

Step 5 : Arrange the classes in the **AbstractFactory** folder with the below structure.



15

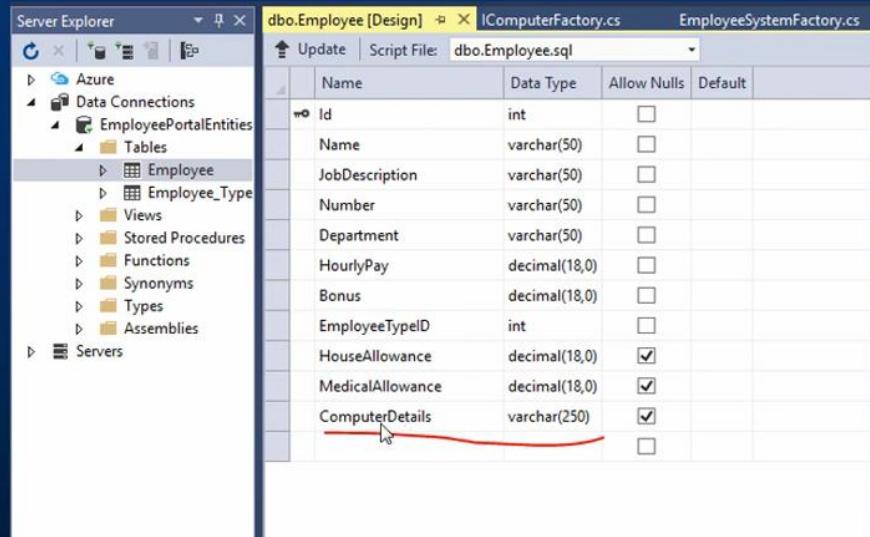
Abstract Factory Representation



16

Abstract Factory Implementation

Step 1 : Add ComputerDetails to the existing Employee table.



	Name	Data Type	Allow Nulls	Default
#	Id	int	<input type="checkbox"/>	
	Name	varchar(50)	<input type="checkbox"/>	
	JobDescription	varchar(50)	<input type="checkbox"/>	
	Number	varchar(50)	<input type="checkbox"/>	
	Department	varchar(50)	<input type="checkbox"/>	
	HourlyPay	decimal(18,0)	<input type="checkbox"/>	
	Bonus	decimal(18,0)	<input type="checkbox"/>	
	EmployeeTypeID	int	<input type="checkbox"/>	
	HouseAllowance	decimal(18,0)	<input checked="" type="checkbox"/>	
	MedicalAllowance	decimal(18,0)	<input checked="" type="checkbox"/>	
	ComputerDetails	varchar(250)	<input checked="" type="checkbox"/>	

8

Abstract Factory Implementation

Step 2 : Open EmployeePortal.edmx under the Models folder of the solution and update the model from the database (Right click on the model designer and choose update from database option).

Step 3 : Create AbstractFactory folder under existing Factory folder and add the below classes.

```
public enum ComputerTypes
{
    Laptop,
    Desktop
}
public enum Brands
{
    APPLE,
    DELL,
}
public enum Processors
{
    I3,
    I5,
    I7
}
```

```
public interface IBrand
{
    string GetBrand();
}
public interface ISysteMType
{
    string GetSystemType();
}
public interface IProcessor
{
    string GetProcessor();
}

public interface IComputerFactory
{
    IProcessor Processor();
    IBrand Brand();
    ISysteMType SystemType();
}
```

9

Abstract Factory Implementation

```
public class MAC : IBrand
{
    public string GetBrand()
    {
        return Brands.APPLE.ToString();
    }
}
public class DELL : IBrand
{
    public string GetBrand()
    {
        return Brands.DELL.ToString();
    }
}
public class I5 : IProcessor
{
    public string GetProcessor()
    {
        return Processors.I5.ToString();
    }
}
public class I7 : IProcessor
{
    public string GetProcessor()
    {
        return Processors.I7.ToString();
    }
}
public class Laptop : ISystemType
{
    public string GetSystemType()
    {
        return ComputerTypes.Laptop.ToString();
    }
}
public class Desktop : ISystemType
{
    public string GetSystemType()
    {
        return ComputerTypes.Desktop.ToString();
    }
}
```

10

DIRECTIVE

Abstract Factory Implementation

```
public class MACFactory : IComputerFactory
{
    public IProcessor Processor() { return new I7(); }

    public IBrand Brand() { return new MAC(); }

    public virtual ISystemType SystemType() { return new Desktop(); }
}
public class MACLapTopFactory : MACFactory
{
    public override ISystemType SystemType() { return new Laptop(); }
}

public class DellFactory : IComputerFactory
{
    public IProcessor Processor() { return new I5(); }

    public IBrand Brand() { return new DELL(); }

    public virtual ISystemType SystemType() { return new Desktop(); }
}
public class DellLaptopFactory : DellFactory
{
    public override ISystemType SystemType() { return new Laptop(); }
}
```

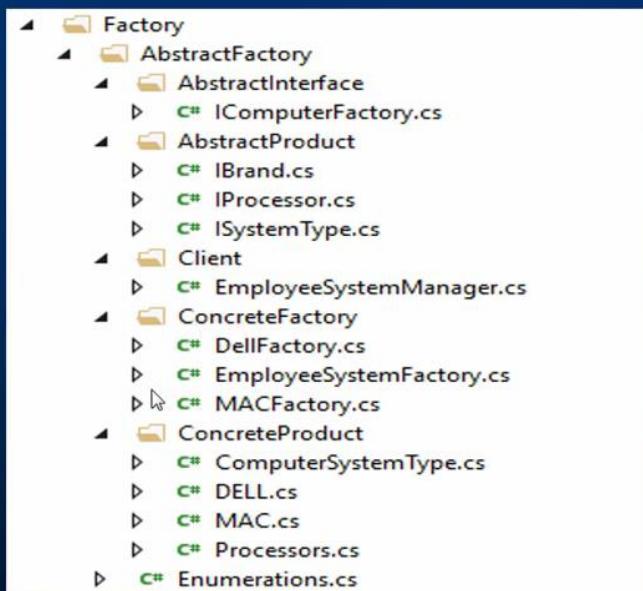
Abstract Factory Implementation

```
public class EmployeeSystemFactory
{
    public IComputerFactory Create(Employee emp)
    {
        IComputerFactory returnValue = null;
        if (emp.EmployeeTypeID == 1)
        {
            if (emp.JobDescription == "Manager")
                returnValue = new MACLapTopFactory();
            else
                returnValue = new MACFactory();
        }
        else if (emp.EmployeeTypeID == 2)
        {
            if (emp.JobDescription == "Manager")
                returnValue = new DellLaptopFactory();
            else
                returnValue = new DellFactory();
        }
        return returnValue;
    }
}
```

12

Abstract Factory Implementation

Step 5 : Arrange the classes in the **AbstractFactory** folder with the below structure.



15

خطوة انشاء فolderات جديدة وتقسيم الكلاسات

ونباء أول خطوة

The screenshot shows the Microsoft Visual Studio interface. In the top navigation bar, the project is named "EmployeePortal - Microsoft Visual Studio". The menu items include FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, SQL, TOOLS, ARCHITECTURE, TEST, ANALYZE, DEVEXPRESS, WINDOW, and HELP. Below the menu is a toolbar with icons for various operations. The title bar shows the current file is "SQLQuery1.sql". The main workspace contains the following T-SQL code:

```
1 CREATE TABLE [dbo].[Employee] (
2     [Id] INT IDENTITY (1, 1) NOT NULL,
3     [Name] VARCHAR (50) NOT NULL,
4     [JobDescription] VARCHAR (50) NOT NULL,
5     [Number] VARCHAR (50) NOT NULL,
6     [Department] VARCHAR (50) NOT NULL,
7     [HourlyPay] DECIMAL (18) NOT NULL,
8     [Bonus] DECIMAL (18) NOT NULL,
9     [EmployeeTypeID] INT NOT NULL,
10    [HouseAllowance] DECIMAL (18) NULL,
11    [MedicalAllowance] DECIMAL (18) NULL,
12    PRIMARY KEY CLUSTERED ([Id] ASC))
13
14
15
```

The code is highlighted with syntax coloring. The bottom pane shows the results of the query execution:

```
T-SQL Message
Command(s) completed successfully.
```

Output window status: 146 %, Query executed successfully at 4:57:26 AM, (LocalDB)\MSSQLLocalDB (13... | DESKTOP-AVE6BFU\bishoe... | C:\USERS\BISHOE\DOCUME... | 00:00:00 | 0 rows

بزود الحقول دى على الجدول

```
CREATE TABLE [dbo].[Employee] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Name] VARCHAR (50) NOT NULL,
    [JobDescription] VARCHAR (50) NOT NULL,
    [Number] VARCHAR (50) NOT NULL,
    [Department] VARCHAR (50) NOT NULL,
    [HourlyPay] DECIMAL (18) NOT NULL,
    [Bonus] DECIMAL (18) NOT NULL,
    [EmployeeTypeID] INT NOT NULL,
    [HouseAllowance] DECIMAL (18) NULL,
    [MedicalAllowance] DECIMAL (18) NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC))
```

وبعدن بعمل ابديت للانتى نكملي

```
public class Enumerations
{
    public enum ComputerTypes
    {
        Laptop,
        Desktop
    }

    public enum Brands
    {
        Apple,
        Dell
    }

    public enum Processors
    {
        I3,
        I5,
        I7
    }
}
```

بعضیف کلاس زی الشرح ونکمل

Add New Item - Web

Installed

Sort by: Default

Search Installed Templates (Ctrl+E)

Type: Visual C#
An empty interface definition

	Class	Visual C#
	DevExpress v15.2 Report Wizard	Visual C#
	DevExpress v15.2 Report Service	Visual C#
	Interface	Visual C#
	ADO.NET Entity Data Model	Visual C#
	Application Manifest File	Visual C#
	Assembly Information File	Visual C#
	Class Diagram	Visual C#
	Code Analysis Rule Set	Visual C#
	Code File	Visual C#

[Click here to go online and find templates.](#)

Name: IBrand.cs

Add Cancel

بضيف الانترفيس بساعدنا ونكمel

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Processor.cs ISystemType.cs Enumerations.cs IBrand.cs

Web Web.Factory.AbstractFactory.IBrand GetBrand()

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Web.Factory.AbstractFactory
8 {
9     public interface IBrand
10    {
11        string GetBrand();
12    }
13}
14
```

121% Output Error List...

Ready

Ln 14 Col 1 Ch 1 INS

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
 - AccountController.cs
 - BaseController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ManageController.cs
- Factory
 - AbstractFactory
 - Enumerations.cs
 - IBrand.cs
 - ISystemType.cs
 - Processor.cs
- FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs
 - EmployeeManagerFactory.cs
- fonts
- Manager
 - ContractEmployeeManager.cs
 - IEmployeeManager.cs

65% CPU

BISHOYNABIL
نكمـل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Processor.cs ISystemType.cs Enumerations.cs IBrand.cs

Web Web.Factory.AbstractFactory.ISystemType GetSystemType()

```
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Web.Factory.AbstractFactory
8 {
9     public interface ISystemType
10    {
11        string GetSystemType();
12    }
13}
14
```

121% Output Error List...

Ready

Ln 10 Col 6 Ch 6 INS

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
 - AccountController.cs
 - BaseController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ManageController.cs
- Factory
 - AbstractFactory
 - Enumerations.cs
 - IBrand.cs
 - ISystemType.cs
 - Processor.cs
- FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs
 - EmployeeManagerFactory.cs
- fonts
- Manager
 - ContractEmployeeManager.cs
 - IEmployeeManager.cs

66% CPU

بصیف انترفیس کمان

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Processor.cs ISystemType.cs Enumerations.cs IBrand.cs

Web Web.Factory.AbstractFactory.Processor GetProcessor()

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Web.Factory.AbstractFactory
8 {
9     public interface Processor
10    {
11        string GetProcessor();
12    }
13 }
```

Server Explorer Toolbox Solution Explorer Properties

Search Solution Explorer (Ctrl+Shift+F)

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
 - AccountController.cs
 - BaseController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ManageController.cs
- Factory
 - AbstractFactory
 - Enumerations.cs
 - IBrand.cs
 - ISystemType.cs
 - Processor.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - fonts
- Manager
 - ContractEmployeeManager.cs
 - IEmployeeManager.cs

Solution Explorer Class View 66% CPU

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Mac.cs Processor.cs ISystemType.cs Enumerations.cs IBrand.cs

Web Web.Factory.AbstractFactory.Mac GetBrand()

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using static Web.Factory.AbstractFactory.Enumerations;
6
7 namespace Web.Factory.AbstractFactory
8 {
9     public class Mac : IBrand
10    {
11        public string GetBrand()
12        {
13            return Brands.Apple.ToString();
14        }
15    }
16 }
```

Server Explorer Toolbox Solution Explorer Properties

Search Solution Explorer (Ctrl+Shift+F)

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
 - AccountController.cs
 - BaseController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ManageController.cs
- Factory
 - AbstractFactory
 - Enumerations.cs
 - IBrand.cs
 - ISystemType.cs
 - Mac.cs
 - Processor.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - fonts
- Manager
 - ContractEmployeeManager.cs

Solution Explorer Class View 45% CPU

هنا بضيف كلاس لـ

Apple

ماركة الكمبيوتر

وبعمل ايميلمینت لللانترفيسيس بتاعتنا

Ibrand

وفي الاميلمینت برجع

Apple

ونكمل

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Dell.cs Mac.cs Processor.cs ISystemType.cs Enumerations.cs IBrand.cs
Web Web.Factory.AbstractFactory.Dell GetBrand()
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using static Web.Factory.AbstractFactory.Enumerations;
6
7  namespace Web.Factory.AbstractFactory
8  {
9      public class Dell : IBrand
10     {
11         public string GetBrand()
12         {
13             return Brands.Dell.ToString();
14         }
15     }
16 }
17 }
18 }
```

Solution Explorer

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
 - AccountController.cs
 - BaseController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ManageController.cs
- Factory
 - AbstractFactory
 - Dell.cs
 - Enumerations.cs
 - IBrand.cs
 - ISystemType.cs
 - Mac.cs
 - Processor.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - fonts
 - Manager

Properties

Search Solution Explorer (Ctrl+Shift+F)

121 % Output Error List... Ln 14 Col 43 Ch 43 INS Ready Publish 60% CPU

بنضيف كلاس لماركة الكمبيوتر

Dell

ونكمل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Dell.cs Mac.cs Processor.cs ISystemType.cs Enumerations.cs IBrand.cs

Web Dell.cs Web.Factory.AbstractFactory.Dell GetBrand()

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using static Web.Factory.AbstractFactory.Enumerations;
6
7 namespace Web.Factory.AbstractFactory
8 {
9     public class Dell : IBrand
10    {
11        public string GetBrand()
12        {
13            return Enumerations.Brands.Dell.ToString();
14        }
15    }
16}
```

121% Output Error List... Item(s) Saved

Quick Launch (Ctrl+Q) Sign in Properties

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
 - AccountController.cs
 - BaseController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ManageController.cs
- Factory
 - AbstractFactory
 - Dell.cs
 - Enumerations.cs
 - IBrand.cs
 - ISystemType.cs
 - Mac.cs
 - Processor.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - fonts
- Manager

Solution Explorer Class View 63% CPU

BISHOYNABIL

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

I7.cs Web Web.Factory.AbstractFactory.I7 GetProcessor()

```
8
9
10    public class I7 : IProcessor
11    {
12        public string GetProcessor()
13        {
14            return Processors.I7.ToString();
15        }
16    }
17
18    public class I5 : IProcessor
19    {
20        public string GetProcessor()
21        {
22            return Processors.I5.ToString();
23        }
24    }
25}
```

121% Output Error List... Item(s) Saved

Quick Launch (Ctrl+Q) Sign in Properties

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
 - AccountController.cs
 - BaseController.cs
 - EmployeesController.cs
 - HomeController.cs
 - ManageController.cs
- Factory
 - AbstractFactory
 - Dell.cs
 - Enumerations.cs
 - I7.cs
 - IBrand.cs
 - ISystemType.cs
 - Mac.cs
 - Processor.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - fonts
- Manager

Solution Explorer Class View 58% CPU

هنا بعمل كلاس

I7

وبرجع فيه الميثود I7 وبنزل تحت اعمل كلاس اسمه

I5

وبرجع فيه الميثود ال I5

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Laptop.cs  I7.cs
Web
2
3     using System.Collections.Generic;
4     using System.Linq;
5     using System.Web;
6     using static Web.Factory.AbstractFactory.Enumerations;
7
8     namespace Web.Factory.AbstractFactory
9     {
10        public class Laptop : ISystemType
11        {
12            public string GetSystemType()
13            {
14                return ComputerTypes.Laptop.ToString();
15            }
16        }
17        public class Desktop : ISystemType
18        {
19            public string GetSystemType()
20            {
21                return ComputerTypes.Desktop.ToString();
22            }
23        }
}
Solution Explorer
App_Data
EmployeePortal.mdf
App_Start
Content
Controllers
AccountController.cs
BaseController.cs
EmployeesController.cs
HomeController.cs
ManageController.cs
Factory
AbstractFactory
Dell.cs
Enumerations.cs
I7.cs
IBrand.cs
ISystemType.cs
Laptop.cs
Mac.cs
Processor.cs
FactoryMethod
BaseEmployeeFactory.cs
ContractEmployeeFactory.cs
EmployeeManagerFactory.cs
PermanentEmployeeFactory.cs
EmployeeManagerFactory.cs
Properties
57% CPU
```

بضيف كلاس
عشان احيب اجهزة
اللاب توب والديسك توب
نكملي زى الشرح هنا بضيف انترفيس

فِيَهَا الْبَرَانَدُ وَالْبُرُوسِيْسُورُ وَنَوْعُ السِّيْسِتَم

```

3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Web.Factory.AbstractFactory
8  {
9      public interface IcomputerFactory
10     {
11         IPProcessor Processor();
12
13         IBrand Brand();
14
15         ISysteMType SystemType();
16     }
17 }
18

```

بَنَعْمَلْ كَلَاسْ جَدِيدْ وَبَنْحَطْ فِيَهَا الْاِنْتَرْفَسْ

```

5  using static Web.Factory.AbstractFactory.Laptop;
6
7  namespace Web.Factory.AbstractFactory
8  {
9      public class MacFactory : IcomputerFactory
10     {
11         public IBrand Brand()
12         {
13             return new Mac();
14         }
15
16         public IPProcessor Processor()
17         {
18             return new I7();
19         }
20
21         public ISysteMType SystemType()
22         {
23             return new Desktop();
24         }
25     }
26 }

```

ونكمل

```
13     return new Mac();
14 }
15
16     1 reference
17     public IPProcessor Processor()
18     {
19         return new I7();
20     }
21
22     2 references
23     public virtual ISystemType SystemType()
24     {
25         return new Desktop();
26     }
27
28     0 references
29     public class MacLaptopFactory : MacFactory{
30
31         2 references
32         public override ISystemType SystemType()
33         {
34             return new Laptop();
35         }
36     }
37 }
```

هنا بضيف كلاس جديد وخليه ينوري الماك فكتوري وقبلها بطلع اغير ميثود ال

ISystemType

باضفة كملة

virtual

وبرجع للكلاس بتاعنا بحط فيه ميثود بتعمل اوفراريد للسيستم تايب

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
EmployeeSystemFactory.cs DellFactory.cs MacFactory.cs IcomputerFactory.cs Laptop.cs I7.cs
Web Web.Factory.AbstractFactory.EmployeeSystemFactory Create(Employee emp)
public class EmployeeSystemFactory
{
    public IcomputerFactory Create(Employee emp)
    {
        IcomputerFactory returnValue = null;

        if (emp.EmployeeTypeID == 1)
        {

            if(emp.JobDescription == "Manager")
            {
                returnValue = new MacLaptopFactory();
            }
            else{
                returnValue = new MacFactory();
            }
        }

        else if (emp.EmployeeTypeID ==3 )
        if(emp.JobDescription == "Manager")
        {
            returnValue = new DellLaptopFactory();
        }
        else
        {
            returnValue = new DellFactory();
        }
    }

    return returnValue;
}
```

يعمل كلاس جديد وبصيغ فيه الميثود الموضحة
يعمل كلاس تانى هيعمل
تشغيل للكلاس اللي اسمه
EmployeeSystemFactory

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Debug Any CPU Start Properties
EmployeeSystemManager.cs EmployeeSystemFactory.cs DellFactory.cs MacFactory.cs IcomputerFactory.cs Laptop.cs I7.cs
Solution Explorer
Search Solution Explorer (Ctrl+Shift+F)
Solution 'EmployeePortal' (2 projects)
Logger
Properties
References
ILog.cs
Log.cs
Web
Properties
References
App_Data
EmployeePortal.mdf
App_Start
Content
Controllers
Factory
AbstractFactory
Dell.cs
DellFactory.cs
EmployeeSystemFactory.cs
EmployeeSystemManager.cs
Enumerations.cs
I7.cs
IBrand.cs
IcomputerFactory.cs
ISystemType.cs
Laptop.cs
Solution Explorer Class View
CPU 52%
Item(s) Saved
Ln 24 Col 16 Ch 16 INS
Output Error List
```

```
EmployeeSystemManager.cs
namespace Web.Factory.AbstractFactory
{
    public class EmployeeSystemManager
    {
        IcomputerFactory _IcomputerFactory;

        public EmployeeSystemManager(IcomputerFactory icomputerFactory)
        {
            _IcomputerFactory = icomputerFactory;
        }

        public string GetSystemDetails()
        {
            IBrand brand = _IcomputerFactory.Brand();
            IProcessor processor = _IcomputerFactory.Processor();
            ISystemType systemType = _IcomputerFactory.SystemType();
            string returnValue = string.Format("{0},{1},{2}",
                brand.GetType(), processor.GetProcessor(), systemType.GetSystemType());
            return returnValue;
        }
    }
}
```

BISHOYNABIL

تکمل هنبدی نقسم البروچیکت لفولدرات زی الشرح ونوزع الكلاسات عشان الشرح
يوضخ

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

Server Explorer Toolbox

Web

```
1 namespace Web.Factory.AbstractFactory
2 {
3     public interface IProcessor
4     {
5         string GetProcessor();
6     }
7 }
```

Solution Explorer

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
- Factory
 - AbstractFactory
 - AbstractProduct
 - IBrand.cs
 - IProcessor.cs
 - ISystemType.cs
 - Dell.cs
 - DellFactory.cs
 - EmployeeSystemFactory.cs
 - EmployeeSystemManager.cs
 - Enumerations.cs
 - I7.cs
 - IcomputerFactory.cs
 - Laptop.cs
 - Mac.cs
 - MacFactory.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs
 - PermanentEmployeeFactory.cs

Properties

Quick Launch (Ctrl+Q) Sign in

120 % Output Error List..

