2023

C# LEVEL I PROGRAMMING

```
SpeechSynthesizer _SS = new SpeechSynthesizer

Form1()

Initializer

foreach (o
{
    var vo
    listBo
}
_SS.Volume =
_SS.Rate = 5;
    I
}
```

مقدمة فى البرم بلغة #^ المستوى الأول



إعداد د. وائل السيد زعويل

مدرس بقسم تكنولوجيا التعليم و مدير وحدة تكنولوجيا المعلومات بكلية التربية النوعية - جامعة طنطا مدرب تكنولوجيا المعلومات والتحول الرقمى لاعضاء هيئة التدريس والدراسات العليا - جامعة طنطا

إهداء

there were the first and the first that the

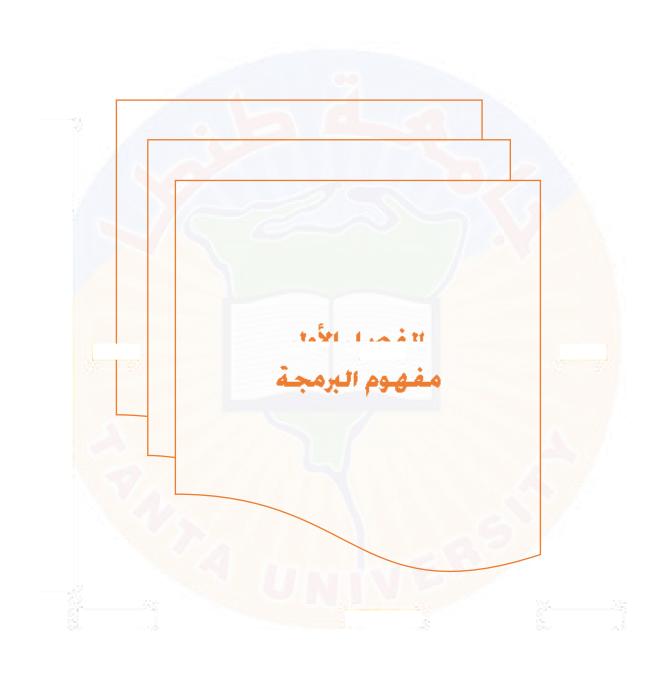
وإلى طلابي بكلية التربية النوعية

ولكل طالب علم ومبتدئ في البرمجة

أهديكم هذا الكتاب

قائمة المحتويات

الصفحة	العنوان	القصل
٧	مفهوم البرمجة	الفصل الأول
71	التعرف على لغة #C	الفصل الثانى
7.1	المتغير ات و الثو ابت	الفصل الثالث
AY	المؤثرات والعمليات الأساسية	القصل الرابع
90	الشرط والتكرار	القصل الخامس
	عامة والنماذج	القصل السادس
157	الأدوات الأساسية	القصل السابع
ነጊነ	مفاهيم برمجيه منقدمة	القصل الثامن
1 / 9	تطبيقى مقدمة في البرمجة	الجزء الثانى





مفهوم لغة البرمجة

يعتمد الحاسب على لغة الآلة التي تعتمد نظام العد الثنائي (۱۰) On-Off التي تعبر عن إيقاف أو تشغيل والتي يصعب على المستخدمين التعامل بها. وبالتالي ظهرت الحاجة إلى ابتكار وتطوير لغات البرمجة التى تسهل التخاطب مع الحاسب وتوجيه الأوامر والتعليمات له ليقوم بتنفيذها بنفس الطريقة التي نتعامل معه بها عند تشغيله أو إيقافه.

ولغة البرمجة عبارة عن رموز نقوم فيها بكتابة تعليمات وإرشادات تخبر الحاسب بالمهام المطلوبة منه، هذه المهام التي تطلبها من الحاسب باستخدام لغة البرمجة تستطيع عمل برامج وألعاب ومواقع ويب، إنشاء تطبيقات الهواتف الذكية والروبوتات وأي شيء يمكنك تخيله على الأجهزة كية (وليس الحاس نفيذه بلغات البرمجة، فهي الو

يستطيع بها المبرمجون فهم لغة الحاسب وإعطائه المهام المطلوبة.

ولغة البرمجة Programming language هي عملية كتابة تعليمات وأوامر لجهاز الحاسب أو أي جهاز إلكتروني آخر، من أجل

توجيهه وإعلامه بكيفية تنفيذ سلسلة من المهام المطلوبة أو كيفية التعامل مع البيانات.

وكل لغة من اللغات المستخدمة في البرمجة لها قواعد خاصة تسمى Syntax والتي يجب على المبرمج الالتزام بها عند البرمجة باستخدامها. والتي عند مخالفة المبرمج لأي منها تظهر له رسالة تفيد بوجود خطأ في القواعد التي تحكم لغة البرمجة أو الصياغة "Syntax Error".

أنواع لغات البرمجة

توجد أنواع مختلفة من اللغات المستخدمة في البرمجة، وهناك العديد من التصنيفات للغات البرمجة فبحسب قرب المفردات المستخدمة فيها والقواعد الأساسية من لغة الإنسان أو لغة الآلة، المستخدمة في ياغتها تنقسم إلى له لي لغات برمجة منخفضة الم ولغات برمجة عالية المستوى

لغات البرمجة عالية المستوي

تتميز لغات البرمجة عالية المستوى High-Level Language بنسبة أعلى من التجريد، مما يعني أنها أقرب إلى لغة الإنسان، وأبعد عن رموز الآلة، كما أنها تتميز بسهولة التعلم والاستخدام وأكثر تلقائية في التعامل معها، ولكنها توفر تحكم مباشر أقل في الحاسب ووظائفه.

كما تُوفر اللغات المستوى إمكانية أضافة الكثر الذي التعليقات، والشروح ضمن البرنامج الذي يعمل عليه المُبرمج؛ الأمر الذي يجعل تلك البرامج أسهل، ويزيد من قابليتها للقراءة، والتعديل من المُستخدِم نفسه، أو حتى من المُستخدِمين الآخرين، تُعدّ كتابة البرامج عبر هذا النوع من اللغات أسهل وأسرع من كتابتها بلغات المستوى المنخفض ومن له هذا النوع من النعات أسهل وأسرع من كتابتها بلغات المستوى المنخفض ومن له هذا النوع من اللغات أسهل وأسرع من كتابتها بلغات المستوى المنخفض ومن له هذا النوع من اللغات أسهل وأسرع من كتابتها بلغات المستوى المنخفض ومن له هذا النوع من اللغات أسهل وأسرع من كتابتها بلغات المستوى المنخفض ومن له هذا النوع من اللغات أسهل وأسرع من كتابتها بلغات المستوى المنخفض ومن له هذا النوع من اللغات أسهل وأسرع من كتابتها بلغات المستوى المنخفض ومن له هذا النوع من اله هذا النوع من اله هذا النوع من اله ولغة اله كاله ولغة النوع من اله ولغة اله كاله ولغة اله هذا النوع من اله ولغة اله كاله ولغة النوع من اله ولغة ا

لغات البرمجة منخفضة المستوي

وعلى عكس اللغات عالية المُستوى فإن اللغات المُنخفضة المُستوى تكون مُعقَّدة وأصعب في التعلم والاستخدام وتحتاج إلى وقت وتعليمات أكثر كثافة للقيام بأمر مكرر قمت بإنشائه مسبقًا وتعد من لغات البرمجة التي تُكتب عبر نظام العد الثنائي Binary Number System المتمثل بالأعداد: • و ١، ومن أمثلتها لغة التجميع Assembly Language، لغة الآلة Machine Language وهي اللغة التي تتمكن الحواسيب فقط من فهمها. وتُعتبر لغة التجميع أسهل نسبياً من لغة الآلة؛ وذلك لاحتوائها على بعض مفردات اللغة الإنجليزية، مما يجعل قراءة برامجها وفهمها أسهل مقارنةً ببرامج لغة الآلة، وتعمل برامج هذا المستوى من لغات البرمجة بمثابة مُترجم يعمل على تحويل برامج اللغات العالية أستوى التي يكتبها لغة الآلة التي تفهمها الأج من لغة الآلة ولغة التجميع هي لغات غير محمولة؛ أي أنه لا يُمكن نقل برنامج مكتوب بها من جهاز حاسب لآخر.