This item does not support previewing

Ln 1 Col 1 Ch 1 INS

44% CPU

BISHOYNABIL

عملت اول فولدر ونقلت فيه الكلاسات ونكم

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

Server Explorer Toolbox

Web

```
10 public class I7 : IProcessor
11 {
12     public string GetProcessor()
13     {
14         return Processors.I7.ToString();
15     }
16
17     public class I5 : IProcessor
18     {
19         public string GetProcessor()
20         {
21             return Processors.I5.ToString();
22         }
23     }
24 }
25
26 }
27 }
```

Solution Explorer

- App_Data
- EmployeePortal.mdf
- App_Start
- Content
- Controllers
- Factory
 - AbstractFactory
 - AbstractProduct
 - IBrand.cs
 - IProcessor.cs
 - ISystemType.cs
 - ConConcreteProduct
 - Dell.cs
 - Mac.cs
 - DellFactory.cs
 - EmployeeSystemFactory.cs
 - EmployeeSystemManager.cs
 - Enumerations.cs
 - I7.cs
 - IcomputerFactory.cs
 - Laptop.cs
 - MacFactory.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs
 - EmployeeManagerFactory.cs

Properties

Quick Launch (Ctrl+Q) Sign in

120 % Output Error List..

This item does not support previewing

Ln 25 Col 1 Ch 1 INS

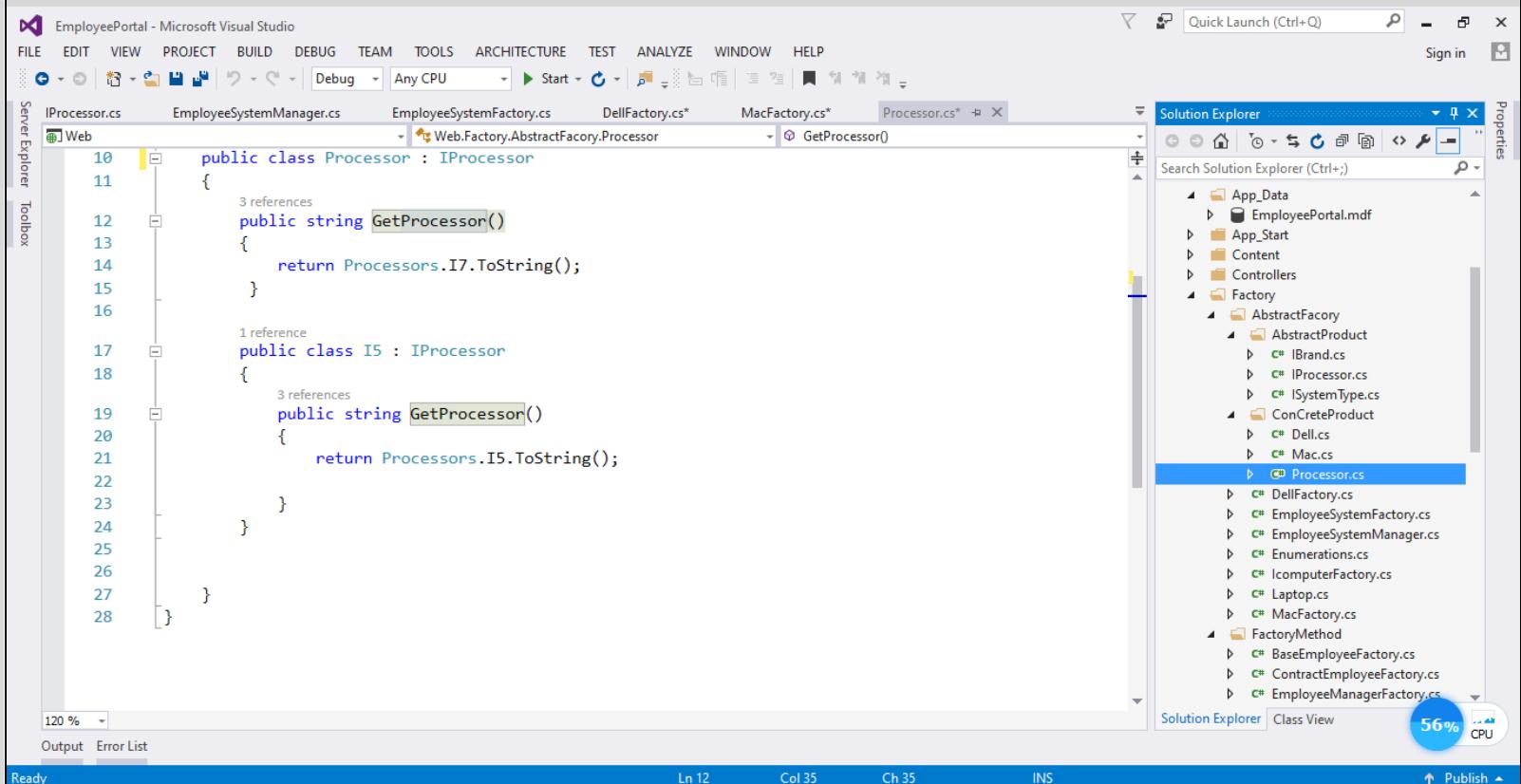
54% CPU

بضييف الفولدر وانقل الكلاسات باقى كلاس اسمه

I7

هنجير اسمه ليكون

Processor



```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
... Debug Any CPU Start ...
IProcessor.cs EmployeeSystemManager.cs EmployeeSystemFactory.cs DellFactory.cs* MacFactory.cs* Processor.cs* GetProcessor()
Server Explorer Toolbox Solution Explorer Properties
Search Solution Explorer (Ctrl+;)
App_Data EmployeePortal.mdf App_Start Content Controllers Factory
AbstractFactory IBrand.cs IProcessor.cs ISystemType.cs
ConCreteProduct Dell.cs Mac.cs Processor.cs
DellFactory.cs EmployeeSystemFactory.cs EmployeeSystemManager.cs Enumerations.cs
IcomputerFactory.cs Laptop.cs MacFactory.cs
FactoryMethod BaseEmployeeFactory.cs ContractEmployeeFactory.cs EmployeeManagerFactory.cs
56% CPU
10 public class Processor : IProcessor
11 {
12     3 references
13     public string GetProcessor()
14     {
15         return Processors.I7.ToString();
16     }
17     1 reference
18     public class I5 : IProcessor
19     {
20         3 references
21         public string GetProcessor()
22         {
23             return Processors.I5.ToString();
24         }
25     }
26 }
27 }
28 }
```

120 % Output Error List

Ready Ln 12 Col 35 Ch 35 INS ↑ Publish ▾

تمام نكملي

```

using static Web.Factory.AbstractFactory.Enumerations;

namespace Web.Factory.AbstractFactory
{
    public class Laptop : ISystemType
    {
        public string GetsSystemType()
        {
            return ComputerTyeps.Laptop.ToString();
        }
    }

    public class Desktop : ISystemType
    {
        public string GetSystemType()
        {
            return ComputerTyeps.Desktop.ToString();
        }
    }
}

```

The Solution Explorer shows the project structure:

- Content
- Controllers
- Factory
 - AbstractFactory
 - AbstractInterface.cs
 - IcomputerFactory.cs
 - AbstractProduct.cs
 - IBrand.cs
 - IProcessor.cs
 - ISystemType.cs
 - Client
 - EmployeeSystemManager.cs
 - ConCreteFactory
 - DellFactory.cs
 - EmployeeSystemFactory.cs
 - MacFactory.cs
 - ConCreteProduct
 - ComputerSystemType.cs
 - Dell.cs
 - Mac.cs
 - Processor.cs
 - Enumerations.cs
 - Laptop.cs
 - FactoryMethod
 - BaseEmployeeFactory.cs
 - ContractEmployeeFactory.cs

بص الترتيب النهائي
فيه اىرور فى الكلاس ده بسب انى مكرر كلاس

Laptop

همسح كلاس اللاب توب واسيب كلاس

Computersystemtype

اللى هو فعليا اسمه

Laptop

مشكله تكرار اسم وتكرار كود المهم همسح الكلاس اللي اسم الملف بتاعه

Laptop

نكملي هنضيف حقل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Server Explorer
Toolbox

dbo.Employee [Design] Index.cshtml ComputerSystemType.cs IcomputerFactory.cs Processor.cs IProcessor.cs

Update Script File: dbo.Employee.sql*

Name	Data Type	Allow Nulls	Default
Bonus	decimal(18,0)	<input checked="" type="checkbox"/>	
EmployeeTypeID	int	<input checked="" type="checkbox"/>	
HouseAllowance	decimal(18,0)	<input checked="" type="checkbox"/>	
MedicalAllowance	decimal(18,0)	<input checked="" type="checkbox"/>	
ComputerDetails	varchar(250)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Keys (1)
<unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
FK_Employee_EmployeeType (Id)
Triggers (0)

Design T-SQL

```
8    [Bonus]      DECIMAL (18) NULL,
9    [EmployeeTypeID] INT      NULL,
10   [HouseAllowance] DECIMAL (18) NULL,
11   [MedicalAllowance] DECIMAL (18) NULL,
```

Connection Ready | (LocalDB)\MSSQLLocalDB | DESKTOP-AVE6BFU\bishoe | C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL.MDF

Data Tools Operations

Update for (LocalDB)\MSSQLLocalDB.C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL.MDF 4:54:1 Cancel
✓ Creating update preview...
✓ Displaying update preview...
✓ Creating database script...
✓ Executing update script on database 'C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL.... View Script
Update complete. View Results

Error List...

Solution Explorer Class View 57% CPU

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Server Explorer
Toolbox

EmployeesController.cs Employee.cs Index.cshtml IcomputerFactory.cs

Web

```
50 // POST: Employees/Create
51 // To protect from overposting attacks, please enable the specific properties you want to bind.
52 // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
53 [HttpPost]
54 [ValidateAntiForgeryToken]
55 public ActionResult Create([Bind(Include = "Id,Name,JobDescription,Number,Department,HourlyPay")]
56 {
57     if (ModelState.IsValid)
58     {
59         BaseEmployeeFactory empFactory = new EmployeeManagerFactory().CreateFactory(employee);
60         //IEmployeeManager empManager = empFactory.GetEmployeeManager(employee.EmployeeTypeID);
61         //employee.Bonus = empManager.GetBonus();
62         //employee.HourlyPay = empManager.GetPay();
63         empFactory.ApplySalary();
64
65         IcomputerFactory factory = new EmployeeSystemFactory().Create(employee);
66
67         EmployeeSystemManager manager = new EmployeeSystemManager(factory);
68
69         employee.ComputerDetails = manager.GetSystemDetails();
70
71         db.Employees.Add(employee);
72         db.SaveChanges();
73         return RedirectToAction("Index");
74     }
75 }
```

Output Error List...

Solution Explorer Class View 65% CPU

بعد علی الکنترولار ونشغل ونجرب وربنا یسستر
آخر تعديل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

EmployeeSystemFactory.cs DellFactory.cs EmployeeSystemManager.cs ISystemType.cs IProcessor.cs IBrand.cs

Web Web.Factory.AbstractFactory.EmployeeSystemManager GetSystemDetails()

IComputerFactory _IcomputerFactory;

1 reference

public EmployeeSystemManager(IComputerFactory icomputerFactory)

{

 _IcomputerFactory = icomputerFactory;

}

1 reference

public string GetSystemDetails()

{

 IBrand brand = _IcomputerFactory.Brand();

 IProcessor processor = _IcomputerFactory.Processor();

 ISystemType systemType = _IcomputerFactory.SystemType();

 string returnValue = string.Format("{0},{1},{2}",

 brand.GetBrand(), processor.GetProcessor(), systemType.GetSystemType());

 return returnValue;

}

}

120 %

Output Error List

Ready

Ln 23 Col 25 Ch 25 INS

Solution Explorer

Search Solution Explorer (Ctrl+F)

- AbstractInterface
- AbstractProduct
- IBrand
- IProcessor
- ISystemType
- Client
- EmployeeSystemManager.cs
- ConCreteFactory
- DellFactory.cs
- EmployeeSystemFactory.cs
- MacFactory.cs
- ConCreteProduct
- ComputerSystemType.cs
- Dell.cs
- Mac.cs
- Processor.cs
- Enumerations.cs
- FactoryMethod
- BaseEmployeeFactory.cs
- ContractEmployeeFactory.cs
- EmployeeManagerFactory.cs
- PermanentEmployeeFactory.cs
- EmployeeManagerFactory.cs
- fonts
- Manager

Class View

57% CPU

هنا أنا كنت عاملها
BISHOYNABIL
Gettytype

وده غلط الصح

Grtbrand

Screenshot of an ASP.NET application showing a Create Employee form.

The browser address bar shows: http://localhost:49936/employees/Create

The page title is: Create - My ASP.NET Appli... (with a yellow arrow pointing to the 'Create' part)

Navigation links: Application name, Home, About, Contact, Register, Log in

Create

Employee

Name	<input type="text" value="beko"/>
JobDescription	<input type="text" value="manager"/> (Yellow arrow points here)
Number	<input type="text" value="C109"/>
Department	<input type="text" value="IT"/>
EmployeeTypeID	<input type="text" value="Permanent"/> (Yellow arrow points here)

[Back to List](#)

© 2019 - My ASP.NET Application

Do you want AutoComplete to remember web form entries? [Learn about AutoComplete](#)

Yes No X

59% CPU

BISHOYNABIL

خلفي بالك لازم ال

Jobdesc

يكون مدير

Manager

تكتب بنفس الشكل عشان حساسية الاحرف

Screenshot of a web browser showing an ASP.NET application's index page. The URL is http://localhost:49936/employees. The page displays a table of employee data with columns: Name, JobDescription, Number, Department, HourlyPay, Bonus, EmployeeType, HouseAllowance, MedicalAllowance, and ComputerDetails. The last row shows an entry for 'ZA' with 'Manager' as the job description, 'C@Manager' as the number, and 'it' as the department. The 'ComputerDetails' column contains the value 'Apple,I7,Laptop'. Two yellow arrows point upwards from the bottom of the page towards this row.

Name	JobDescription	Number	Department	HourlyPay	Bonus	EmployeeType	HouseAllowance	MedicalAllowance	ComputerDetails
Jhon	Manager	A123	It	8.00	10.00	Permanent			Edit Details Delete
Smith	Contract Dev	c1234	It			Contract			Edit Details Delete
BISHOE	HR	C109	Iti	8.00	10.00	Permanent			Edit Details Delete
asd	sad	asd	asd	12.00	5.00	Contract			Edit Details Delete
alex	dev	T1909	it	8.00	10.00	Permanent	150.00		Edit Details Delete
mary	Qa	Q1040	it	12.00	5.00	Contract	100.00		Edit Details Delete
Employee	Employee	Employee	Employee	12.00	5.00	Contract	100.00	Dell,I5,Desktop	Edit Details Delete
ZA	Manager	C@Manager	it	8.00	10.00	Permanent	150.00	Apple,I7,Laptop	Edit Details Delete

© 2019 - My ASP.NET Application

Screenshot of a web browser showing the same ASP.NET application's index page. The URL is http://localhost:49936/employees. The page displays a table of employee data, identical to the first screenshot. The last row shows an entry for 'ZA' with 'Manager' as the job description, 'C@Manager' as the number, and 'it' as the department. The 'ComputerDetails' column contains the value 'Apple,I7,Laptop'. A single yellow arrow points upwards from the bottom of the page towards this row.

Name	JobDescription	Number	Department	HourlyPay	Bonus	EmployeeType	HouseAllowance	MedicalAllowance	ComputerDetails
Jhon	Manager	A123	It	8.00	10.00	Permanent			Edit Details Delete
Smith	Contract Dev	c1234	It			Contract			Edit Details Delete
BISHOE	HR	C109	Iti	8.00	10.00	Permanent			Edit Details Delete
asd	sad	asd	asd	12.00	5.00	Contract			Edit Details Delete
alex	dev	T1909	it	8.00	10.00	Permanent	150.00		Edit Details Delete
mary	Qa	Q1040	it	12.00	5.00	Contract	100.00		Edit Details Delete
Employee	Employee	Employee	Employee	12.00	5.00	Contract	100.00	Dell,I5,Desktop	Edit Details Delete
ZA	Manager	C@Manager	it	8.00	10.00	Permanent	150.00	Apple,I7,Laptop	Edit Details Delete
Nader	Hr	C107	CHR	8.00	10.00	Permanent	150.00	Apple,I7/Desktop	Edit Details Delete

© 2019 - My ASP.NET Application

نجح اى موظف اخر بنديله ديسكتوب لكن المدير ياخد لاب توب بالنسبة لل
Perm

لکن ال
Contract
نجرب

BISHOYNABIL

http://localhost:49936/employees/Create

Create - My ASP.NET Appl... X

Application name Home About Contact Register Log in

Create

Employee

Name	<input type="text" value="sara"/>
JobDescription	<input type="text" value="Manager"/> 
Number	<input type="text" value="C17"/>
Department	<input type="text" value="Manager"/>
EmployeeTypeID	<input type="text" value="Contract"/> 
	<input type="button" value="Create"/>

[Back to List](#)

© 2019 - My ASP.NET Application

60% CPU

http://localhost:49936/employees

Index - My ASP.NET Appl... X

Application name Home About Contact Register Log in

Jhon	Manager	A123	It	8.00	10.00	Permanent	Edit Details Delete	
Smith	Contract Dev	c1234	It			Contract	Edit Details Delete	
BISHOE	HR	C109	Iti	8.00	10.00	Permanent	Edit Details Delete	
asd	sad	asd	asd	12.00	5.00	Contract	Edit Details Delete	
alex	dev	T1909	it	8.00	10.00	Permanent	150.00	Edit Details Delete
mary	Qa	Q1040	it	12.00	5.00	Contract	100.00	Edit Details Delete
Employee	Employee	Employee	Employee	12.00	5.00	Contract	100.00	Dell,I5/Desktop Edit Details Delete
ZA	Manager	C@Manager	it	8.00	10.00	Permanent	150.00	Apple,I7,Laptop Edit Details Delete
Nader	Hr	C107	CHR	8.00	10.00	Permanent	150.00	Apple,I7/Desktop Edit Details Delete
sara	Manager	C0	Manager	12.00	5.00	Contract	100.00	Dell,I5,Laptop    Edit Details Delete

© 2019 - My ASP.NET Application

69% CPU

http://localhost:49936/employees/Create

Create - My ASP.NET Appl... X

Application name Home About Contact Register Log in

Create

Employee

Name	<input type="text" value="John"/>
JobDescription	<input type="text" value="hr"/>
Number	<input type="text" value="H1009"/>
Department	<input type="text" value="Hr"/>
EmployeeTypeID	<input style="background-color: #0078D4; color: white;" type="text" value="Contract"/>
<input type="button" value="Create"/>	

[Back to List](#)

© 2019 - My ASP.NET Application

74% CPU

http://localhost:49936/employees

Index - My ASP.NET Appl... X

Application name Home About Contact Register Log in

Smith	Contract Dev	c1234	It	Contract					
BISHOE	HR	C109	It	8.00	10.00	Permanent			Edit Details Delete
asd	sad	asd	asd	12.00	5.00	Contract			Edit Details Delete
alex	dev	T1909	it	8.00	10.00	Permanent	150.00		Edit Details Delete
mary	Qa	Q1040	it	12.00	5.00	Contract	100.00		Edit Details Delete
Employee	Employee	Employee	Employee	12.00	5.00	Contract	100.00	Dell,I5/Desktop	Edit Details Delete
ZA	Manager	C@Manager	it	8.00	10.00	Permanent	150.00	Apple,I7,Laptop	Edit Details Delete
Nader	Hr	C107	CHR	8.00	10.00	Permanent	150.00	Apple,I7/Desktop	Edit Details Delete
sara	Manager	C0	Manager	12.00	5.00	Contract	100.00	Dell,I5,Laptop	Edit Details Delete
John	hr	H1009	Hr	12.00	5.00	Contract	100.00	Dell,I5/Desktop	Edit Details Delete

© 2019 - My ASP.NET Application

76% CPU

نجحنا

Builder

هو يعتبر

creational pattern

يحلل مشاكل لما يكون الكونستركتور بارمتر في زيادة مستمرة و معروف
الكونستركتور في كل مرة بتحتاجه عشان نعمل

Initialize

لكن لما يكون عندك كومبليكس او بجيكتس عدد البرامتر اللي هتتبع للكلاس بتاعنا
بتزيد كل شوية وفي النهاية بترجملك

BISHOYNABIL

كومبليكس او بجيكت

والبيلدر دايزين بتحلل المشكلة دي

Complex object

أوجيكت مركب بمعنى يحتوى على متغيرات او ميثود مركبة بص زي الاوجيكت يكون فيه
الاسم والنوع والوظيفة والعنوان والفون وهكذا

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Server Explorer Toolbox Solution Explorer Properties

dbo.Employee [Design] EmployeeSystemFactory.cs DellFactory.cs EmployeeSystemManager.cs ISystemType.cs IProcessor.cs

Update Script File: dbo.Employee.sql

Name	Data Type	Allow Nulls	Default
Name	varchar(50)	<input type="checkbox"/>	
JobDescription	varchar(50)	<input type="checkbox"/>	
Number	varchar(50)	<input type="checkbox"/>	
Department	varchar(50)	<input type="checkbox"/>	
HourlyPay	decimal(18,0)	<input checked="" type="checkbox"/>	
Bonus	decimal(18,0)	<input checked="" type="checkbox"/>	
FmnlEmployeeID	int	<input checked="" type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
 FK_Employee_EmployeeType (Id)
Triggers (0)

T-SQL

```

12 [ComputerDetails] VARCHAR (250) NULL,
13 [SystemConfigurationDetails] VARCHAR(500) NULL, ← Yellow arrow
14 PRIMARY KEY CLUSTERED ([Id] ASC),
15 CONSTRAINT [FK_Employee_EmployeeType] FOREIGN KEY ([EmployeeTypeID]) REFERENCES [dbo].[Employee_Type] ← Yellow arrow
  
```

Connection Ready | (LocalDB)\MSSQLLocalDB | DESKTOP-AVE6BFU\bishoe | C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL.MDF 2:55:55 PM

Data Tools Operations

- ✓ Update for (LocalDB)\MSSQLLocalDB.C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL.MDF 2:55:55 PM
- ✓ Creating update preview...
- ✓ Displaying update preview...
- ✓ Creating database script...
- ✓ Executing update script on database 'C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL....' View Script View Results

Output Error List

61% CPU

Solution Explorer Class View

BISHOYNABIL

نكمـل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Server Explorer Toolbox Solution Explorer Properties

EmployeesController.cs Web Web.Controllers.EmployeesController BuildSystem(int? employeeID)

```

16 {
17     public class EmployeesController : BaseController
18     {
19         private EmployeePortalEntities db = new EmployeePortalEntities();
20
21         [HttpGet] ← Yellow arrow
22         public ActionResult BuildSystem(int? employeeID)
23         {
24
25             return View(employeeID); ← Yellow arrow
26         }
27
28         // GET: Employees
29     }
  
```

Data Tools Operations

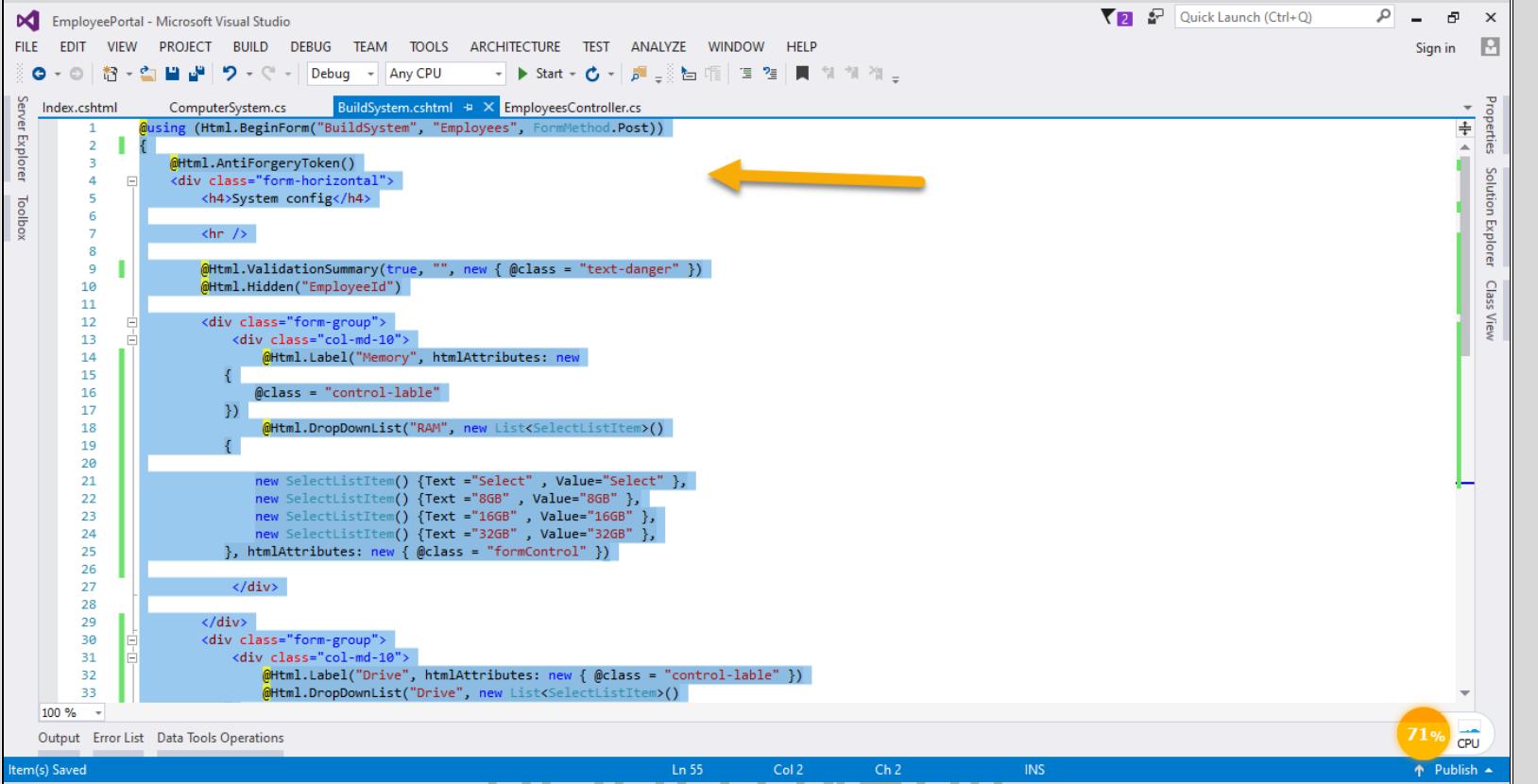
- ✓ Update for (LocalDB)\MSSQLLocalDB.C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL.MDF 2:55:55 PM
- ✓ Creating update preview...
- ✓ Displaying update preview...
- ✓ Creating database script...
- ✓ Executing update script on database 'C:\USERS\BISHOE\DOCUMENTS\VISUAL STUDIO 2015\PROJECTS\EMPLOYEEPORTAL\WEB\APP_DATA\EMPLOYEEPORTAL....' View Script View Results

Output Error List

64% CPU

Solution Explorer Class View

بصيغه الاكشن ده واروح اعمله فيو



EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Server Explorer Toolbox

Index.cshtml ComputerSystem.cs BuildSystem.cshtml EmployeesController.cs

```
1 @using (Html.BeginForm("BuildSystem", "Employees", FormMethod.Post)) {  
2     @Html.AntiForgeryToken()  
3     <div class="form-horizontal">  
4         <h4>System config</h4>  
5  
6             <hr />  
7  
8             @Html.ValidationSummary(true, "", new { @class = "text-danger" })  
9             @Html.Hidden("EmployeeId")  
10  
11            <div class="form-group">  
12                <div class="col-md-10">  
13                    @Html.Label("Memory", htmlAttributes: new  
14                    {  
15                        @class = "control-label"  
16                    })  
17                    @Html.DropDownList("RAM", new List<SelectListItem>()  
18                    {  
19  
20                        new SelectListItem() {Text = "Select", Value="Select"},  
21                        new SelectListItem() {Text = "8GB", Value="8GB"},  
22                        new SelectListItem() {Text = "16GB", Value="16GB"},  
23                        new SelectListItem() {Text = "32GB", Value="32GB"},  
24                        new SelectListItem() {Text = "64GB", Value="64GB"},  
25                    }, htmlAttributes: new { @class = "formControl" })  
26  
27                </div>  
28            </div>  
29            <div class="form-group">  
30                <div class="col-md-10">  
31                    @Html.Label("Drive", htmlAttributes: new { @class = "control-label" })  
32                    @Html.DropDownList("Drive", new List<SelectListItem>())  
33            </div>  
34        </div>  
35    </div>  
36}
```

100 %

Output Error List Data Tools Operations

Item(s) Saved

Ln 55 Col 2 Ch 2 INS

71% CPU

```
@using (Html.BeginForm("BuildSystem", "Employees", FormMethod.Post))  
{  
    @Html.AntiForgeryToken()  
    <div class="form-horizontal">  
        <h4>System config</h4>  
  
        <hr />  
  
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })  
        @Html.Hidden("EmployeeId")  
  
        <div class="form-group">  
            <div class="col-md-10">  
                @Html.Label("Memory", htmlAttributes: new  
                {  
                    @class = "control-label"  
                })  
                @Html.DropDownList("RAM", new List<SelectListItem>()  
                {  
  
                    new SelectListItem() {Text = "Select", Value="Select"},  
                    new SelectListItem() {Text = "8GB", Value="8GB"},  
                    new SelectListItem() {Text = "16GB", Value="16GB"},  
                    new SelectListItem() {Text = "32GB", Value="32GB"},  
                    new SelectListItem() {Text = "64GB", Value="64GB"},  
                }, htmlAttributes: new { @class = "formControl" })  
            </div>  
        </div>  
    </div>  
}
```

```
    new SelectListItem() {Text = "32GB" , Value="32GB" },
}, htmlAttributes: new { @class = "formControl" })

</div>

</div>
<div class="form-group">
<div class="col-md-10">
    @Html.Label("Drive", htmlAttributes: new { @class = "control-lable" })
    @Html.DropDownList("Drive", new List<SelectListItem>()
{
    new SelectListItem() {Text = "Select" , Value="Select" },
    new SelectListItem() {Text = "500GB" , Value="500GB" },
    new SelectListItem() {Text = "1TBGB" , Value="1TBGB" },
}, htmlAttributes: new { @class = "formControl" })

</div>

</div>
<div class="form-group">

<div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Create" class="btn btn-default" />
</div>

</div>
</div>
```

{}

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

Index.cshtml ComputerSystem.cs BuildSystem.cshtml EmployeesController.cs

Web ComputerSystem

```
7  namespace Web
8  {
9      public class ComputerSystem
10     {
11         private string _RAM;
12         private string _HDDSize;
13
14         public ComputerSystem()
15         {
16         }
17
18         public ComputerSystem(string RAM, string HDD)
19         {
20             _RAM = RAM;
21             _HDDSize = HDD;
22         }
23
24         public string Build()
25         {
26             StringBuilder sb = new StringBuilder();
27
28             sb.Append(string.Format(" RAM: {0}", _RAM));
29
30             sb.Append(string.Format(" HDDSize: {0}", _HDDSize));
31
32             return sb.ToString();
33         }
34     }
35 }
```

100% 56% CPU

Output Error List Data Tools Operations

Item(s) Saved Ln 36 Col 1 Ch 1 INS

BISHOY NABIL كلاس عمل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start

Index.cshtml ComputerSystem.cs BuildSystem.cshtml EmployeesController.cs

Web Web.Controllers.EmployeesController

```
17  public class EmployeesController : BaseController
18  {
19      private EmployeePortalEntities db = new EmployeePortalEntities();
20
21      [HttpGet]
22      public ActionResult BuildSystem(int? employeeID)
23      {
24
25          return View(employeeID);
26      }
27      [HttpPost]
28      public ActionResult BuildSystem(int employeeID, string RAM, string HDDSize)
29      {
30          Employee employee = db.Employees.Find(employeeID);
31          ComputerSystem computerSystem = new ComputerSystem(RAM, HDDSize);
32          employee.SystemConfigurationDetails = computerSystem.Build();
33          db.Entry(employee).State = EntityState.Modified;
34          db.SaveChanges();
35          return RedirectToAction("Index");
36      }
37
38      // GET: Employees
39      public ActionResult Index()
40      {
41          var employees = db.Employees.Include(e => e.Employee_Type);
42          return View(employees.ToList());
43      }
44
45      // GET: Employees/Details/5
```

100% 65% CPU

Output Error List Data Tools Operations

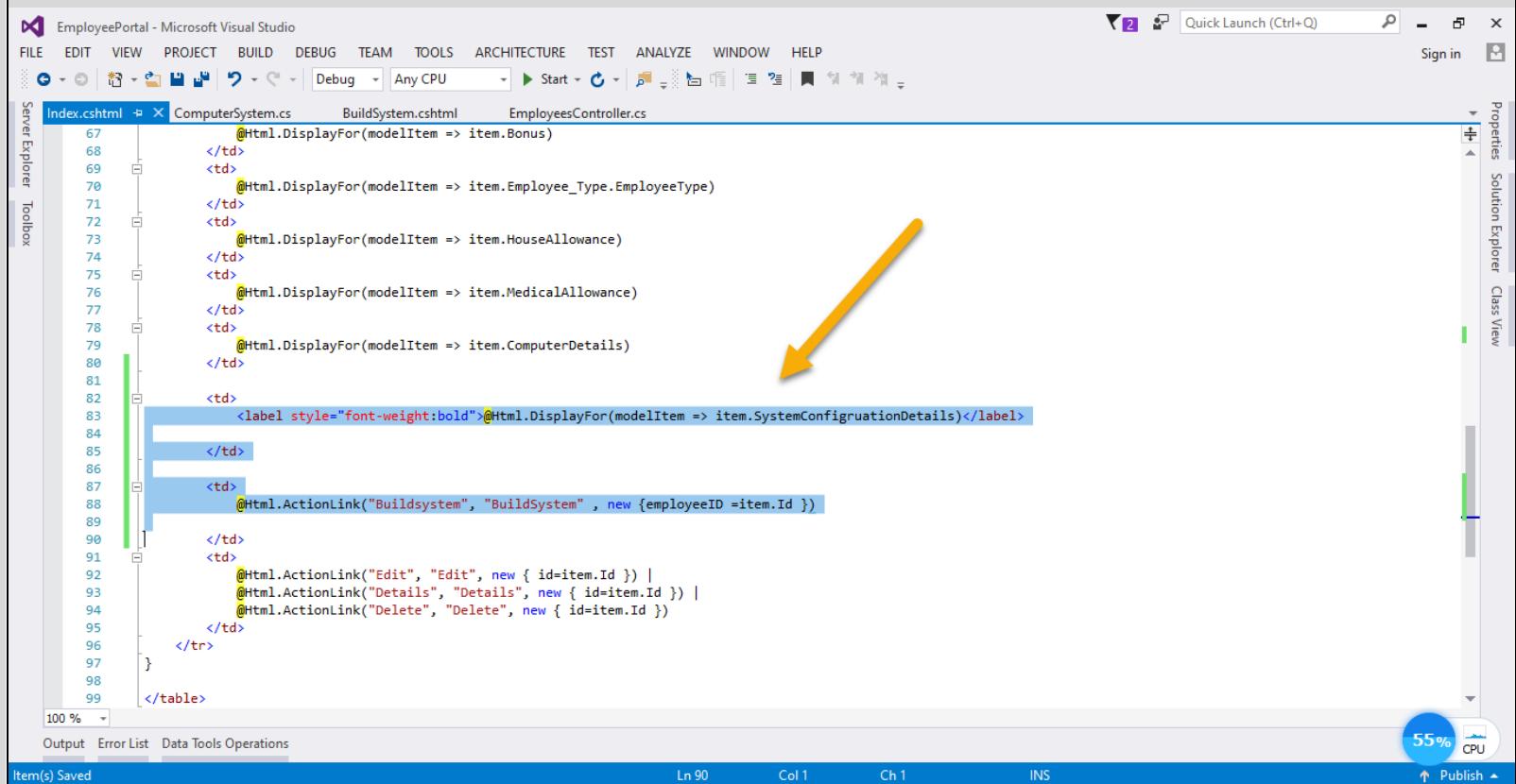
Item(s) Saved Ln 32 Col 48 Ch 48 INS

بضيف ميثود لـ

Get

9

Post



EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Index.cshtml ComputerSystem.cs BuildSystem.cshtml EmployeesController.cs

```
67     @Html.DisplayFor(modelItem => item.Bonus)
68     </td>
69     <td>
70         @Html.DisplayFor(modelItem => item.Employee_Type.EmployeeType)
71     </td>
72     <td>
73         @Html.DisplayFor(modelItem => item.HouseAllowance)
74     </td>
75     <td>
76         @Html.DisplayFor(modelItem => item.MedicalAllowance)
77     </td>
78     <td>
79         @Html.DisplayFor(modelItem => item.ComputerDetails)
80     </td>
81
82     <td>
83         <label style="font-weight:bold">@Html.DisplayFor(modelItem => item.SystemConfigruationDetails)</label>
84     </td>
85
86     <td>
87         @Html.ActionLink("Buildsystem", "BuildSystem" , new {employeeID =item.Id })
88
89     </td>
90
91     <td>
92         @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
93         @Html.ActionLink("Details", "Details", new { id=item.Id }) |
94         @Html.ActionLink("Delete", "Delete", new { id=item.Id })
95     </td>
96
97     </tr>
98 }
99 </table>
```

100 %

Output Error List Data Tools Operations

Item(s) Saved

Ln 90 Col 1 Ch 1 INS

55% CPU

↑ Publish ▾

نجرب

Screenshot of a web application showing system configuration settings:

URL: http://localhost:49936/Employees/BuildSystem?employeeID=1

Page Title: My ASP.NET Application

Header: Application name, Home, About, Contact, Register, Log in

Section: System config

Form fields:

- Memory: 8GB (dropdown)
- Drive: 500GB (dropdown)

Buttons:

- Create (highlighted with a yellow arrow)

Footer:

© 2019 - My ASP.NET Application

System status indicator: 78% CPU usage

Screenshot of a web application showing employee index and build system details:

URL: http://localhost:49936/Employees/Index

Page Title: Index - My ASP.NET Application

Header: Application name, Home, About, Contact, Register, Log in

Section: Index

Text: Create New

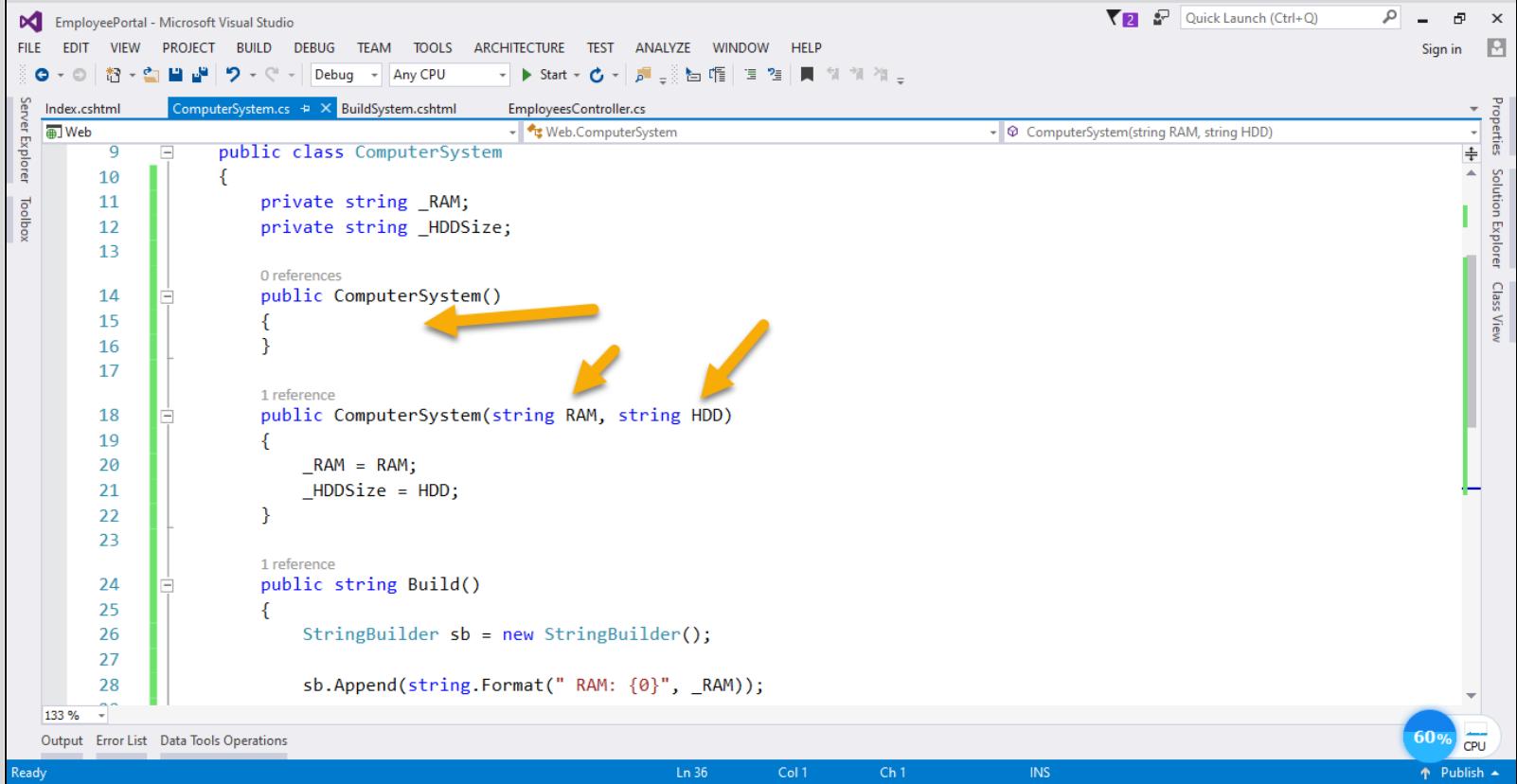
Name	JobDescription	Number	Department	HourlyPay	Bonus	EmployeeType	HouseAllowance	MedicalAllowance	ComputerDetails	RAM:	HDDSize:	Action
Jhon	Manager	A123	It	8.00	10.00	Permanent				8GB		Buildsystem Edit Details Delete
Smith	Contract Dev	c1234	It			Contract						Buildsystem Edit Details Delete
BISHOE	HR	C109	Iti	8.00	10.00	Permanent						Buildsystem Edit Details Delete
asd	sad	asd	asd	12.00	5.00	Contract						Buildsystem Edit Details
alex	dev	T1909	it	8.00	10.00	Permanent	150.00					Buildsystem Edit Details

System status indicator: 54% CPU usage

تمام

بس هنا عندنا مشكلة لو انا عندي كمبيوتر ديسك توب ومن خصائصه لازم له ماوس وكمبيوتر عايي لهم يكون خاصين بالديسك توب مش يكونوا خاصين بالديسك توب واللابتوب وبالحل اللي انا عامله فوق ده غير مفيد ليه لاني هضرط اروح لكلاس ال

Computer system



```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
... Debug Any CPU Start ...
Index.cshtml ComputerSystem.cs BuildSystem.cshtml EmployeesController.cs
Web ComputerSystem(string RAM, string HDD)
public class ComputerSystem
{
    private string _RAM;
    private string _HDDSize;

    0 references
    public ComputerSystem()
    {
    }

    1 reference
    public ComputerSystem(string RAM, string HDD)
    {
        _RAM = RAM;
        _HDDSize = HDD;
    }

    1 reference
    public string Build()
    {
        StringBuilder sb = new StringBuilder();

        sb.Append(string.Format(" RAM: {0}", _RAM));
    }
}
```

وازود كونستركتور كمان ببرامتر تانية كتير وده غلط

وفي الحالة دي هنحتاج

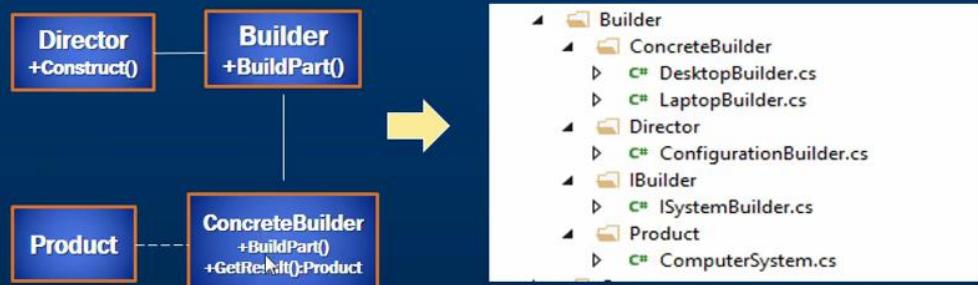
ال

Builder design

هنتطبقه في المثال القادر المثال اللي فوق لعرض المشكلة فقط اليك الحل

Builder Representation

Builder Design Pattern



Builder : Defines a template for the steps to construct the product. To simplify, Builder Specifies an abstract interface for creating parts of a Product object.