أنواع لغات البرمجة وفق طريقة تنفيذ البرامج

اللغات المفسرة Interpreted Language هي اللغات التي تُترجم أوامرها البرمجية وتُحوّل إلى لغة الآلة أولاً بأول عند كتابتها من قبل المُترجِم، ولا تُترجم برامج هذا النوع من اللغات إلى لغة الآلة الخاصة بمعالج الجهاز نفسه، وإنما باستخدام برنامج خاص باللغة نفسها يُطلق عليه اسم المُفسّر Interpreter، وبعاب عليها البطء لأنها تترجم للغة لآلة أولا بأول ولكنها تبلية تشغيل البرنامج على أجهزة مُختلفة وعلى أنظمة تشغيل مُتعددة، حيث يتطلب ذلك تنزيل نسخة من لغة البرمجة على الجهاز المُراد تشغيل البرنامج عليه، كما تكون أوامر البرامج المكتوبة بهذا النوع من اللغات قابلة للقراءة والتعديل بشكل دائم، ومن أمثلتها Python

اللغات المترجمة Compiled Language هي اللغات التي ألغات التي تُحوّل الكود إلى لغة الآلة الخاصة بالجهاز بعد الانتهاء من كتابة البرنامج كاملاً، ويتم التحويل من لغة البرمجة التي كُتب بها البرنامج إلى لغة

الآلة باستخدام ما يُعرف بالمُترجِم Compiler، ويمتاز هذا النوع من اللغات بسرعة التشغيل؛ حيث يتم تشغيل البرنامج بشكل مُباشر دون الحاجة لتحويل الأوامر أولاً بأول، إلا أنّ برامج هذا النوع من لغات البرمجة لا يُمكن نقله وتشغيله على أجهزة تعمل بأنظمة تشغيل مُختلفة عن تلك التي يحتويها الجهاز الذي تمت كتابة البرنامج عليه. ومن أمثلتها لغة سي (C).

أنواع لغات البرم حيث نمط البرمجة

- برمجة الأوامر Imperative programming

نمط قديم جدا يحاكي الطريقة التي يعمل بها الحاسب من حيث العطاء الأوامر، زمن لغات البرمجة القديمة التي تعتمد عليه C a

- البرمجة الإجرائية Procedural Programming

البرمجة الإجرائية تمثل أول تطوير للبرمجة بالأوامر، والفرق بينهما أن البرمجة الإجرائية تقوم بتقسيم الأوامر إلى مجموعة من بينهما أن البرمجة الإجرائية تقوم بتقسيم الأوامر إلى مجموعة من الإجراءات حتى تكون عملية التنفيذ أسرع. ومن تلك اللغات Java لغة C بغة PHP، لغة C بغة PHP، لغة C بغة المرابعة التنفيذ أسرع.

- البرمجة الوظ ctional Programmin

اللغات الوظيفية تستخدم البيانات المخزنة لإجراء العمليات الحسابية وغيرها من العمليات التي تحتاج للتعامل مع مدخلات سابقة يحتاج الرجوع اليها، ومن أهم هذه اللغات #C و Kotlin.

- البرمجة الشيئية OOP Programming -

البرمجة الشيئية أو (البرمجة كائنية التوجه) تعد أكثر أنماط البرمجة البرمجة البرمجة كائنية التوجه) تعد أكثر أنماط البرمجة استخداماً في الوقت الحالي، وتعمل على تقسيم البرنامج إلى تصنيفات Classes تتكون من وحدات (كائنات Objects) يتم ربطها ببعضها

البعض لبناء البرنامج. والبرمجة كائنية التوجه لها العديد من المميزات فيمكنك بسهولة إعادة استخدام الأكواد الخاصة بك. وسهولة التعاون بين أكثر من مبرمج على نفس المشروع لأن أغلب الأكواد واضحة ومعرفة بشكل يسهل فهمه. ومن أمثلتها لغات ++C)، Objective-C

- البرمجة النصية scripting language

يُقصد بها اللغات شتمل على عناصر لغة موجهة مكنك من كتابة وتتفيذ سطر كود أو أكثر دون الحاجة إلى Compile،

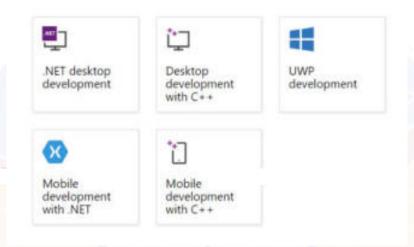
لغات البرمجة من حيث الاستخدام

- تطوير التطبيقات والبرامج: وهي اللغات التي تُعنى بتطوير المج والتطبيقات المُ يتعامل معها المُستخدِم يومياً، المج والتطبيقات المُ Visual Basic ،Java ،C++ ،C# ،C كل من لغات C++ ،C أشهر الأمثلة على هذا النوع من لغات البرمجة.