Concrete Builder : Implements the builder interface and provides an interface for getting the product.

Director : Is responsible to constructs the object through the builder interface.

Product : is the Main object that's constructed and Represents the complex object.

4

Builder Design Pattern Example

Step 1 : Add ISystemBuilder interface.

```

public interface ISystemBuilder
{
    void AddMemory(string memory);
    void AddDrive(string size);

    void AddKeyboard(string type);
    void AddMouse(string type);

    void AddTouchScreen(string enabled);
    ComputerSystem GetSystem();
}
  
```

5

Builder Design Pattern Example

Step 2 : Add desktop builder and inherit ISystemBuilder interface and implement the Interface methods and assign the properties to the computer system that is being built.

```
public class DesktopBuilder : ISystemBuilder
{
    ComputerSystem desktop = new ComputerSystem();
    public void AddDrive(string size)
    {
        desktop.HDDSize = size;
    }
    public void AddKeyBoard(string type)
    {
        desktop.KeyBoard = type;
    }
    public void AddMemory(string memory)
    {
        desktop.RAM = memory;
    }
    public void AddMouse(string type)
    {
        desktop.Mouse = type;
    }
    public void AddTouchScreen(string enabled)
    {
        return;
    }
    public ComputerSystem GetSystem()
    {
        return desktop;
    }
}
```

6

Builder Design Pattern Example

Step 3 : Add laptop builder and inherit ISystemBuilder interface and implement the Interface methods and assign the properties to the computer system that is being built.

```
public class LaptopBuilder : ISystemBuilder
{
    ComputerSystem laptop = new ComputerSystem();
    public void AddDrive(string size)
    {
        laptop.HDDSize = size;
    }
    public void AddKeyBoard(string type)
    {
        return;
    }
    public void AddMemory(string memory)
    {
        laptop.RAM = memory;
    }
    public void AddMouse(string type)
    {
        return;
    }
    public void AddTouchScreen(string enabled)
    {
        laptop.TouchScreen = enabled;
    }
    public ComputerSystem GetSystem()
    {
        return laptop;
    }
}
```

7

Builder Design Pattern Example

Step 4 : Add configuration builder class which is the director to build the system.

```
public class ConfigurationBuilder
{
    public void BuildSystem(ISystemBuilder systembuilder
                           , NameValueCollection collection)
    {
        systembuilder.AddDrive(collection["Drive"]);
        systembuilder.AddMemory(collection["RAM"]);
        systembuilder.AddMouse(collection["Mouse"]);
        systembuilder.AddKeyBoard(collection["Keyboard"]);
        systembuilder.AddTouchScreen(collection["TouchScreen"]);
    }
}
```

8

Builder Design Pattern Example

Step 5 : Remove the constructor with the parameters and change the private properties to public so that they can be initialized from the concrete builder classes.

```
public class ComputerSystem
{
    public string RAM { get; set; }
    public string HDDSize { get; set; }
    public string KeyBoard { get; set; }
    public string Mouse { get; set; }
    public string TouchScreen { get; set; }
    public ComputerSystem()
    {
    }
}
```

9

Builder Design Pattern Example

Step 6 : Enhance the BuildSystem action method from the previous session by returning desktop and laptop views based on the system allocated to the employees.

```
[HttpGet]
public ActionResult BuildSystem(int? employeeID)
{
    Employee employee = db.Employees.Find(employeeID);
    if (employee.ComputerDetails.Contains("Laptop"))
        return View("BuildLaptop", employeeID);
    else
        return View("BuildDesktop", employeeID);
}
```

10

Builder Design Pattern Example

Step 7 : Add Build laptop action method as shown below.

```
[HttpPost]
public ActionResult BuildLaptop(FormCollection formCollection)
{
    Employee employee =
        db.Employees.Find(Convert.ToInt32(formCollection["employeeID"]));
    //Concrete Builder
    ISystemBuilder systemBuilder = new LaptopBuilder();
    //Director
    ConfigurationBuilder builder = new ConfigurationBuilder();
    builder.BuildSystem(systemBuilder, formCollection);
    ComputerSystem system= systemBuilder.GetSystem();

    employee.SystemConfigurationDetails =
        string.Format("RAM : {0}, HDDSize : {1}, TouchScreen: {2}"
        , system.RAM, system.HDDSize, system.TouchScreen);

    db.Entry(employee).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

11

Builder Design Pattern Example

Step 8 : Add build desktop action method as shown below.

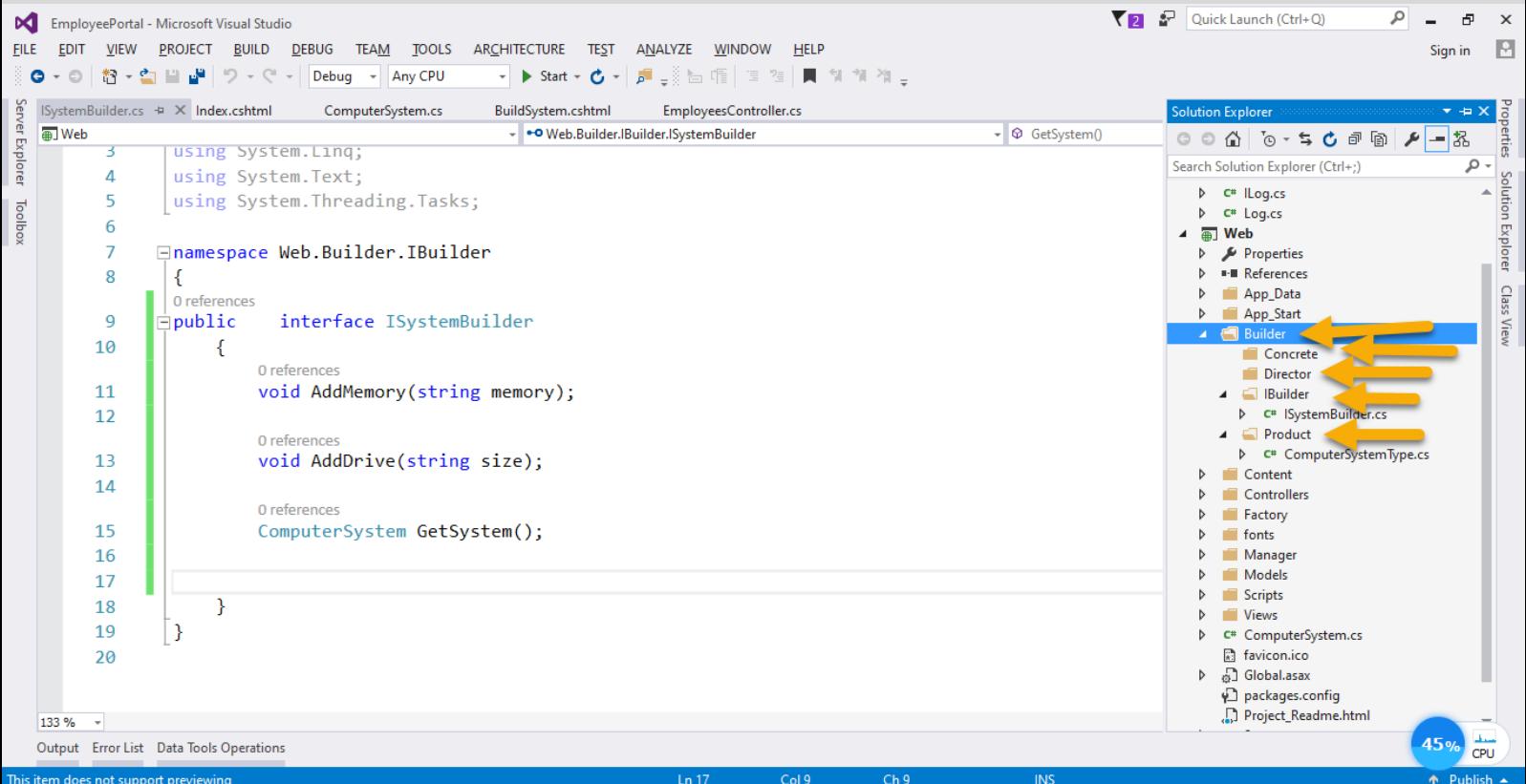
```

[HttpPost]
public ActionResult BuildDesktop(FormCollection formCollection)
{
    //Step 1
    Employee employee
        = db.Employees.Find(Convert.ToInt32(formCollection["employeeID"]));
    //Step 2 Concrete Builder
    ISystemBuilder systemBuilder = new DesktopBuilder();
    //Step 3 Director
    ConfigurationBuilder builder = new ConfigurationBuilder();
    builder.BuildSystem(systemBuilder, formCollection);
    //Step 4 return the system
    ComputerSystem system = systemBuilder.GetSystem();
    employee.SystemConfigurationDetails =
        string.Format("RAM : {0}, HDDSize : {1}, Keyboard: {2}, Mouse : {3}"
            , system.RAM, system.HDDSize, system.KeyBoard, system.Mouse);
    db.Entry(employee).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}

```

12

Slide 12 of 15



بنطبق زى المثال عملت الفولدرات ونقلت كلاس الكمبيوتر للمنتجات ونزلت عملت انترفيسيس كما موضح ونكمel

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
ISystemBuilder.cs Index.cshtml ComputerSystem.cs* BuildSystem.cshtml EmployeesController.cs
Sign in
Server Explorer Toolbox Properties Solution Explorer Class View
Web
4 references
9     public class ComputerSystem
10    {
11        private string _RAM;
12        private string _HDDSize;
13
14        1 reference
15        public ComputerSystem()
16        {
17
18            //public ComputerSystem(string RAM, string HDD)
19            //{
20                //    _RAM = RAM;
21                //    _HDDSize = HDD;
22            //}
23
24        1 reference
25        public string Build()
26        {
27            StringBuilder sb = new StringBuilder();
28
29            sb.Append(string.Format(" RAM: {0}", _RAM));
30
31            return sb.ToString();
32        }
33
34    }
35
36    1 reference
37    public void SetRAM(string RAM)
38    {
39        _RAM = RAM;
40    }
41
42    1 reference
43    public void SetHDDSize(string HDDSize)
44    {
45        _HDDSize = HDDSize;
46    }
47
48    1 reference
49    public string GetBuild()
50    {
51        return string.Format(" RAM: {0} | HDD: {1}", _RAM, _HDDSize);
52    }
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133 %
```

بعطل الكونستركتور

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

SystemBuilder.cs Index.cshtml ComputerSystem.cs* BuildSystem.cshtml EmployeesController.cs

Web ComputerSystem

```
public class ComputerSystem
{
    private string _RAM;
    private string _HDDSize;
```

1 reference

```
public ComputerSystem()
{}
```

1 reference

```
//public ComputerSystem(string RAM, string HDD)
//{
//    _RAM = RAM;
//    _HDDSize = HDD;
//}
```

1 reference

```
public string Build()
{
    StringBuilder sb = new StringBuilder();

    sb.Append(string.Format(" RAM: {0}", _RAM));
```

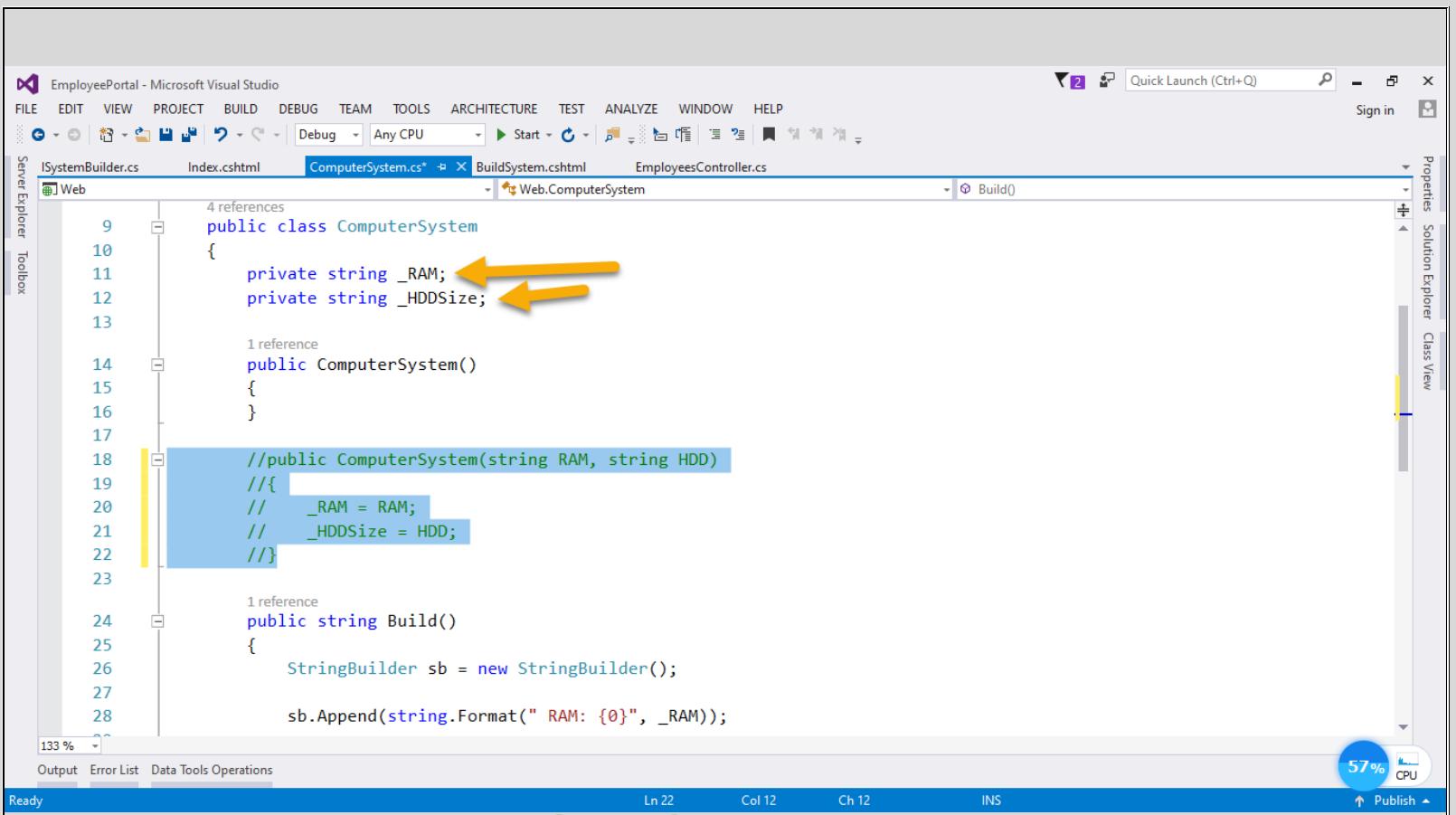
133 %

Output Error List Data Tools Operations

Ready

Ln 22 Col 12 Ch 12 INS

57% CPU



بغير في المتغيرات دی الاسم وبضيف لها الاتى

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

SystemBuilder.cs Index.cshtml ComputerSystem.cs* BuildSystem.cshtml EmployeesController.cs

Web ComputerSystem

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Web;
```

namespace Web

```
{
```

4 references

```
public class ComputerSystem
{
    1 reference
    private string RAM { get; set; }
    1 reference
    private string HDDSize { get; set; }
```

1 reference

```
public ComputerSystem()
{}
```

1 reference

```
//public ComputerSystem(string RAM, string HDD)
//{
//    _RAM = RAM;
//    _HDDSize = HDD;
```

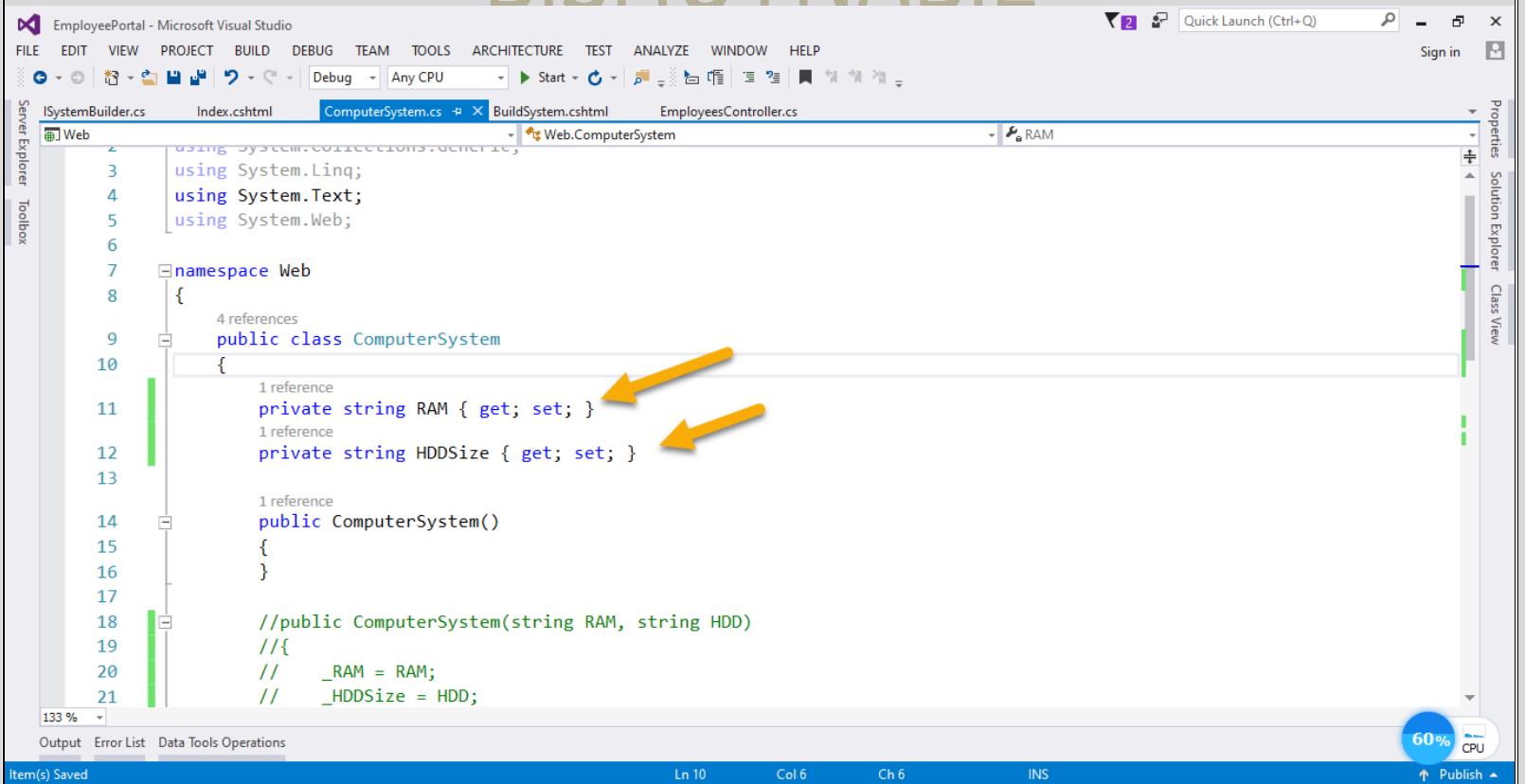
133 %

Output Error List Data Tools Operations

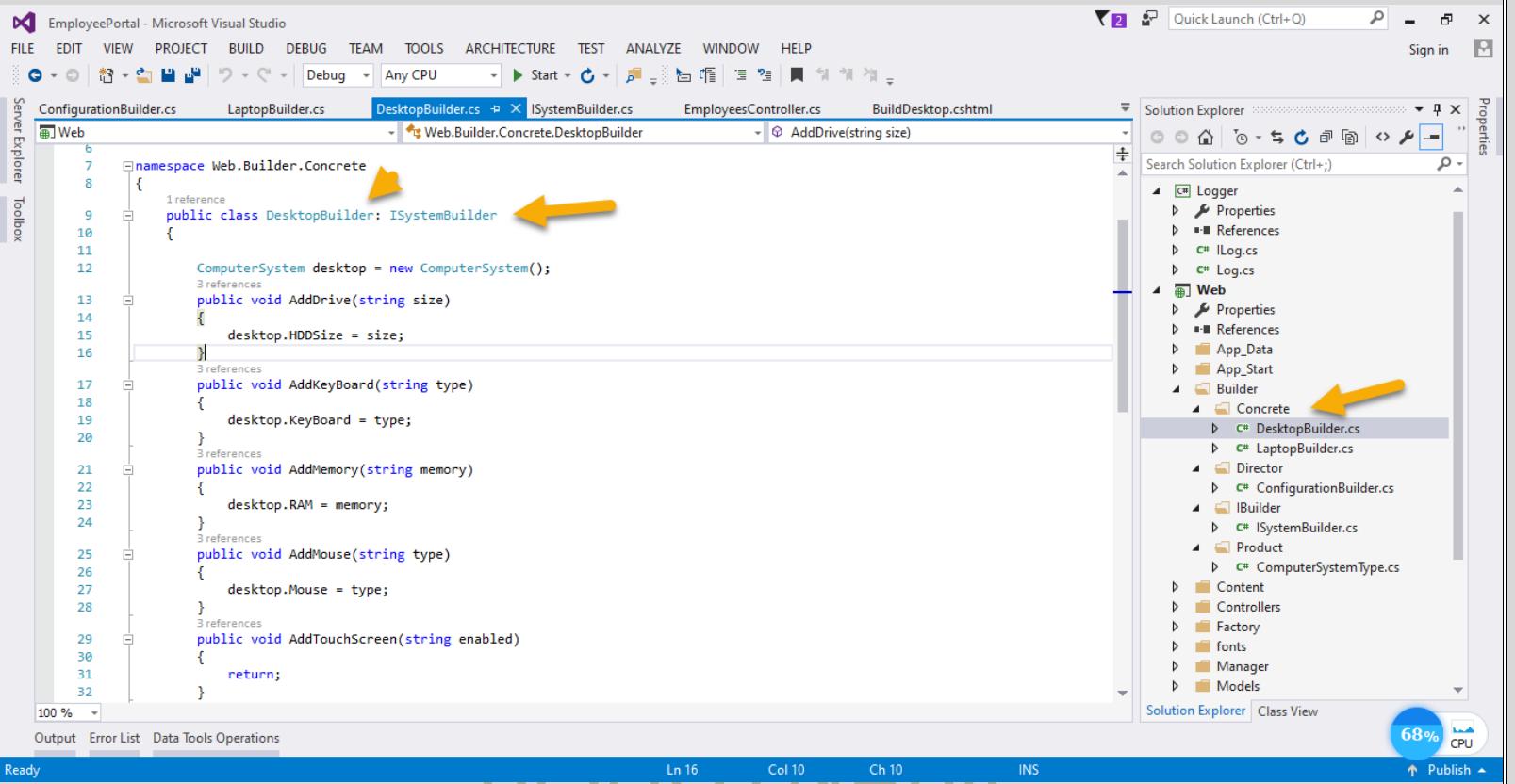
Item(s) Saved

Ln 10 Col 6 Ch 6 INS

60% CPU



نکمل



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Web.Builder.IBuilder;

namespace Web.Builder.Concrete
{
    public class DesktopBuilder: ISystemBuilder
    {

        ComputerSystem desktop = new ComputerSystem();
        public void AddDrive(string size)
        {
            desktop.HDDSize = size;
        }
        public void AddKeyBoard(string type)
        {
            desktop.KeyBoard = type;
        }
        public void AddMemory(string memory)
        {
            desktop.RAM = memory;
        }
        public void AddMouse(string type)
        {
            desktop.Mouse = type;
        }
        public void AddTouchScreen(string enabled)
        {
            return;
        }
    }
}
```

```
    }
    public void AddMouse(string type)
    {
        desktop.Mouse = type;
    }
    public void AddTouchScreen(string enabled)
    {
        return;
    }
    public ComputerSystem GetSystem()
    {
        return desktop;
    }
}
```