- برمجة تطبيقات الأندرويد: لتطوير تطبيقات الموبايل ومن

أمثلتها لغة Kotlin.

Desktop & Mobile



- برمجة ألعاب الحاسب نستخدم: وهي التي تُعنى بتطوير الألعاب

أو أي برامج ترفيهية أُخرى كلغة +++ Dark BASIC، C# ،Java، C++.

Gaming



Game development with Unity



Game development with C++

- برمجة الذكاء الإصطناعي: وهي اللغات الخاصة بتطوير برامج الذكاء الاصطناعي؛ كبرامج الروبوتات، ومن تلك اللغات اللغات التي الذكاء الاصطناعي؛ كبرامج الروبوتات، ومن الأمثلة على اللغات التي اللغات التي أمكن استخدامها لتطوير برمجيات الذكاء الاصطناعي المُختلفة.

- لبرمجة قواعد البيانات: وهي اللغات التي يتم استخدامها من قبل مبرمجي قواعد القواعد؛ بهدف إنشاءها، والحفاظ على عملها، ومن الأمثلة لي هذا النوع من اللغا SQL، MySQL، Oracle . SQL،

Other toolsets



Data storage & processing



Data science & analytical applications



Visual Studio extension development



Office/SharePoint



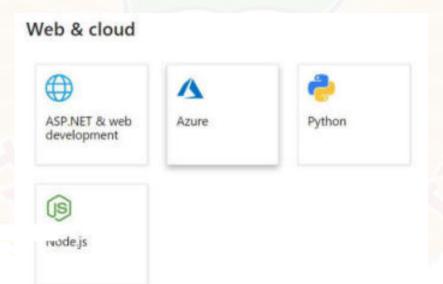
Linux development with C++



.NET Core crossplatform development

- تطوير أنظمة التشغيل: وتُعنى هذه اللغات بتطوير البرامج اللازمة لإنشاء أنظمة تشغيل أجهزة الحاسب وواجهاتها المُختلفة، ومن أهمها لغة Assembly ، C

- لبرمجة صفحات الويب: تُعتبر هذه اللغات أساس وجود الإنترنت، ومن الإنترنت في العالم، حيث يتم من خلالها تصميم مواقع الإنترنت، ومن اللغات HDML ، HTML ، Python ، PHP ، ASP. Net تلك اللغات JavaScript ، Jav



وسيقتصر دراسة البرمجة داخل الكتاب على لغة 2019 #C# 2019. ضمن مجموعة Visual Studio .net.

خرائط التدفق Flow Chart:

هي مجموعة من الرموز البصرية التي يستخدمها المبرمج لصياغة فكرته البرمجية وفق قواعد متعارف عليها مما يسهل على غيره من المبرمجين فهم تلك الخريطة وترجمتها إلى برنامج بأى لغة برمجة، حيث أن خريطة التدفق غير مرتبطة بلغة برمجة معينه.

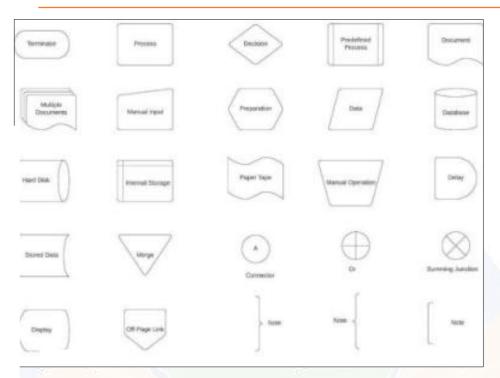
رموز خرائط التدفق الأكثر شيوعا:

	الرمز الطرفي (inal	
	إدخال/ إخراج (Input/ Output)	
	معالجة / عملية (Process)	
	قرار/ اختیار (Decision)	
-	خط اتجاه (Flow line)	