يُعمل كلاس للديسكتوب واحد فيه الانترفيس وابتدئ اتعامل

BISHOYNABIL

نَكْمَل

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Debug Any CPU Start
ConfigurationBuilder.cs LaptopBuilder.cs DesktopBuilder.cs ISystemBuilder.cs EmployeesController.cs BuildDesktop.cshtml
Web
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using Web.Builder.IBuilder;
6
7 namespace Web.Builder.Concrete
8 {
9     public class LaptopBuilder : ISystemBuilder
10    {
11        ComputerSystem laptop = new ComputerSystem();
12        public void AddDrive(string size)
13        {
14            laptop.HDDSize = size;
15        }
16
17        public void AddKeyboard(string type)
18        {
19            return;
20        }
21
22        public void AddMemory(string memory)
23        {
24            laptop.RAM = memory;
25        }
26
27        public void AddMouse(string type)
28    }
```

Ready

100 % Output Error List Data Tools Operations

Ln 17 Col 45 Ch 45 INS

73% CPU

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Web.Builder.IBuilder;

namespace Web.Builder.Concrete

{

public class LaptopBuilder : ISystemBuilder

 {

ComputerSystem laptop = new ComputerSystem();

public void AddDrive(string size)

 {

laptop.HDDSize = size;

 }

public void AddKeyboard(string type)

 {

return;

 }

public void AddMemory(string memory)

 {

laptop.RAM = memory;

 }

public void AddMouse(string type)

 }

```

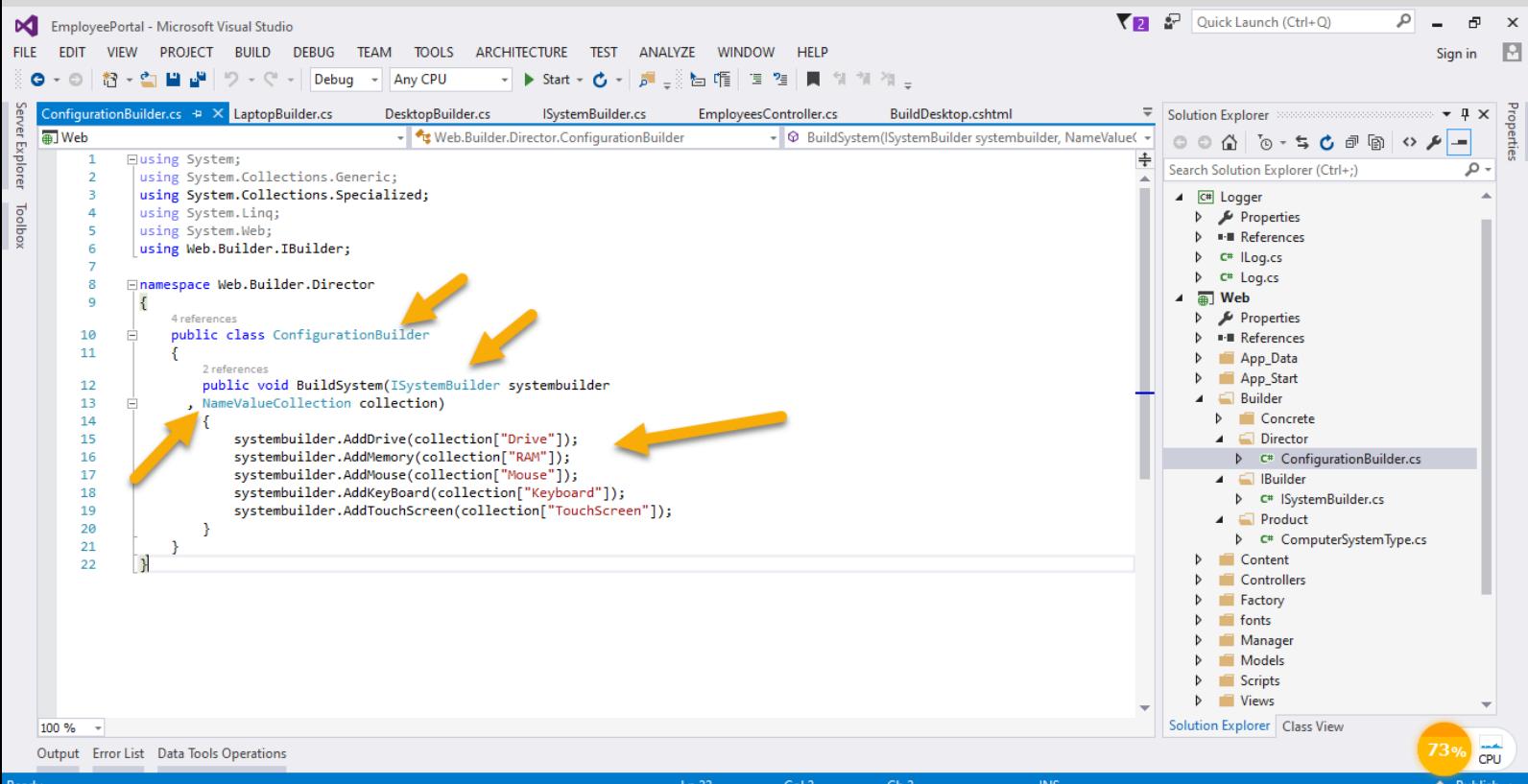
        return;
    }

    public void AddTouchScreen(string enabled)
    {
        laptop.TouchScreen = enabled;
    }

    public ComputerSystem GetSystem()
    {
        return laptop;
    }
}

```

**يُعمل كلاس لlap توب كما موضح
وبزود عليه خاصية التاتس سكرين**



يُعمل كلاس جديد واحد الكولكشن بتوعى

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

ConfigurationBuilder.cs LaptopBuilder.cs DesktopBuilder.cs ISystemBuilder.cs EmployeesController.cs BuildDesktop.cshtml

Web

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Web.Builder.IBuilder
8 {
9     public interface ISystemBuilder
10    {
11        void AddMemory(string memory);
12
13        void AddDrive(string size); ←
14
15        void AddKeyBoard(string type); ←
16
17        void AddMouse(string type); ←
18
19        void AddTouchScreen(string enabled);
20
21        ComputerSystem GetSystem();
22    }
23 }
```

100 % Output Error List Data Tools Operations

Ready Ln 16 Col 36 Ch 36 INS

Quick Launch (Ctrl+Q) Sign in Properties

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

- Logger
- Properties
- References
- C# ILog.cs
- C# Log.cs
- Web
- Properties
- References
- App_Data
- App_Start
- Builder
- Concrete
- Director
- C# ConfigurationBuilder.cs
- IBuilder
- C# ISystemBuilder.cs
- Product
- C# ComputerSystemType.cs
- Content
- Controllers
- Factory
- fonts
- Manager
- Models
- Scripts
- Views

Solution Explorer Class View 75% CPU

بضمف ميثنود جديدة داخل الانترفيسيس

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

ConfigurationBuilder.cs LaptopBuilder.cs DesktopBuilder.cs ISystemBuilder.cs EmployeesController.cs

Web

```
16 using Web.Models;
17
18 namespace Web.Controllers
19 {
20     public class EmployeesController : BaseController
21     {
22         private EmployeePortalEntities db = new EmployeePortalEntities();
23
24         [HttpGet]
25         //public ActionResult BuildSystem(int? employeeID)
26         //{
27
28             // return View(employeeID);
29         //}
30
31         [HttpGet]
32         public ActionResult BuildSystem(int? employeeID)
33         {
34             Employee employee = db.Employees.Find(employeeID);
35             if (employee.ComputerDetails.Contains("Laptop"))
36                 return View("BuildLaptop", employeeID);
37             else
38                 return View("BuildDesktop", employeeID);
39
40         }
41
42         //[[HttpPost]]
43         //public ActionResult BuildSystem(int employeeID, string RAM, string HDDSize)
44         //{
45             // Employee employee = db.Employees.Find(employeeID);
46             // ComputerSystem computerSystem = new ComputerSystem(RAM, HDDSize);
47             // employee.SystemConfigurationDetails = computerSystem.Build();
48             // db.Entry(employee).State = EntityState.Modified;
49
50         }
51 }
```

100 % Output Error List Data Tools Operations

Ready Ln 38 Col 10 Ch 10 INS

Quick Launch (Ctrl+Q) Sign in Properties

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

- Logger
- Properties
- References
- C# ILog.cs
- C# Log.cs
- Web
- Properties
- References
- App_Data
- App_Start
- Builder
- Content
- Controllers
- AccountController.cs
- BaseController.cs
- C# EmployeesController.cs
- HomeController.cs
- ManageController.cs
- Factory
- fonts
- Manager
- Models
- Scripts
- Views
- C# ComputerSystem.cs

Solution Explorer Class View 74% CPU

توقف الميثنود الحالية ونغيرها بميثنود جديدة وفق للمتغيرات اللي حصلت للمشروع

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
- Toolbars:** Standard, Debug, Start, Build, Analyze, Window
- Solution Explorer:** Shows Solution 'EmployeePortal' (2 projects) with files like Logger, Properties, References, ILog.cs, Log.cs, Web, and Controllers.
- Properties View:** Shows settings for selected files.
- Task List:** Search Solution Explorer (Ctrl+Shift+F)
- Code Editor:** Displays the EmployeesController.cs file with the following code:

```
38 //[[HttpPost]]  
39 //public ActionResult BuildSystem(int employeeID, string RAM, string HDDSize)  
40 //{{  
41 //    Employee employee = db.Employees.Find(employeeID);  
42 //    ComputerSystem computerSystem = new ComputerSystem(RAM, HDDSize);  
43 //    employee.SystemConfigurationDetails = computerSystem.Build();  
44 //    db.Entry(employee).State = EntityState.Modified;  
45 //    db.SaveChanges();  
46 //    return RedirectToAction("Index");  
47 //}}  
48 [HttpPost]  
49 References  
50 public ActionResult BuildLaptop(FormCollection formCollection)  
51 {  
52     Employee employee =  
53         db.Employees.Find(Convert.ToInt32(formCollection["employeeID"]));  
54     //Concrete Builder  
55     ISystemBuilder systemBuilder = new LaptopBuilder();  
56     //Director  
57     ConfigurationBuilder builder = new ConfigurationBuilder();  
58     builder.BuildSystem(systemBuilder, formCollection);  
59     ComputerSystem system = systemBuilder.GetSystem();  
60  
61     employee.SystemConfigurationDetails =  
62     string.Format("RAM : {0}, HDDSize : {1}, TouchScreen: {2}"  
63     , system.RAM, system.HDDSize, system.TouchScreen);  
64  
65     db.Entry(employee).State = EntityState.Modified;  
66     db.SaveChanges();  
67     return RedirectToAction("Index");  
68 }
```

بوقف ميثود البوست

واستبدلها بالميثود الجديدة

اللى هيا لابتوب ونكمel

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Solution Explorer:** Shows the solution structure for 'EmployeePortal' with two projects: 'Logger' and 'Web'. The 'Web' project contains files like Properties, References, App_Data, App_Start, Builder, Content, Controllers (AccountController.cs, BaseController.cs, EmployeesController.cs, HomeController.cs, ManageController.cs), Factory, fonts, Manager, Models, Scripts, Views, and ComputerSystem.cs.
- Properties Window:** Located on the right side of the interface.
- Code Editor:** Displays the 'EmployeesController.cs' file. The code implements the建造者模式 (Builder pattern) to build computer systems based on employee requirements. It includes methods for building desktops and listing employees.
- Status Bar:** Shows 'Ln 38 Col 10 Ch 10 INS'.
- Bottom Navigation:** Includes tabs for Output, Error List, Data Tools Operations, and a Publish button.

بتصيفكمان مينهود البوست للديسك توب
ونكمل

هندروج نعمل

2 انترفيو باسم اللاب توب والديسكتوب

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

BuildLaptop.cshtml ComputerSystemType.cs ConfigurationBuilder.cs LaptopBuilder.cs DesktopBuilder.cs ISystemBuilder.cs

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

EmployeesController.cs
HomeController.cs
ManageController.cs
Factory
fonts
Manager
Models
Scripts
Views
Account
Employees
BuildDesktop.cshtml
BuildLaptop.cshtml
BuildSystem.cshtml
Create.cshtml
Delete.cshtml
Edit.cshtml
Index.cshtml
Home
Manage
Shared
_ViewStart.cshtml
Web.config
ComputerSystem.cs

Properties

Server Explorer Toolbox

Output Error List... Data Tools Operations

Ready

100 %

Ln 74 Col 2 Ch 2 INS

65% CPU

```
@model Int32
 @{
    ViewBag.Title = "BuildSystem";
}
<h2>Build System</h2>
<style>
    input[type=radio] {
        border: 0px;
        width: 100%;
        height: 1em;
    }
</style>
@using (Html.BeginForm("BuildLaptop", "Employees", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>System Configuration</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.Hidden("employeeID", this.Model.ToString())

        <div class="form-group">
            @Html.Label("Memory", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("RAM",
                    new List<SelectListItem>() {
                        new SelectListItem(){ Text = "Select", Value="Select"},
                        new SelectListItem(){ Text = "8GB", Value="8GB"},
                        new SelectListItem(){ Text = "16GB", Value="16GB"},
                        new SelectListItem(){ Text = "32GB", Value="32GB"}},
                    htmlAttributes: new { @class = "form-control" })
            </div>
        </div>
    </div>
}
```

BISHOYNABIL

```
@model Int32
 @{
    ViewBag.Title = "BuildSystem";
}
<h2>Build System</h2>
<style>
    input[type=radio] {
        border: 0px;
        width: 100%;
        height: 1em;
    }
</style>
@using (Html.BeginForm("BuildLaptop", "Employees", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>System Configuration</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.Hidden("employeeID", this.Model.ToString())

        <div class="form-group">
            @Html.Label("Memory", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("RAM",
                    new List<SelectListItem>() {
                        new SelectListItem(){ Text = "Select", Value="Select"},
```

```

        new SelectListItem(){ Text = "8GB", Value="8GB"},  

        new SelectListItem(){ Text = "16GB", Value="16GB"},  

        new SelectListItem(){ Text = "32GB", Value="32GB"},  

    }, htmlAttributes: new { @class = "form-control" })  
  

    </div>  

</div>  

<div class="form-group">  

    @Html.Label("Drive", htmlAttributes: new { @class = "control-label col-md-2" })  

<div class="col-md-10">  

    @Html.DropDownList("Drive",  

    new List<SelectListItem>() {  

        new SelectListItem(){ Text = "Select", Value="Select"},  

        new SelectListItem(){ Text = "500GB", Value="500GB"},  

        new SelectListItem(){ Text = "1TB", Value="1TB"},  

    }, htmlAttributes: new { @class = "form-control" })  
  

    </div>  

</div>  

<div class="form-group">  

    @Html.Label("Touch Enabled", htmlAttributes: new { @class = "control-label col-md-2" })  

<div class="col-md-10">  

    <div class="col-md-1">  

        <label>@Html.RadioButton("TouchScreen", "Yes", false, htmlAttributes: new { @class = "radio-inline" }) YES</label>  

        BISHOYNABIL  

        </div>  

<div class="col-md-1">  

        <label>@Html.RadioButton("TouchScreen", "NO", false, htmlAttributes: new { @class = "radio-inline" }) NO</label>  

        </div>  

</div>  

<div class="form-group">  

    <div class="col-md-offset-2 col-md-10">  

        <input type="submit" value="Create" class="btn btn-default" />  

    </div>  

</div>  

</div>  

}>  

<div>  

    @Html.ActionLink("Back to List", "Index")
</div>  
  

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

كود صفحة الفيو ونكمـل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Server Explorer Toolbox Solution Explorer Properties

BuildLaptop.cshtml ComputerSystemType.cs ConfigurationBuilder.cs LaptopBuilder.cs ISystemBuilder.cs BuildDesktop.cshtml

```
1 @model Int32
2 @{
3     ViewBag.Title = "BuildSystem";
4 }
5 <h2>Build System</h2>
6 <style>
7     input[type=radio] {
8         border: 0px;
9         width: 100%;
10        height: 1em;
11    }
12 </style>
13 @using (Html.BeginForm("BuildDesktop", "Employees", FormMethod.Post))
14 {
15     @Html.AntiForgeryToken()
16
17     <div class="form-horizontal">
18         <h4>System Configuration</h4>
19         <hr />
20         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
21         @Html.Hidden("employeeID", this.Model.ToString());
22
23         <div class="form-group">
24             @Html.Label("Memory", htmlAttributes: new { @class = "control-label col-md-2" })
25             <div class="col-md-10">
26                 @Html.DropDownList("RAM",
27                     new List<SelectListItem>() {
28                         new SelectListItem (){ Text = "Select", Value="Select"},
29                         new SelectListItem (){ Text = "8GB", Value="8GB"},
30                         new SelectListItem (){ Text = "16GB", Value="16GB"}, 
31                         new SelectListItem (){ Text = "32GB", Value="32GB"}, 
32                     }, htmlAttributes: new { @class = "form-control" })
33     </div>
34 }
35 
```

Ready Output Error List Data Tools Operations

Ln 90 Col 2 Ch 2 INS

67% CPU

```
@model Int32
{@
    ViewBag.Title = "BuildSystem";
}
<h2>Build System</h2>
<style>
    input[type=radio] {
        border: 0px;
        width: 100%;
        height: 1em;
    }
</style>
@using (Html.BeginForm("BuildDesktop", "Employees", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>System Configuration</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.Hidden("employeeID", this.Model.ToString());

        <div class="form-group">
            @Html.Label("Memory", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("RAM",
```

```
new List<SelectListItem>() {
    new SelectListItem (){ Text = "Select", Value="Select"},
    new SelectListItem (){ Text = "8GB", Value="8GB"},
    new SelectListItem (){ Text = "16GB", Value="16GB"},
    new SelectListItem (){ Text = "32GB", Value="32GB"},
}, htmlAttributes: new { @class = "form-control" })
```

```
</div>
</div>
<div class="form-group">
    @Html.Label("Drive", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Drive",
    new List<SelectListItem>() {
        new SelectListItem (){ Text = "Select", Value="Select"},
        new SelectListItem (){ Text = "500GB", Value="500GB"},
        new SelectListItem (){ Text = "1TB", Value="1TB"},
}, htmlAttributes: new { @class = "form-control" })
```

```
</div>
</div>
<div class="form-group">
    @Html.Label("Mouse", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("Mouse",
    new List<SelectListItem>() {
        new SelectListItem (){ Text = "Select", Value="Select"},
        new SelectListItem (){ Text = "WireLess", Value="USB-WireLess"},
        new SelectListItem (){ Text = "Regular-USB", Value="USB"},
}, htmlAttributes: new { @class = "form-control" })
```

```
</div>
</div>
<div class="form-group">
    @Html.Label("KeyBoard", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("KeyBoard",
    new List<SelectListItem>() {
        new SelectListItem (){ Text = "Select", Value="Select"},
        new SelectListItem (){ Text = "Wireless", Value="Wireless"},
        new SelectListItem (){ Text = "Regular-USB", Value="Regular-USB"},
}, htmlAttributes: new { @class = "form-control" })
```

```
</div>
</div>
```

```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Create" class="btn btn-default" />
    </div>
</div>
</div>
```

```

}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

کود الغیو بتاع الديسلک توب ونكملي

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Index.cshtml ComputerSystem.cs BuildSystem.cshtml

Server Explorer Toolbox

```

13     public string Keyboard { get; set; }
14     public string Mouse { get; set; }
15     public string TouchScreen { get; set; }
16
17     public ComputerSystem()
18     {
19     }
20
21     //public ComputerSystem(string RAM, string HDD)
22     //{
23     //    _RAM = RAM;
24     //    _HDDSize = HDD;
25     //}
26
27     //public string Build()
28     //{
29     //    StringBuilder sb = new StringBuilder();
30
31     //    sb.Append(string.Format(" RAM: {0}", RAM));
32
33     //    sb.Append(string.Format(" HDDSize: {0}", HDDSize));
34
35     //    return sb.ToString();
36     //}
37 }
38
39

```

Solution Explorer

Logger Properties References ILog.cs Log.cs Web Properties References App_Data App_Start Builder Concrete DesktopBuilder.cs LaptopBuilder.cs Director ConfigurationBuilder.cs IBuilder SystemBuilder.cs Product Content Controllers Factory fonts Manager Models Scripts

Output Error List Data Tools Operations

Ready

Ln 21 Col 6 Ch 6 INS

67% CPU

وبوقف الميثود دی لاني مش محتاجها

PowerPoint Slide Show - [Part 13 - Fluent Builder Design Pattern [Compatibility Mode]] - PowerPoint

Fluent Builder Design Pattern

Fluent Interface

- The idea behind a Fluent interface is that one can apply multiple properties to an object by connecting them with dots and without having to re-specify the object each time
- C# uses fluent programming extensively in LINQ to build queries using the standard query operators. The implementation is based on extension methods

```
var filtered = translations.Where (t => t.Key.Contains("a"));
var sorted = filtered.OrderBy (t => t.Value.Length);
var finalQuery = sorted.Select (t => t.Value.ToUpper());
```



```
IEnumerable<string> query = translations
    .Where (t => t.Key.Contains("a"))
    .OrderBy (t => t.Value.Length)
    .Select (t => t.Value.ToUpper());
```

5

Fluent pattern

هو انترفيس بنتيج لـك انه تنادى اكتر من ميثود او بروبرتس على نفس الاوبجيكت بدون
ماترجع تحده الاوبجيكت نفسه فى كل مرة

وكمان اقدر انادى على ميثود ومن خاللها اعمل اكسيس على باقى الميثود الثانية

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Server Explorer Toolbox

IWeb ISystemBuilder.cs Index.cshtml ComputerSystem.cs BuildSystem.cshtml

Web.WebBuilder.IBuilder

```
7  namespace Web.Builder.IBuilder
8  {
9      public interface ISystemBuilder
10     {
11         void AddMemory(string memory);
12
13         void AddDrive(string size);
14
15         void AddKeyBoard(string type);
16         void AddMouse(string type);
17
18         void AddTouchScreen(string enabled);
19         ComputerSystem GetSystem();
20     }
21 }
22 }
```

133% Output Error List Data Tools Operations

Ready

LN 1 Col 1 Ch 1 INS

68% CPU

Solution Explorer Class View

Search Solution Explorer (Ctrl+F)

- App_Data
- App_Start
- Builder
 - Concrete
 - DesktopBuilder.cs
 - LaptopBuilder.cs
 - Director
 - ConfigurationBuilder.cs
 - IBuilder
 - ISystemBuilder.cs
 - Product
- Content
- Controllers
- Factory
- fonts
- Manager
- Models
- Scripts
- Views
- ComputerSystem.cs
- favicon.ico
- Global.asax
- packages.config
- Project_Readme.html
- Startup.cs
- Web.config

LN 1 Col 1 Ch 1 INS

68% CPU

هنا انا عايز اخلى الميثود دى كلهم يرجعوا بنا دو نفسهم ازاي هغير التایب بتاعهم

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Server Explorer Toolbox

IWeb ISystemBuilder.cs Index.cshtml ComputerSystem.cs BuildSystem.cshtml

Web.WebBuilder.IBuilder

```
7  namespace Web.Builder.IBuilder
8  {
9      public interface ISystemBuilder
10     {
11         void AddMemory(string memory);
12
13         void AddDrive(string size);
14
15         void AddKeyBoard(string type);
16         void AddMouse(string type);
17
18         void AddTouchScreen(string enabled);
19
20         ComputerSystem GetSystem();
21     }
22 }
```

133% Output Error List Data Tools Operations

Ready

LN 16 Col 33 Ch 33 INS

77% CPU

Solution Explorer Class View

Search Solution Explorer (Ctrl+F)

- App_Data
- App_Start
- Builder
 - Concrete
 - DesktopBuilder.cs
 - LaptopBuilder.cs
 - Director
 - ConfigurationBuilder.cs
 - IBuilder
 - ISystemBuilder.cs
 - Product
- Content
- Controllers
- Factory
- fonts
- Manager
- Models
- Scripts
- Views
- ComputerSystem.cs
- favicon.ico
- Global.asax
- packages.config
- Project_Readme.html
- Startup.cs
- Web.config

LN 16 Col 33 Ch 33 INS

77% CPU

نكمـل

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

LaptopBuilder.cs DesktopBuilder.cs

Web

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using Web.Builder.IBuilder;
6
7 namespace Web.Builder.Concrete
8 {
9     public class LaptopBuilder : ISystemBuilder
10    {
11        ComputerSystem laptop = new ComputerSystem();
12
13        public void AddDrive(string size)
14        {
15            laptop.HDDSize = size;
16        }
17
18        public void AddKeyBoard(string type)
19        {
20            return;
21        }
22    }
}
```

133 %

Output Error List Data Tools Operations

Ready Ln 12 Col 16 Ch 16 INS

Solution Explorer

Search Solution Explorer (Ctrl+J)

- App_Data
- App_Start
- Builder
 - Concrete
 - DesktopBuilder.cs
 - LaptopBuilder.cs
 - Director
 - ConfigurationBuilder.cs
 - IBuilder
 - ISystemBuilder.cs
 - Product
- Content
- Controllers
- Factory
- fonts
- Manager
- Models
- Scripts
- Views
- ComputerSystem.cs
- favicon.ico
- Global.asax
- packages.config
- Project_Readme.html
- Startup.cs
- Web.config

Solution Explorer Class View

60% CPU

هنجير كل الغويد هنا اللى
BISHOYNABIL
ISystemBuilder

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

BundleConfig.cs LaptopBuilder.cs DesktopBuilder.cs

Web

```
1 using Web.Builder.IBuilder;
2
3 namespace Web.Builder.Concrete
4 {
5     public class DesktopBuilder: ISystemBuilder
6     {
7
8         ComputerSystem desktop = new ComputerSystem();
9
10        public ISystemBuilder AddDrive(string size)
11        {
12            desktop.HDDSize = size;
13            return this;
14        }
15
16        public ISystemBuilder AddKeyBoard(string type)
17        {
18
19            desktop.KeyBoard = type;
20            return this;
21        }
22
23        public ISystemBuilder AddMemory(string memory)
24        {
25            desktop.RAM = memory;
26        }
27    }
}
```

121 %

Output Error List Data Tools Operations

Item(s) Saved Ln 41 Col 10 Ch 10 INS

Solution Explorer

Search Solution Explorer (Ctrl+J)

- App_Data
- App_Start
- Builder
 - Concrete
 - DesktopBuilder.cs
 - LaptopBuilder.cs
 - Director
 - ConfigurationBuilder.cs
 - IBuilder
 - ISystemBuilder.cs
 - Product
- Content
- Controllers
- Factory
- fonts
- Manager
- Models
- Scripts
- Views
- ComputerSystem.cs
- favicon.ico
- Global.asax
- packages.config
- Project_Readme.html
- Startup.cs
- Web.config

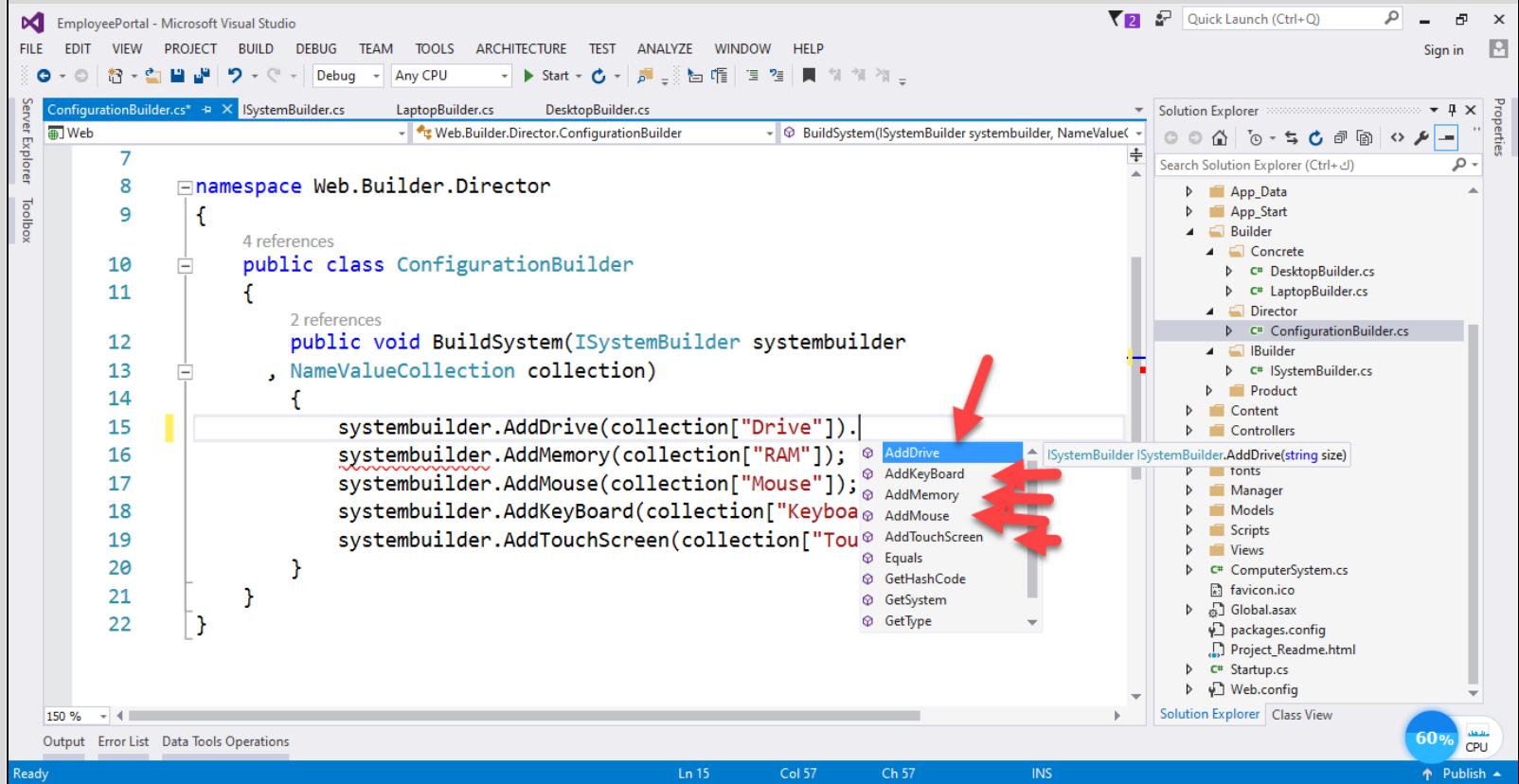
Solution Explorer Class View

64% CPU

ونعمل نفس الكلام في كلاس الـ

Laptop

ولو شغلت مفيش اختلاف في الكود لكن كود البرنامج اختلف



The screenshot shows the Microsoft Visual Studio interface with the following details:

- File Menu:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
- Toolbars:** Standard, Debug, Start, Stop, Run, Task List, Solution Explorer, Properties, Task List, Status Bar.
- Solution Explorer:** Shows the project structure with nodes like App_Data, App_Start, Builder, Concrete, Director, ConfigurationBuilder.cs, IBuilder, ISystemBuilder.cs, Product, Content, Controllers, tons, Manager, Models, Scripts, Views, ComputerSystem.cs, favicon.ico, Global.asax, packages.config, Project_Readme.html, Startup.cs, and Web.config.
- Code Editor:** Displays the `ConfigurationBuilder.cs` file with the following code:

```
7
8     namespace Web.Builder.Director
9     {
10        public class ConfigurationBuilder
11        {
12            public void BuildSystem(ISystemBuilder systembuilder
13            , NameValueCollection collection)
14            {
15                systembuilder.AddDrive(collection["Drive"]);
16                systembuilder.AddMemory(collection["RAM"]);
17                systembuilder.AddMouse(collection["Mouse"]);
18                systembuilder.AddKeyBoard(collection["Keyboard"]);
19                systembuilder.AddTouchScreen(collection["TouchScreen"]);
20            }
21        }
22    }
```
- IntelliSense:** A tooltip is displayed over the `systembuilder.AddDrive` call, listing methods: AddDrive, AddKeyBoard, AddMemory, AddMouse, Equals, GetHashCode, GetSystem, GetType. Three red arrows point from the text above to this tooltip.
- Status Bar:** Shows 150% zoom, Ln 15, Col 57, Ch 57, INS.
- Output Tab:** Ready.
- Properties Tab:** Quick Launch (Ctrl+Q).

بعض اللي كنا بنتكلم عنه اهو من اول ميثود قدت اشوف باقى الميثود وادخل لها واتعامل معها تعالى نستفيد من التعديل ده

EmployeePortal - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Server Explorer Toolbox

ConfigurationBuilder.cs ISystemBuilder.cs LaptopBuilder.cs DesktopBuilder.cs Web

7
8 namespace Web.Builder.Director
9 {
10 4 references
11 public class ConfigurationBuilder
12 {
13 2 references
14 public void BuildSystem(ISystemBuilder systembuilder,
15 NameValueCollection collection)
16 {
17 systembuilder.AddDrive(collection["Drive"])
18 .AddMemory(collection["RAM"])
19 .AddMouse(collection["Mouse"])
20 .AddKeyBoard(collection["Keyboard"])
21 .AddTouchScreen(collection["TouchScreen"]);
22 }
23 }
24 }
150 % Output Error List Data Tools Operations

Solution Explorer

Search Solution Explorer (Ctrl+J)

- App_Data
- App_Start
- Builder
 - Concrete
 - DesktopBuilder.cs
 - LaptopBuilder.cs
 - Director
 - ConfigurationBuilder.cs
 - IBuilder
 - ISystemBuilder.cs
 - Product
- Content
- Controllers
- Factory
- fonts
- Manager
- Models
- Scripts
- Views
- ComputerSystem.cs
- favicon.ico
- Global.asax
- packages.config
- Project_Readme.html
- Startup.cs
- Web.config

Solution Explorer Class View 67% CPU

Item(s) Saved

Ln 21 Col 6 Ch 6 INS

BISHOYNABIL تمام نجرب

```
7  
8    namespace Web.Builder.Director  
9    {  
10      4 references  
11      public class ConfigurationBuilder  
12      {  
13        2 references  
14        public void BuildSystem(ISystemBuilder systembuilder,  
15            NameValueCollection collection)  
16        {  
17            systembuilder.AddDrive(collection["Drive"])  
18            .AddMemory(collection["RAM"])  
19            .AddMouse(collection["Mouse"])  
20            .AddKeyBoard(collection["Keyboard"])  
21            .AddTouchScreen(collection["TouchScreen"]);  
22        }  
23      }  
24 }
```

http://localhost:49936/employees/Create

Create - My ASP.NET Appl... ×

File Edit View Favorites Tools Help

Home Page - My ASP.NET...

Application name Home About Contact Register Log in

Create

Employee

Name	Gr
JobDescription	Manager
Number	c015
Department	Itl
EmployeeTypeID	Permanent

[Create](#)

[Back to List](#)

© 2019 - My ASP.NET Application

http://localhost:49936/employees

Index - My ASP.NET Appl... ×

File Edit View Favorites Tools Help

Home Page - My ASP.NET...

Application name Home About Contact Register Log in

sara	Manager	C0	Manager	12.00	5.00	Contract	100.00	Dell,I5,Laptop	Buildsystem	Edit Details Delete
John	hr	H1009	Hr	12.00	5.00	Contract	100.00	Dell,I5/Desktop	RAM : 8GB, HDDSize : 500GB, Keyboard: Wireless, Mouse : USB-WireLess	Edit Details Delete
Mano	Manager	R587	It	8.00	10.00	Permanent	150.00	Apple,I7,Laptop	RAM : 8GB, HDDSize : 500GB, TouchScreen: Yes	Edit Details Delete
Gr	Manager	c015	Itl	8.00	10.00	Permanent	150.00	Apple,I7,Laptop	Buildsystem	Edit Details Delete

© 2019 - My ASP.NET Application

http://localhost:49936/employees/BuildSystem?employeeID=18

BuildSystem - My ASP.NET ... ×

File Edit View Favorites Tools Help

Home Page - My ASP.NET...

Application name Home About Contact Register Log in

Build System

System Configuration

Memory: 32GB

Drive: 1TB

Touch Enabled: YES

[Create](#)

[Back to List](#)

© 2019 - My ASP.NET Application

http://localhost:49936/Employees/BuildLaptop

Index - My ASP.NET Applic... ×

File Edit View Favorites Tools Help

Home Page - My ASP.NET...

Application name Home About Contact Register Log in

Employee Name	Role	Employee ID	Department	Hours Worked	Hourly Rate	Overtime Rate	Contract Type	Salary	Computer Model	RAM	HDD Size	Keyboard	Mouse	Wireless	Buildsystem	Action
John	hr	H1009	Hr	12.00	5.00	Contract	100.00	Dell,I5/Desktop	RAM : 8GB, HDD Size : 500GB, Keyboard: Wireless, Mouse : USB-WireLess	Edit Details						
Mano	Manager	R587	It	8.00	10.00	Permanent	150.00	Apple,i7,Laptop	RAM : 8GB, HDD Size : 500GB, TouchScreen: Yes	Edit Details						
Gr	Manager	c015	Itl	8.00	10.00	Permanent	150.00	Apple,i7,Laptop	RAM : 32GB, HDD Size : 1TB, TouchScreen: Yes	Edit Details						

© 2019 - My ASP.NET Application

بدون مشاكل

Fluent Builder Design Pattern

Fluent Interface

- **Fluent interface is a method for constructing object oriented APIs, where the readability of the source code is close to that of ordinary written prose**
- **Fluent interface is normally implemented by using method cascading (concretely method chaining)**
- **Fluent code is much more readable and allows to vary a product's internal representation**
- **Fluent Encapsulates code for construction and representation and Provides control over steps of an object construction process**
- **Searching, Sorting, pagination, grouping with a blend of LINQ are some of the real world usage of fluent interface in combination with builder design pattern.**

6

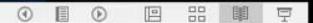
BISHOYNABIL

Builder Vs Factory and Abstract Factory

- **Builder design pattern encapsulates complex creation into a single method .**
- **Builder design pattern focuses on construction of object in a step by step manner whereas Abstract factory pattern is used to solve problems related to the creation of families of products**
- **Builder design pattern returns the object after step by step construction of the complex object where as in Abstract factory or Factory pattern, we return the created product immediately**
- **To conclude, Many application designs start out using Factory which is less complex and evolve towards Abstract Factory and Builder as the complexity increases with a demand of flexibility**

7

Slide 7 of 12



BISHOY NABIL
مهمة الصورة دى

Builder Vs Factory and Abstract Factory

Builder design pattern encapsulates complex creation into a single method.

Builder design pattern focuses on construction of object in a step by step manner whereas Abstract factory pattern is used to solve problems related to the creation of families of products.

Builder design pattern returns the object after step by step construction of the complex object where as in Abstract factory or Factory pattern, we return the created product immediately.

To conclude, many application designs start out using Factory which is less complex and evolve towards Abstract Factory and Builder as the complexity increases with a demand of flexibility.

وبيقولك كمان الديفلوبير بيروحو دايما للفاكتوري باترن لانه اقل تعقيدا لكن اللي بعده لو بروجيكت بيزنس وبحصل فيه تشعب بامكانك استخدام البيلدر باترن و الايستركت فاكتوري

ولكن كل بيزنس له الباترن الخاص بها وليس باترن معين يمكن استخدامها في الجميع على حسب البيزنس

Examples of Builder Design Pattern

- Building a Pizza
- Building a Vehicle in the Manufacturing Factory
- Building a Lunch/Dinner Menu (Online ordering)
- Building a House (Step by step construction of a house)

امثله نقدر نستخدم فيها البيلدر

In this session we will learn

- What is prototype design pattern
- Implementation guidelines
- Shallow and deep copy

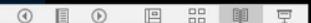
Suggested Videos :

Part 1 to 13 Design Patterns Tutorial

Link to Dot Net Basics, C#, LINQ,SQL Server, ASP.NET, ADO.NET and EF video series
<https://www.youtube.com/c/kudvenkatarabic/playlists>

2

Slide 2 of 15



Prototype Design Pattern

Gang Of Four Definition

“Prototype Design Pattern Specify the kind of objects to create using a prototypical instance, and create new objects by copying this prototype”

To Simplify, Instead of creating object from scratch every time, you can make copies of an original instance and modify it as required.

Prototype is unique among the other creational patterns as it doesn't require a class but only an end object.

3

Slide 3 of 15



حسب تعريف حاج اوف فور ده موقع برمجي دور عليه على النت نكمـل

Proto type design pattern

هو بيحدد نوع الاوبيجيكس اللي هيتم انشائها بواسطه او مايعرف بـ

Prototypical instance

بيعملك اوبيجيكت موجود قبل كده بيأخذ منه كوبى وليس بينشئ اوبيجيكت من الاسكراتش ترجمتها من الصفر | يعني مش بي عملك اوبيجيكت جديد من الصفر نكمel

بمعنى نسهل التعريف

To simplify, instead of creating object from scratch every time, you can make copies of an original instance and modify it as required

بيقولك بدل ما كل مرة تعمل اوبيجيكت

Scratch

من الصفر يعني لا تقدر تعمل اوبيجيكت منسوخ من الاوبيجيكت الاصلى (بمعنى عندك فعليا اوبيجيكت معمول هناخد منه نسخه ونعدل عليها زى ما حنا عايزين) وتعديل فيه على حسب احتجاجك

Prototype is unique among the other creational patterns as it doesn't require a class but only an end object.

بيقولك كمان البروتو تايب يونييك وده اللي بيتميزه عن الباترن الآخرى وهو مش بيحتاج كلاس هو بيحتاج فقط اوبيجيكت

Implementation Guidelines

Choose Prototype Design Pattern when

- Creating an object is an expensive operation and it would be more efficient to copy an object.
- System should be independent of how its products are created, composed, and represented.
- Objects are required that are similar to existing objects.
- We need to hide the complexity of creating new instance from the client.

4

Slide 4 of 15

BISHOY NABIL
انا متأملاً بحاجة استخدمه

يقولك ام يكون تكلفة انشاء الاوبيجيكت مكلفة جداً بالمقارنة مع عملية نسخه وه تكون اكتر كفاءة

ثانياً

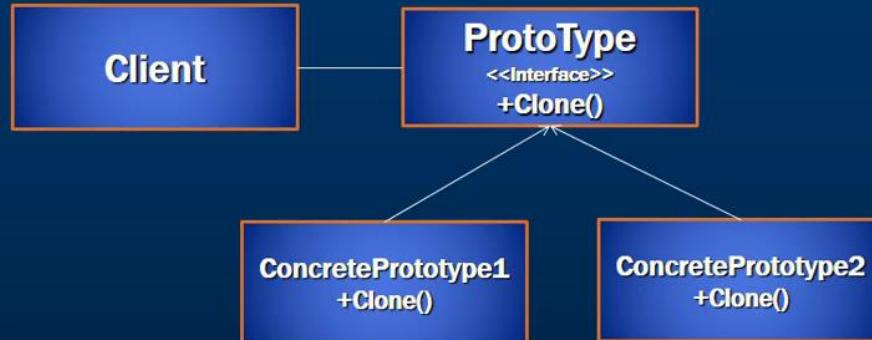
لو النظم مستقلأً مش فاهمنا الللى بيسرح مش بيترجم صح

ثالثاً

رابعاً لما اكون عايز اخفى عملية انشاء الاوبيجيكت عن اليوزر واستخدم البروتو تايب عشان اخفى
العملية كلها

Prototype Representation

Prototype Design Pattern



Client : Creates new object by asking prototype to clone itself.

Prototype : Specifies an abstract interface for cloning the required object.

Concrete Prototype : Implements the prototype interface for cloning itself.

5

BISHOYNABIL

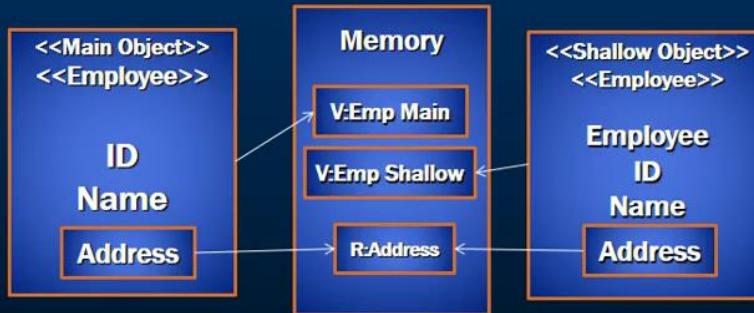
Shallow and Deep Copy

The idea of using copy is to create a new object of the same type without knowing the exact type of the object we are invoking.

Shallow and deep copying are two mechanism used in copy an object.

Shallow Copy

Shallow Copy copies an object's value type fields into the target object and the object's reference types are copied as references into the target object.



6

فكرة استخدام الكوبى من اجل انشاء اوبيجكت من نفس النوع بدون مانرجع مرة تانية نقوم بإنشاء هذا الاوبيجكت وبنفس الفكرة بدون ما اعرف نوع الاوبيجكت اللى يستدعيه

عندى 2

Mechanism

لاستخدامهم لعمل كوبى من اوبيجكت

هما ولا

ال

Shallow

And

Deep coping

ونستخدمهم لانشاء اوبيجكت من نفس النوع بدون مانكون عارفين نوع الاوبيجكت

اولا

Shallow copy

كما فى الصورة

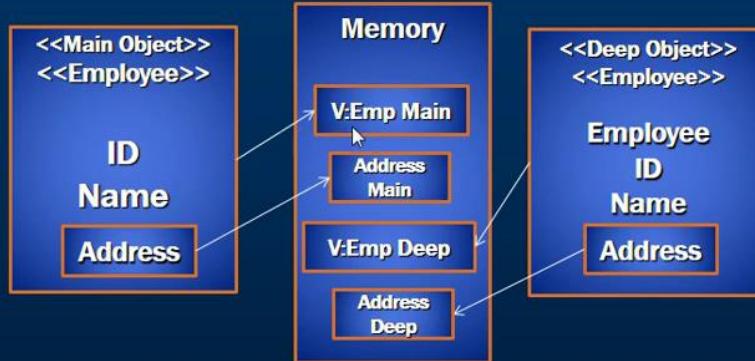
D:

Deep copy

Shallow and Deep Copy

Deep Copy

Deep Copy copies an object's value and reference types into a complete new copy of the target objects.



7

BISHOY NABIL
وبس

بنعمل بروجيكت ويندوز

فورم

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CopyWindowsDemo
{
    class Employee
    {
        public string ID { get; set; }
        public string Name { get; set; }
        public Address EmpAddress { get; set; }

        #region Copy Methods
        public Employee ShallowCopy()
        {
            return (Employee)this.MemberwiseClone();
        }
    }
}