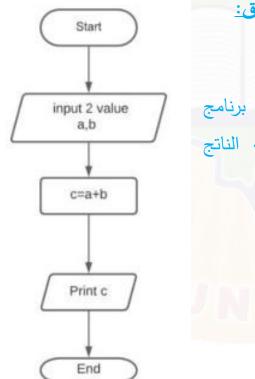
كما يمكنك إنشاء خرائط تدفق إلكترونيه باستخدام أحد المواقع المتعددة التي تتيح تلك الخدمة مثل:

https://lu

وجدير بالذكر أن هناك العديد من الأشكال التي تستخدم في بناء خرائط التدفق لتساعد المبرمج في صياغة أفكاره بشكل سلس يسهل فهمه وتطبيقه وفيما يلي مجموعة من تلك الأشكال:

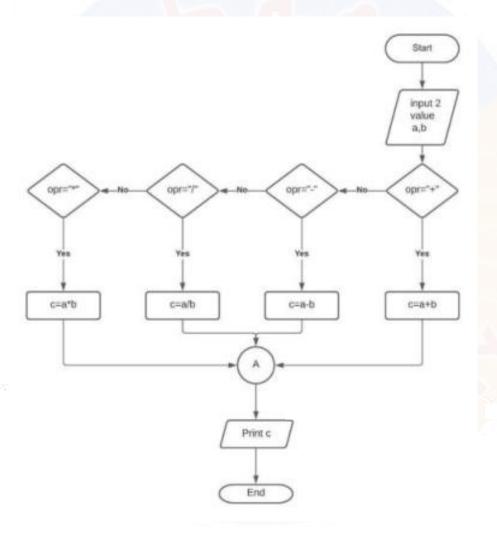






(۱) نموذج لخريطة تدفق برنامج لإدخال رقمين وجمعهم وطباعة الناتج وإنهاء البرنامج

(٢) خريطة تدفق لآلة حاسبة بسيطة يتم إدخال رقمين ونوع العملية الحسابية ثم يختبر متغير العملية الحسابية وبناء على قيمته يجرى عملية الجمع، الطرح، الضرب، أو القسمة ثم يطبع الناتج وإما أن ينهى البرنامج أو يعود ليدخل قيم جديده وهكذا.







Visual Studio 2019 أولا تثبيت 💇

لغة C# .net تأتى كأحد لغات البرمجة المتوفرة في المجموعة

Visual Studio .net وفيما يلى تاريخ تطور لغة #C:

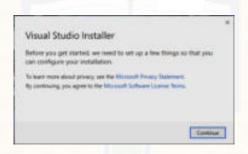
إصدار Visual Studio	إصدار NET.	التاريخ	الإصدار
Visual Studio 2002	.NET Framework 1.0	2002	C# 1.0
Visual Studio 2003	.NET Framework 1.1	2003	C# 1.1 C# 1.2
Visual Studio 2005 Visual Studio 2008	.NET Framework 2.0 .NET Framework 3.0	2005	C# 2.0
Visual Studio 2008	.NET Framework 2.0 .NET Framework 3.0 .NET Fra 3.5	2007	C# 3.0
Studio 2010	.NET Frame	2010	C# 4.0
Visual Studio 2012 Visual Studio 2013	.NET Framework 4.5	2012	C# 5.0
Visual Studio 2015	.NET Framework 4.6 .NET Core 1.0 .NET Core 1.1	2015	C# 6.0
Visual Studio 2017 version 15.0	.NET Framework 4.7	2017	C# 7.0
Visual Studio 2017 version 15.3	.NET Core 2.0	2017	C# 7.1
Visual Studio 2017 version 15.5		2017	C# 7.2
Visual Studio sion 15.7	.NET Core 2.1 .NET Cor .NET Fra	2018	C# 7
Visual Studio 2019 version 16.3	.NET Core 3.0 .NET Core 3.1	2019	C# 8.0
Visual Studio 2019 version 16.8	.NET 5.0	2020	C# 9.0

والإصدار الذي يتناوله الكتاب Visual Studio 2019 والتي تأتى في ٣ إصدارات هي Home للاستخدام المنزلي Visual Studio كالمحترفين و Enterprise والتي تحتوي كافة عناصر وإضافات Visual كلمحترفين و Studio ولتثبيت اللغة نتبع الخطوات التالية:

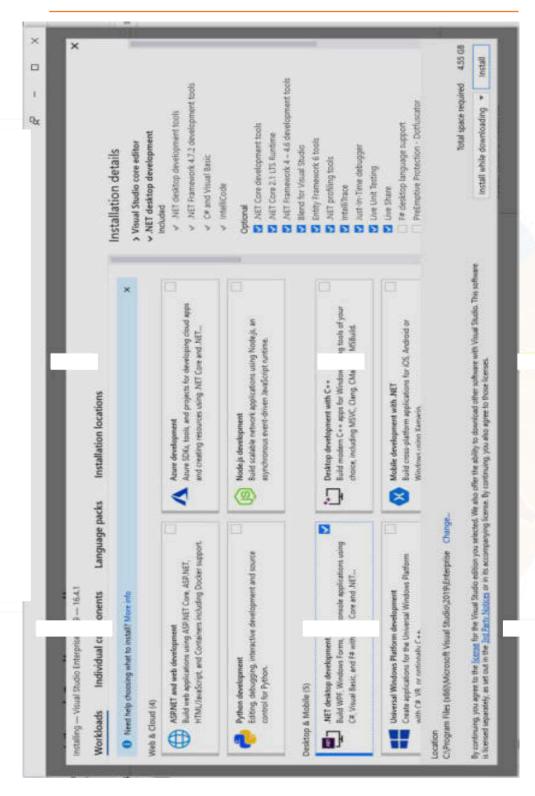
- معرفة متطلبات التثبيت وهل يفى جهازك بها أم لا وتحميل نسخة من Visual Studio 2019 من خلال اللينك التالى:

https://docs.microsoft.com/en-us/visualstudio/releases/2019/system-requirements

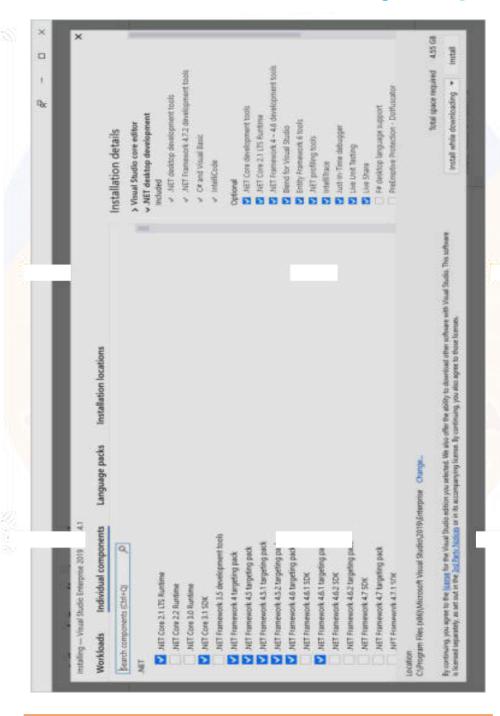
تشغيل ملف setup يسخة فتظهر رسالة التمهيد للتثبي



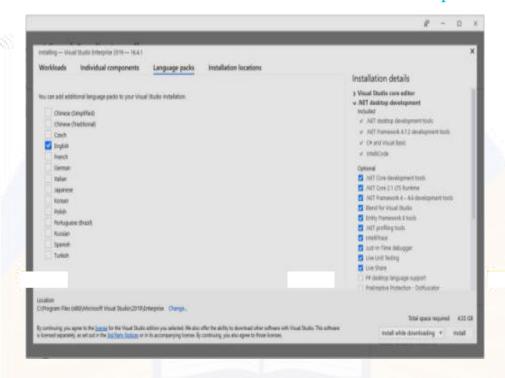
- ثم تظهر نافذة التثبيت الرئيسية والمكونة من أربعة تبويبات الأول Workload ومن خلاله بيتم اختيار اللغة وطبيعة المشاريع والإضافات طلوب تحميلها من Visual Studio. فنختار بشك NET Desktop Development. لتطوير تطبيقات سط بلغة #C كما يمكنك تجربة اختيارات أخرى.