```

```

public Employee DeepCopy()
{
    Employee other = (Employee)this.MemberwiseClone();
    other.EmpAddress = new Address(this.EmpAddress.DoorNumber,
        this.EmpAddress.StreetNumber, this.EmpAddress.Zipcode,
        this.EmpAddress.Country);
    return other;
}

public override string ToString()
{
    return string.Format("Emp ID :{0}, Emp Name : {1}, {2}",
        this.ID, this.Name, this.EmpAddress.ToString());
}

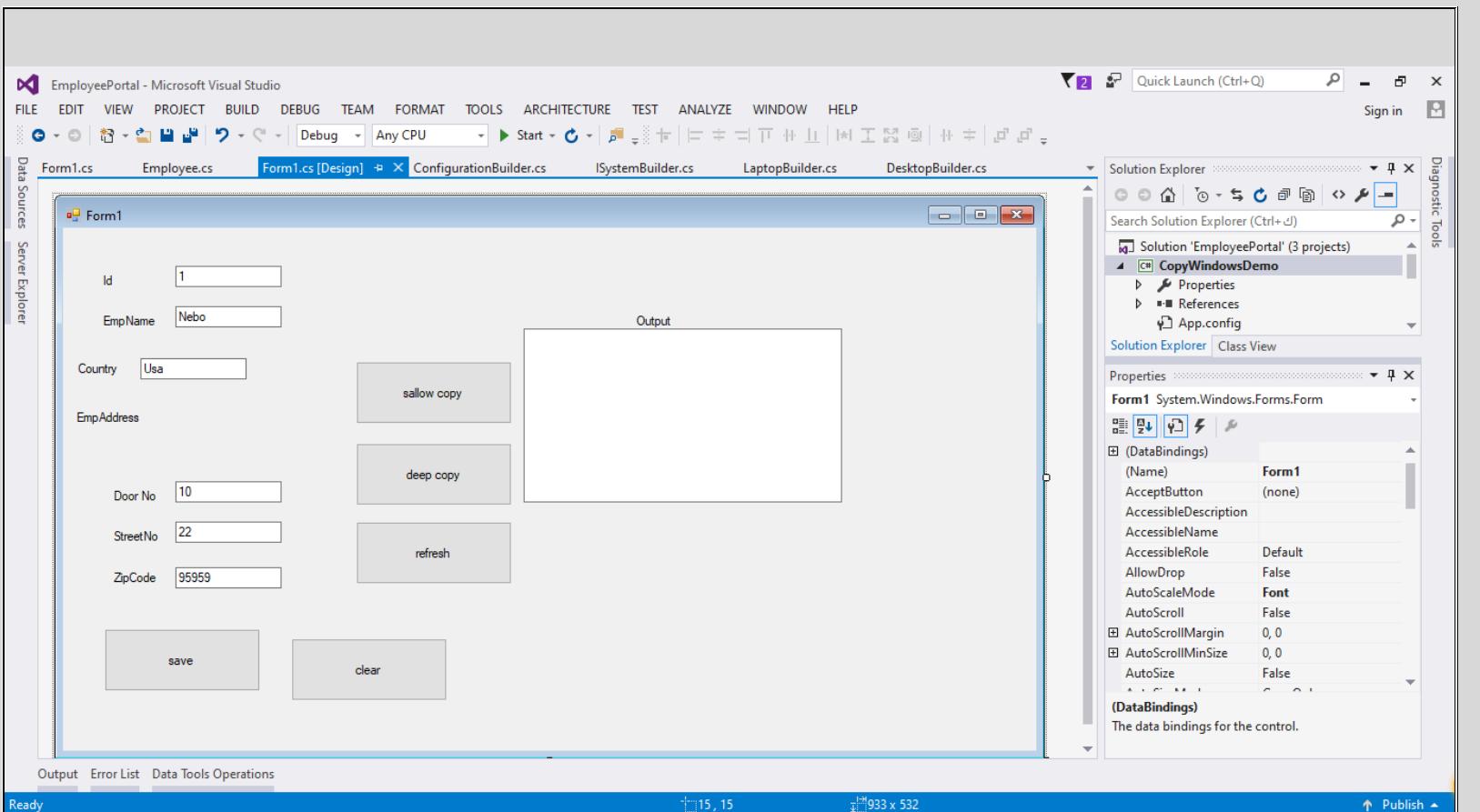
#endregion
}

public class Address
{
    public Address() { }
    public Address(int doorNumber, int streetNumber,
        int zipCode, string country)
    {
        this.Country = country;
        this.DoorNumber = doorNumber;
        this.StreetNumber = streetNumber;
        this.Zipcode = zipCode;
    }

    public int DoorNumber { get; set; }
    public int StreetNumber { get; set; }
    public int Zipcode { get; set; }
    public string Country { get; set; }
    public override string ToString()
    {
        return string.Format(Environment.NewLine + "Emp Address: {0}",
            string.Format("{0}, {1}, {2}, {3}",
                this.DoorNumber, this.StreetNumber,
                this.Zipcode.ToString(), this.Country));
    }
}

```

عمل کلاس



BISHOY NABIL الفورم

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

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

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {
```

```
}

Employee emp = new Employee() { EmpAddress = new Address() };
Employee empCopied;

StringBuilder sb = new StringBuilder();

private void button3_Click(object sender, EventArgs e)
{
    empCopied = (Employee)this.emp.ShallowCopy();
    Refresh();
}

private void button4_Click(object sender, EventArgs e)
{
    empCopied = (Employee)this.emp.DeepCopy();
    Refresh();
}

private void button1_Click(object sender, EventArgs e)
{
    //lblResult.ResetText();
    emp.ID = textBox1.Text;
    emp.Name = textBox2.Text;
    emp.EmpAddress.Country = textBox6.Text;
    emp.EmpAddress.DoorNumber = int.Parse(textBox4.Text);
    emp.EmpAddress.StreetNumber = int.Parse(textBox5.Text);
    emp.EmpAddress.Zipcode = int.Parse(textBox3.Text);
    MessageBox.Show("Updated");
}

private void Refresh()
{
    sb.Clear();
    sb.AppendLine("-----");
    sb.AppendLine(string.Format("Main Employee : {0} ",
        this.emp.ToString()));
    sb.AppendLine();
    sb.AppendLine(string.Format("Copied Employee : {0} ",
        empCopied.ToString()));
    sb.AppendLine("-----");
    sb.AppendLine(textBox7.Text);

    textBox7.Text = sb.ToString();
}

private void button5_Click(object sender, EventArgs e)
{
    Refresh();
}
```

}{

BISHOYNABIL

Form1

Id	1
EmpName	Nebo
Country	Usa
EmpAddress	
Door No	10
StreetNo	22
ZipCode	95959

sallow copy
deep copy
refresh

save clear

Output

Updated

OK

```
graph LR; save[save] --> Updated{Updated};
```

Form1

Id	1
EmpName	Nebo
Country	Usa
EmpAddress	
Door No	10
StreetNo	22
ZipCode	95959

sallow copy
deep copy
refresh

save clear

Output

Main Employee : Emp ID :1, Emp Name : Nebo,
Emp Address: 22, 95959, 10, Usa

Copied Employee : Emp ID :1, Emp Name : Nebo,
Emp Address: 22, 95959, 10, Usa

```
graph LR; sallow[sallow copy] --> MainData[Main Employee]; sallow --> CopiedData[Copied Employee];
```

شوف فوق الاصل تحت الكوبى مفيش فرق هنغير البيانات ونحفظ ونعمل ريفresh

Form1

Id	<input type="text" value="1"/>
EmpName	<input type="text" value="Nebo"/>
Country	<input type="text" value="Usa"/>
EmpAddress	
Door No	<input type="text" value="10"/>
StreetNo	<input type="text" value="22"/>
ZipCode	<input type="text" value="95959"/>

sallow copy

deep copy

refresh

save

clear

Output

```
Main Employee : Emp ID :1, Emp Name : Nebo,  
Emp Address: 22, 95959, 10, Usa  
  
Copied Employee : Emp ID :1, Emp Name : Nebo,  
Emp Address: 22, 95959, 10, Usa
```

Form1

Id	<input type="text" value="1"/>
EmpName	<input type="text" value="Nebo"/>
Country	<input type="text" value="Usa"/>
EmpAddress	
Door No	<input type="text" value="11"/>
StreetNo	<input type="text" value="23"/>
ZipCode	<input type="text" value="95959"/>

sallow copy

deep copy

refresh

save

clear

Output

```
Main Employee : Emp ID :1, Emp Name : Nebo,  
Emp Address: 23, 95959, 11, Usa  
  
Copied Employee : Emp ID :1, Emp Name :  
Nebo,  
Emp Address: 23, 95959, 11, Usa  
  
-----  
  
Main Employee : Emp ID :1, Emp Name : Nebo,  
Emp Address: 22, 95959, 10, Usa  
  
Copied Employee : Emp ID :1, Emp Name :  
Nebo,  
Emp Address: 22, 95959, 10, Usa  
  
-----
```

غيرت البيانات زحفت وبعدين عملت ريفرش

Form1

Id	<input type="text" value="1"/>
EmpName	<input type="text" value="Nebo"/>
Country	<input type="text" value="Usa"/>
EmpAddress	
Door No	<input type="text" value="12"/>
StreetNo	<input type="text" value="25"/>
ZipCode	<input type="text" value="123"/>

sallow copy

deep copy

refresh

save

clear

Output

```
Main Employee : Emp ID :1, Emp Name : Nebo,  
Emp Address: 25, 123, 12, Usa  
  
Copied Employee : Emp ID :1, Emp Name :  
Nebo,  
Emp Address: 25, 123, 12, Usa
```

بنجرب الديب كوبى

Form1

Id	<input type="text" value="1"/>
EmpName	<input type="text" value="Nebo"/>
Country	<input type="text" value="Usa"/>
EmpAddress	
Door No	<input type="text" value="13"/>
StreetNo	<input type="text" value="27"/>
ZipCode	<input type="text" value="123"/>

sallow copy

deep copy

refresh

save

clear

Output

```
Main Employee : Emp ID :1, Emp Name : Nebo,  
Emp Address: 27, 123, 13, Usa  
  
Copied Employee : Emp ID :1, Emp Name :  
Nebo,  
Emp Address: 25, 123, 12, Usa  
  
Main Employee : Emp ID :1, Emp Name : Nebo,  
Emp Address: 25, 123, 12, Usa  
  
Copied Employee : Emp ID :1, Emp Name :  
Nebo,  
Emp Address: 25, 123, 12, Usa
```

بعض شوف فيه فرق بين الاوبيجيكت الاصلى والاوبيجيكت الكوبى الفرق واضح والديب كزبى فعليا بيكون فيه 2 ريفرنس واحد خاص بالاوريجال والثانى بالكوبى

The screenshot shows a PowerPoint slide with a dark blue background. The title 'Significance of Copying' is at the top in yellow. Below it is a text block: 'The idea of using copy is to create a new object of the same type without knowing the exact type of the object we are invoking'. Another text block follows: 'Shallow Copy and Deep copy plays prominent role in copying the objects in Prototype Design Pattern'. A third text block: 'Creating the required object once and by creating the subsequent required objects by cloning helps reducing the time for creating the objects'. A fourth text block: 'With Prototype design pattern, based on the requirement situations we can save memory by cloning the objects'. Below these are two examples: 'Example : Adapting to clone an object which consists of many strings (immutable) is a good idea than creating an object' and 'Example : Retrieve some details from DB for each time we create an object. Importing thousands of employee and each employee details need to be associated with company address and other details from database.' A small circular icon with the number '3' is in the bottom left corner.

العديد من الامثلة اللي بنسخدم فيها ال

Prototype

واللى بتتوفرلى مساحه فى الميموري

مثال لو عايز ارجع بعض الداتا من الداتا بيز فكل مرة بحتاج اعمل اوبيجيكت بدلًا فى كل مرة اعمل اوبيجيكت جديد انا ممكن استخدم الكوبى وهيوفرلك الوقت

MemberwiseClone and ICloneable Interface

MemberwiseClone Method

- The MemberwiseClone method is part of `System.Object` and creates a shallow copy of the given object.
- MemberwiseClone Method copies the nonstatic fields of the chosen object to the new object.
- In the process of copying, if a field is a value type, a bit by bit copy of the field is performed. If a field is reference type, the reference is copied but the referenced object is not.

ICloneable Interface

- The ICloneable interface provides with a customized implementation that creates copy of an existing object.
- The ICloneable interface contains one member, the `Clone` method, which is intended to provide support beyond MemberwiseClone method.

Note : in some cases use `Clone` method and use MemberwiseClone within `Clone` method.

4

BISHOY NABI
هنتعرف على الـ 2 ميثود اولاً

MemberwiseClone

يقولك هى ميثود جزء من

System.Object

وال

System

المسئولة عن انشاء الـ

Shallow copy

للاوبجكت بتاعنا

ثانياً الـ

MemberwiseClone

يقولك بتعمل كوبى لـ

Non static filed

للاوجيكت اللى انا عايز انشئه

ثالث

بيقولك فى داخل عملية النسخ لو كان الحقل نوعه

Value type

هيننسخ بت ورا بت

لكن لو العملية شغاله بتاعت الكوبى وكان النوع

Reference

هيننسخ الريفرنس بشكل عام لكن مش هيننسخ الريفرنس او بجيكت

BISHOYNABIL

ICloneable

هى عبارة عن انترفيس لو عايز اعمل

Customization

على عملية الكوبى على او بجيكت موجودة باستخدام الـ

ICloneable

ثانيا

يتحتوى على ميثود واحدة فقط وهيا

Clone method

وبتدعم الـ

MemberwiseClone

ملوحة

يقولك في بعض الحالات يتم استخدام

Clone method

MemberwiseClone

داخل

الـ

BISHOYNABIL

Clone method

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace ProtoType
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            ShallowCopy();
        }
    }
```

```
public partial class Employee
{
    public Guid Id { get; set; }
    public string Name { get; set; }
    public int DepartmentID { get; set; }
    //AddressDetails.ToString()
    public override string ToString()
    {
        return string.Format(" Name : {0} " + "DepartmentID : {1} ",
```

```
        this.Name, this.DepartmentID.ToString());
    }
}
public partial class Employee : ICloneable
{
    public object Clone()
    {
        return this.MemberwiseClone();
    }
}
public partial class Employee
{
    public Address AddressDetails { get; set; }
}

public class Address
{
    public Address() { }

    public int DoorNumber { get; set; }
    public int StreetNumber { get; set; }
    public int Zipcode { get; set; }
    public string Country { get; set; }
    public override string ToString()
    {
        return string.Format("AddressDetails : Door : {0}, Street: {1}, ZipCode : {2},"
            + " Country : {3}", this.DoorNumber, this.StreetNumber, this.Zipcode.ToString(),
            this.Country);
    }
}

public static void ShallowCopy()
{
    Employee empJohn = new Employee()
    {
        Id = Guid.NewGuid(),
        Name = "John",
        DepartmentID = 150,
    };

    Console.WriteLine(empJohn.ToString());

    Employee empSam = (Employee)empJohn.Clone();
    empSam.Name = "Sam Paul";
    Console.WriteLine(empSam.ToString());

    Console.WriteLine("Changed Johns DepartmentID to 161");
    empJohn.DepartmentID = 161;
    Console.WriteLine(empJohn.ToString());
    Console.WriteLine(empSam.ToString());

    Console.ReadLine();
}
```

```
}
```

```
}
```

بنعمل كونسول ابلكشن فى المشروع ونحط الميثود دى

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
... Debug Any CPU ProtoType Start ...
Program.cs + X
ProtoType
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
...
public partial class Employee
{
    public Guid Id { get; set; }
    public string Name { get; set; }
    public int DepartmentID { get; set; }
    //AddressDetails.ToString()
    public override string ToString()
    {
        return string.Format(" Name : {0} " + "DepartmentID : {1} " ,
            this.Name, this.DepartmentID.ToString());
    }
}
public partial class Employee ...
public partial class Employee ...
123 %
Error List Output
Ready Ln 35 Col 38 Ch 38 INS Publish
```

كلas بحط فيه البروبرتس بتاعتي

وهنعمل الاول

Shallow copy

```
EmployeePortal - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Quick Launch (Ctrl+Q) Sign in
Data Sources Server Explorer Solution Explorer Class View Properties
Program.cs + X
11  static void Main(string[] args)
12  {
13      ShallowCopy();
14  }
15  6 references
16  public partial class Employee...
17  6 references
18  public partial class Employee : ICloneable
19  {
20      1 reference
21      public object Clone()
22      {
23          return this.MemberwiseClone();
24      }
25  }
26  6 references
27  public partial class Employee...
28
29
30
31
32
33
34
35
36
37
38
39
40  public class Address...
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56  public static void ShallowCopy()...
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79 }
```

بعمل الميثود دى عشان اقدر اطبع البيانات وهنا بتعمل اوفريد على ال
Icolonable

وبعملها كاستميز وبحط كلاس الموظفين بداخلها كلاس ال

Icloneable

انهريتن يعني

return this.MemberwiseClone();

الميثود دى هترجعلى نفس الاوجيكت اللي بعمله كوبى فالاتالى الاوجيكت اللي بتعمل منه كوبى هي فعليا هترجعه ونوعه

Object

كما واضح فى الكود

The screenshot shows the Microsoft Visual Studio interface with the title bar "EmployeePortal - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, WINDOW, and HELP. The toolbar has icons for file operations like Open, Save, and Print. The status bar at the bottom shows "Ready", "Ln 56", "Col 41", "Ch 41", and "INS".

The code editor window displays a file named "Program.cs*". The code implements a shallow copy of an Employee object:

```
public static void ShallowCopy()
{
    Employee empJohn = new Employee()
    {
        Id = Guid.NewGuid(),
        Name = "John",
        DepartmentID = 150,
    };

    Console.WriteLine(empJohn.ToString());

    Employee empSam = (Employee)empJohn.Clone();
    empSam.Name = "Sam Paul";
    Console.WriteLine(empSam.ToString());

    Console.WriteLine("Changed Johns DepartmentID to 161");
    empJohn.DepartmentID = 161;
    Console.WriteLine(empJohn.ToString());
    Console.WriteLine(empSam.ToString());

    Console.ReadLine();
}
```

The code uses the `Guid.NewGuid()` method to generate a new GUID for the employee's ID. It demonstrates cloning an object using the `Clone()` method and modifying a copy of the object without changing the original. The `ToString()` method is used to print the employee details to the console.

الميثود دى بتعمل الاسم والايدي ونكمel BISHOYNABIL

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Title Bar:** EmployeePortal - Microsoft Visual Studio
- Menu Bar:** FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, WINDOW, HELP
- Toolbar:** Includes icons for Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Quick Launch:** Quick Launch (Ctrl+Q) button.
- User Account:** Sign in button.
- Code Editor:** The file `Program.cs` is open, showing the following C# code:

```
public static void ShallowCopy()
{
    Employee empJohn = new Employee()
    {
        Id = Guid.NewGuid(),
        Name = "John",
        DepartmentID = 150,
    };

    Console.WriteLine(empJohn.ToString());

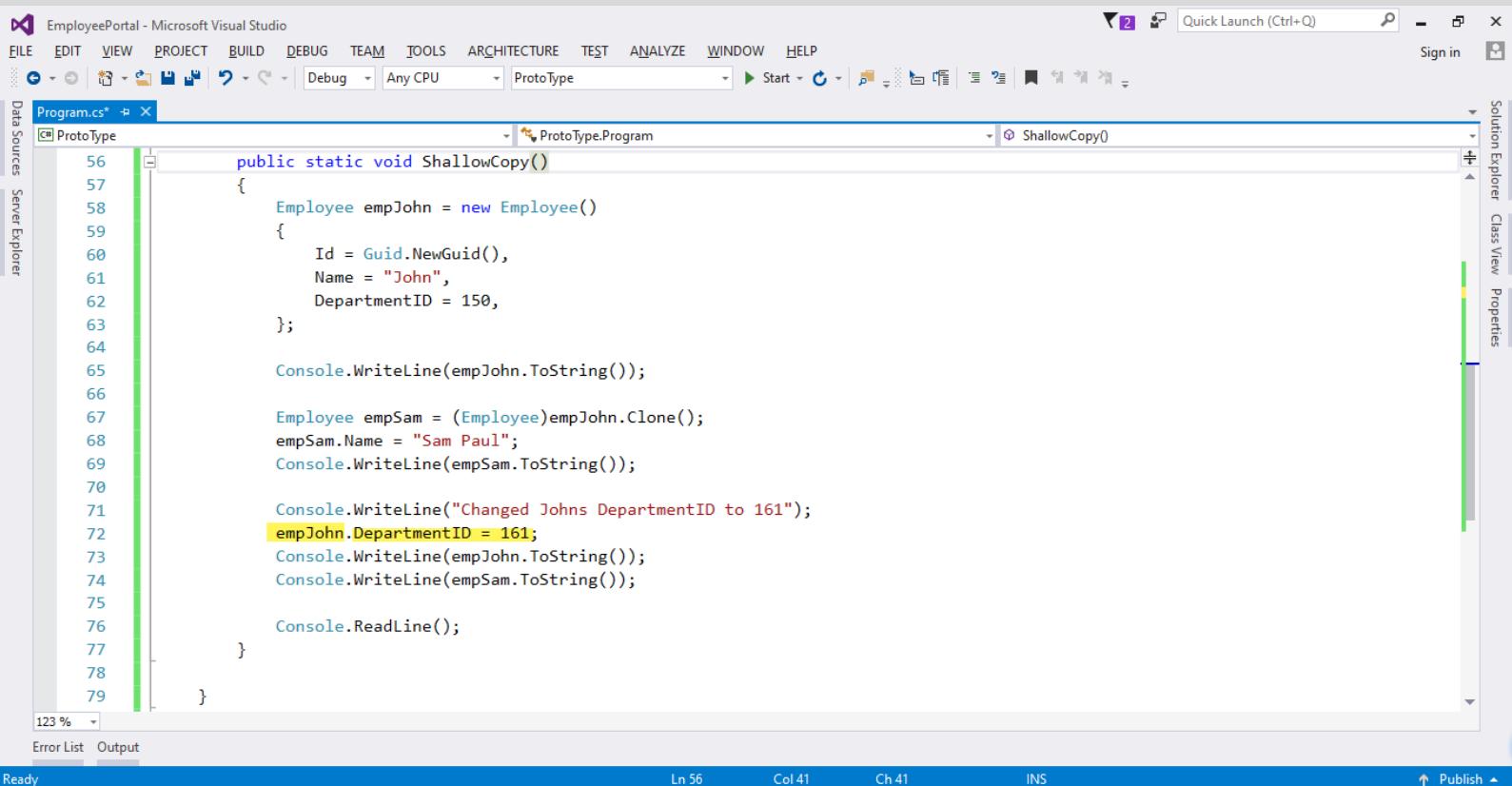
    Employee empSam = (Employee)empJohn.Clone();
    empSam.Name = "Sam Paul";
    Console.WriteLine(empSam.ToString());

    Console.WriteLine("Changed Johns DepartmentID to 161");
    empJohn.DepartmentID = 161;
    Console.WriteLine(empJohn.ToString());
    Console.WriteLine(empSam.ToString());

    Console.ReadLine();
}
```
- Solution Explorer:** Shows the project `ProtoType` with a file `Program.cs`.
- Properties:** Properties tab selected in the Solution Explorer.
- Status Bar:** Displays 123%, Line 56, Col 41, Ch 41, INC.

هنا بعمل طباعه للاوبجيكت بتاعى الموظف وانزل تحت
عمل متغير جديد
.clone

بالشكل ده هجيب الاوبجيكت الاصلى بتاعى واحد منه كوبى
وهنعدل عليه ونطبع البيانات



The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "EmployeePortal - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, WINDOW, and HELP. The toolbar has various icons for file operations like Open, Save, and Build. The status bar at the bottom shows "Ready", "Ln 56", "Col 41", "Ch 41", "INS", and "Publish".

The main code editor window displays the following C# code:

```
public static void ShallowCopy()
{
    Employee empJohn = new Employee()
    {
        Id = Guid.NewGuid(),
        Name = "John",
        DepartmentID = 150,
    };

    Console.WriteLine(empJohn.ToString());

    Employee empSam = (Employee)empJohn.Clone();
    empSam.Name = "Sam Paul";
    Console.WriteLine(empSam.ToString());

    Console.WriteLine("Changed Johns DepartmentID to 161");
    empJohn.DepartmentID = 161;
    Console.WriteLine(empJohn.ToString());
    Console.WriteLine(empSam.ToString());

    Console.ReadLine();
}
```

The code creates an employee object named empJohn and prints its details. It then creates a shallow copy of empJohn named empSam and changes empSam's name to "Sam Paul". Both objects are then printed again. Finally, the department ID of empJohn is changed to 161, and both objects are printed once more.

هنا بعد الاوبجيكت الاصلى بغير رقم القسم للاوبجيكت الاصلى
المفترض بعد التعديلات يطبع جون و 161

The screenshot shows the Microsoft Visual Studio interface. The title bar reads "EmployeePortal - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, ARCHITECTURE, TEST, ANALYZE, WINDOW, and HELP. The toolbar has icons for file operations like Open, Save, and Build. The status bar at the bottom shows "Ready", "Ln 56", "Col 41", "Ch 41", and "INS". A watermark in the center of the screen displays the Arabic text "هنا هي بطبعى" above "BISHOYNABIL" and "Sam".

```
public static void ShallowCopy()
{
    Employee empJohn = new Employee()
    {
        Id = Guid.NewGuid(),
        Name = "John",
        DepartmentID = 150,
    };

    Console.WriteLine(empJohn.ToString());

    Employee empSam = (Employee)empJohn.Clone();
    empSam.Name = "Sam Paul";
    Console.WriteLine(empSam.ToString());

    Console.WriteLine("Changed Johns DepartmentID to 161");
    empJohn.DepartmentID = 161;
    Console.WriteLine(empJohn.ToString());
    Console.WriteLine(empSam.ToString());

    Console.ReadLine();
}
```

ورقم القسم

150

والمفروض انه ميتاشرش بالتعديلات لانه قولنا قبل كده

انه لو الحقل نوعه

Value type

هيتم النسخ بت ورا بت

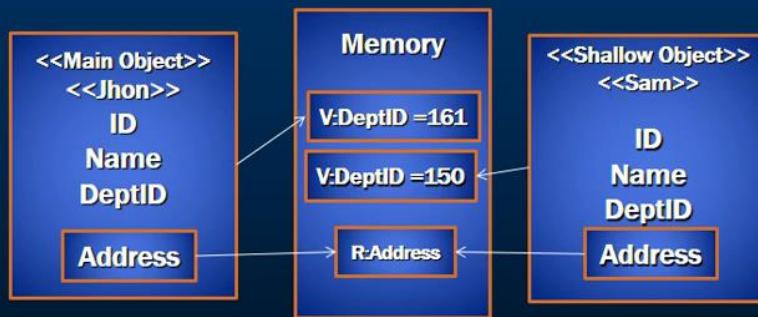
```
C:\Windows\system32\cmd.exe
Name : John DepartmentID : 150
Name : Sam Paul DepartmentID : 150
Changed Johns DepartmentID to 161
Name : John DepartmentID : 161
Name : Sam Paul DepartmentID : 150
```

تمام فى الاول قبل التعديل تحت بعد التعديل وبالفعل
Sam
لم يتأثر بالتعديلات

Prototype Design Pattern Implementation

Step 3 : Invoke Shallow Copy method from Console main program and observe the below output. Notice that change in department doesn't impact the target object

```
cmd C:\Windows\system32\cmd.exe
Name : John DepartmentID : 150
Name : Sam Paul DepartmentID : 150
Changed Johns DepartmentID to 161
Name : John DepartmentID : 161
Name : Sam Paul DepartmentID : 150
```



على الشمال عدل على رقم القسم وهو فاليو تايب
فخزنه في الميموري الشكل اللي في النص ونكمel
وفي الشالو كوبى عدل على رقم القسم فكان

سام

رقم القسم بتاعه

150 عشان فاليو تايب المهم نكمel

تعالي نعمل
امبلميتنشن لـ
Deep copy

A screenshot of Microsoft Visual Studio showing the code editor for a C# project named "ConsoleApplication2". The file "Program.cs" is open, displaying the following code:

```
35 public partial class Employee{...}
39
40     2 references
41     public class Address
42     {
43         0 references
44         public Address() { }
45
46         1 reference
47         public int DoorNumber { get; set; }
48         1 reference
49         public int StreetNumber { get; set; }
50
51         1 reference
52         public int Zipcode { get; set; }
53
54         7 references
55         public override string ToString()
56         {
57             return string.Format("AddressDetails : Door : {0}, Street: {1}, ZipCode : {2},"
58                 " Country : {3}", this.DoorNumber, this.StreetNumber, this.Zipcode.ToString(),
59                 this.Country);
60         }
61     }
62 }
```

The code editor shows the "Address" class definition, which contains properties for DoorNumber, StreetNumber, Zipcode, and Country, along with a ToString() method. The "Address" class has 2 references from other parts of the application.

عمل کلاس جديد عشان اظهر تفاصيل اخرى عن العنوان زى الرقم البريد
وعشان استخدمه لازم استخدم کلاس

Partial

اللى هو ده وبحط فيه كلاس العنوان

ConsoleApplication2 - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Server Explorer Solution Explorer Class View Properties

Program.cs + X

```
ConsoleApplication2 ConsoleApplication2.Program.Employee AddressDetails
30      public object Clone()
31      {
32          return this.MemberwiseClone();
33      }
34  6 references
35  public partial class Employee
36  {
37      0 references
38      public Address AddressDetails { get; set; }
39  }
40  2 references
41  public class Address
42  {
43      0 references
44      public Address() { }
45
46      1 reference
47      public int DoorNumber { get; set; }
48      1 reference
49      public int StreetNumber { get; set; }
50      1 reference
51      public int Zipcode { get; set; }
52      1 reference
53      public string Country { get; set; }
54
55
56      7 references
57      public override string ToString()
58      {
59          return string.Format("AddressDetails : Door : {0}, Street: {1}, ZipCode : {2},"
60          "Country : {3}", this.DoorNumber, this.StreetNumber, this.Zipcode.ToString(),
61          this.Country);
62      }
63
64
65      1 reference
66      public static void ShallowCopy()...
67
68  }
```

123 % Error List Output

Ln 35 Col 38 Ch 38 INS

Ready Publish

الميثود دى بتطبعلى تفاصيل العنوان

ConsoleApplication2 - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Server Explorer Solution Explorer Class View Properties

Program.cs + X

```
ConsoleApplication2 ConsoleApplication2.Program.Employee AddressDetails
42      public Address() { }
43
44      1 reference
45      public int DoorNumber { get; set; }
46      1 reference
47      public int StreetNumber { get; set; }
48      1 reference
49      public int Zipcode { get; set; }
50      1 reference
51      public string Country { get; set; }
52
53      7 references
54      public override string ToString()
55      {
56          return string.Format("AddressDetails : Door : {0}, Street: {1}, ZipCode : {2},"
57          "Country : {3}", this.DoorNumber, this.StreetNumber, this.Zipcode.ToString(),
58          this.Country);
59      }
60
61
62      1 reference
63      public static void ShallowCopy()...
64
65  }
```

123 % Error List Output

Ln 35 Col 38 Ch 38 INS

Ready Publish

ConsoleApplication2 - Microsoft Visual Studio

FILE EDIT PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Server Explorer Solution Explorer Class View Properties

Program.cs

```
14
15
16     public partial class Employee
17     {
18         public Guid Id { get; set; }
19         public string Name { get; set; }
20         public int DepartmentID { get; set; }
21     //AddressDetails.ToString()
22     public override string ToString()
23     {
24         return string.Format(" Name : {0} " + "DepartmentID : {1} {2} ",
25             this.Name, this.DepartmentID.ToString() , this.AddressDetails);
26     }
27 }
28 public partial class Employee : ICloneable
29 {
30     public object Clone()
31     {
32         return this.MemberwiseClone();

```

123 % Error List Output

Ln 25 Col 82 Ch 82 INS

Item(s) Saved Publish

بزود على الطياعه كمان تفاصيل العنوان

ConsoleApplication2 - Microsoft Visual Studio

FILE EDIT PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Server Explorer Solution Explorer Class View Properties

Program.cs

```
40
41
42     public class Address...
43
44
45
46     public static void ShallowCopy()...
47
48
49
50     public static void ShallowCopyRef()
51     {
52         Employee empJohn = new Employee()
53         {
54             Id = Guid.NewGuid(),
55             Name = "John",
56             DepartmentID = 150,
57             AddressDetails = new Address()
58             {
59                 DoorNumber = 10,
60                 StreetNumber = 20,
61                 Zipcode = 90025,
62                 Country = "US"
63             }
64         };
65
66         Console.WriteLine(empJohn.ToString());
67
68         Employee empSam = (Employee)empJohn.Clone();
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97

```

123 % Error List Output

Ln 115 Col 1 Ch 1 INS

Item(s) Saved Publish

بعمل ميثود جديدة وادخل البيانات وتفاصيل العنوان

A screenshot of Microsoft Visual Studio showing a C# code editor. The code demonstrates shallow copy by cloning an Employee object and modifying its properties. The code is as follows:

```
        }
    };

    Console.WriteLine(empJohn.ToString());

    Employee empSam = (Employee)empJohn.Clone();
    empSam.Name = "Sam Paul";
    empSam.DepartmentID = 151;
    empSam.AddressDetails.StreetNumber = 21;
    empSam.AddressDetails.DoorNumber = 11;

    Console.WriteLine(empSam.ToString());

    Console.WriteLine("Modified Details of John");
    empJohn.AddressDetails.DoorNumber = 30;
    empJohn.AddressDetails.StreetNumber = 40;

    empJohn.DepartmentID = 160;
    Console.WriteLine(empJohn.ToString());
    Console.WriteLine(empSam.ToString());

    Console.ReadLine();
}
```

هنا باحد نسخه من الاوبيكت الاصل جون واحطها فى متغير جديد اسمه سام
وبدخله بيانات جديدة واطبعها واروح انزل تحت اعجل على جون واطبع بياناته وتحت خالص
اطبع سام

ConsoleApplication2 - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q)

Sign in

Server Explorer

Solution Explorer Class View Properties

Program.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApplication2
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            ShallowCopyRef(); -----^
14        }
15    }
16    public partial class Employee...
17    {
18        public partial class Employee ...
19        {
20            public partial class Employee ...
21        }
22    }
23    public class Address...
24}
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

123 % Error List... Output

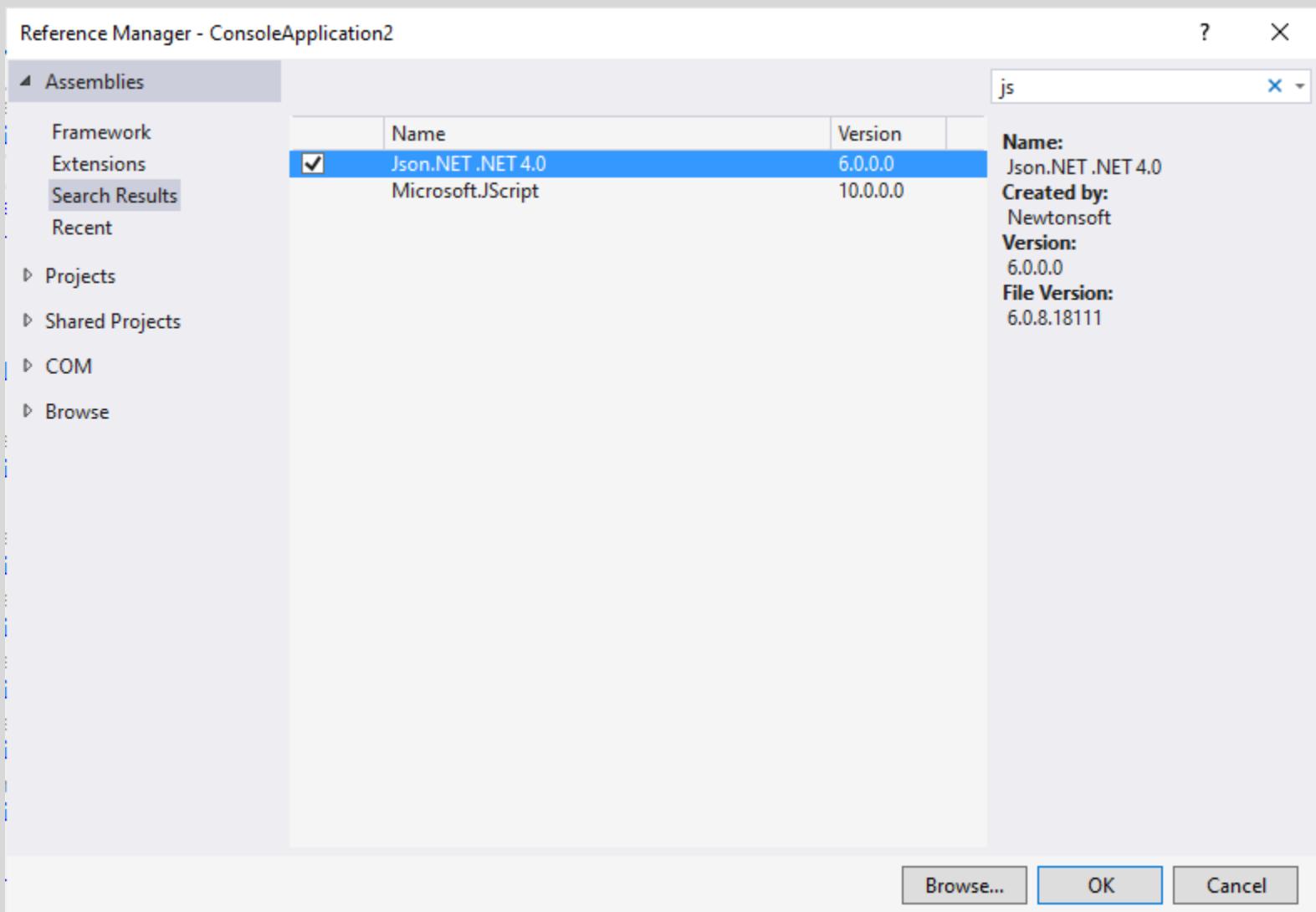
Ln 13 Col 27 Ch 27 INS

c:\Windows\system32\cmd.exe

```
Name : John DepartmentID : 150 AddressDetails : Door : 10, Street: 20, ZipCode : 90025, Country : US
Name : Sam Paul DepartmentID : 151 AddressDetails : Door : 11, Street: 21, ZipCode : 90025, Country :
US
Modified Details of John
Name : John DepartmentID : 160 AddressDetails : Door : 30, Street: 40, ZipCode : 90025, Country : US
Name : Sam Paul DepartmentID : 151 AddressDetails : Door : 30, Street: 40, ZipCode : 90025, Country :
US
```

شوف حصل ايه جون سام فوق تمام
نزلت عدلت تحت على جون

سام اتأثر بالتعديلات هو كمان واحد التعديلات وتأثر بيه
ليه اتأثر
لأنه نوعه رiferنس تايب
زي ماقولنا في الشرح في ال
MemberwiseClone



```
Console.ReadLine();  
}  
1 reference  
public abstract class CloneablePrototype<T>  
{  
    // Shallow copy  
    0 references  
    public T Clone()  
    {  
        return (T)this.MemberwiseClone();  
    }  
  
    // Deep Copy  
    1 reference  
    public T DeepCopy()  
    {  
        string result = JsonConvert.SerializeObject(this);  
        return JsonConvert.DeserializeObject<T>(result);  
    }  
}  
15 references  
public partial class Employee : CloneablePrototype<Employee>  
{  
    3 references
```

```
15 references  
public partial class Employee : CloneablePrototype<Employee>  
{  
    3 references  
    public Guid Id { get; set; }  
    7 references  
    public string Name { get; set; }  
  
    9 references  
    public int DepartmentID { get; set; }  
  
    16 references  
    public override string ToString()  
    {  
        return string.Format(" Name : {0}, DepartmentID : {1} , " +  
            "Address : {2}",  
            this.Name, this.DepartmentID.ToString(),  
            this.AddressDetails.ToString());  
    }  
}  
0 references  
private static void PrototypeDemo()  
{  
    Employee empJohn = new Employee()
```

بنعدل الكلاس بتاعنا ببورث ال

Clone proto

BISHOYNABIL

ConsoleApplication2 - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start | To String() Sign in

Program.cs

```
private static void PrototypeDemo()
{
    Employee empJohn = new Employee()
    {
        Id = Guid.NewGuid(),
        Name = "John",
        DepartmentID = 150,
        AddressDetails = new Address()
        {
            DoorNumber = 10,
            StreetNumber = 20,
            Zipcode = 90025,
            Country = "US"
        }
    };
    Console.WriteLine(empJohn.ToString());

    Employee empSam = (Employee)empJohn.DeepCopy();

    empSam.Name = "Sam Paul";
    empSam.DepartmentID = 151;
    empSam.AddressDetails.StreetNumber = 21;
    empSam.AddressDetails.DoorNumber = 11;
}
```

123 % 69% CPU

Error List Output Find Symbol Results

Item(s) Saved

Ln 141 Col 36 Ch 36 INS

ConsoleApplication2 - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP

Debug Any CPU Start | To String() Sign in

Program.cs

```
Console.WriteLine(empJohn.ToString());

Employee empSam = (Employee)empJohn.DeepCopy();

empSam.Name = "Sam Paul";
empSam.DepartmentID = 151;
empSam.AddressDetails.StreetNumber = 21;
empSam.AddressDetails.DoorNumber = 11;

Console.WriteLine(empSam.ToString());

Console.WriteLine("Modified Details of John");
empJohn.AddressDetails.DoorNumber = 30;
empJohn.AddressDetails.StreetNumber = 40;

empJohn.DepartmentID = 160;
Console.WriteLine(empJohn.ToString());
Console.WriteLine(empSam.ToString());
Console.ReadLine();
}
```

123 % 69% CPU

Error List Output Find Symbol Results

Item(s) Saved

Ln 141 Col 36 Ch 36 INS

C:\Windows\system32\cmd.exe

```
Name : John, DepartmentID : 150 , Address : AddressDetails : Door : 10, Street: 20, ZipCode : 90025, Country : US
Name : Sam Paul, DepartmentID : 151 , Address : AddressDetails : Door : 11, Street: 21, ZipCode : 90025, Country : U
S
Modified Details of John
Name : John, DepartmentID : 160 , Address : AddressDetails : Door : 30, Street: 40, ZipCode : 90025, Country : US
Name : Sam Paul, DepartmentID : 151 , Address : AddressDetails : Door : 11, Street: 21, ZipCode : 90025, Country : U
S
```

شوف سام فى الاخر مش متاثر باى حاجة لانى بعمل

BISHOYNABIL
Deep copy