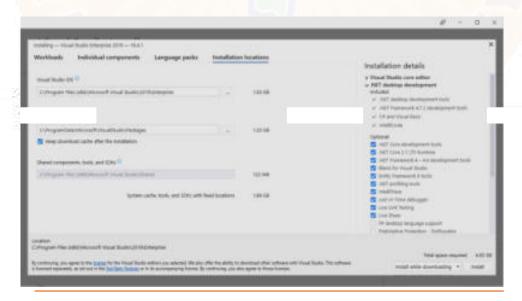
- ويمكن التحكم في العناصر المراد تثبيتها تفصيلا من خلال التبويب الثانى Individual Components



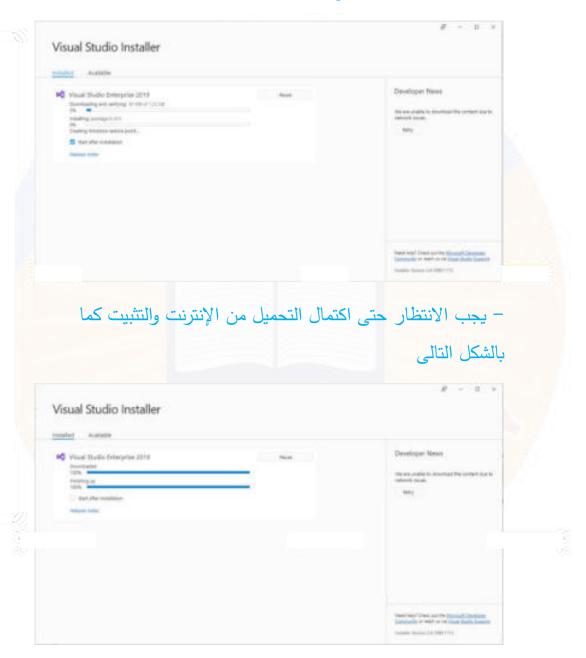
- ويتم اختيار اللغة الإنجليزية من خلال التبويب الثالث packs



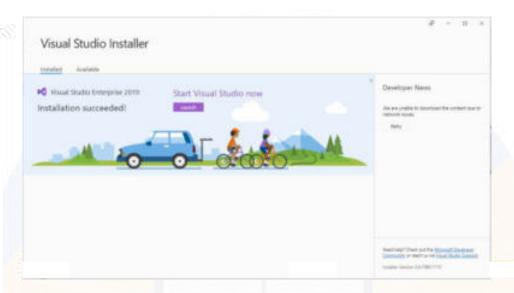
- ويتم اختيار مكان التثبيت من خلال التبويب الرابع Installation لدويت الرابع Locations



- يتم البدء في عملية التثبيت وقد تطلب الاتصال بالإنترنت لتحميل التحديثات خلال تلك الفترة.

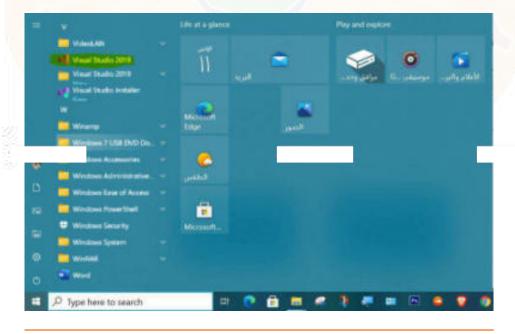


- عند ظهور نافذة اكتمال التثبيت كما في الشكل التالى نقوم بإغلاق برنامج التثبيت وبهذا نكون قد انتهينا من تثبيت . Visual Studio 2019.



تانيا تشغيل Visual Studio لأول مره

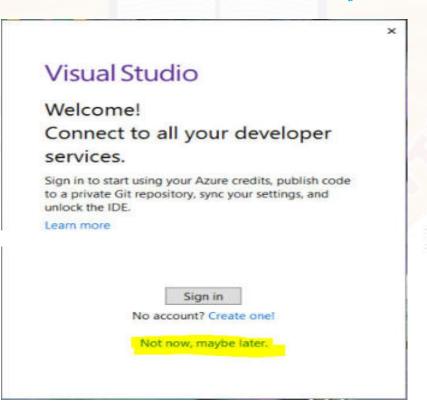
- نذهب الى قائمة Start ثم البحث عن 2019 Visual Studio



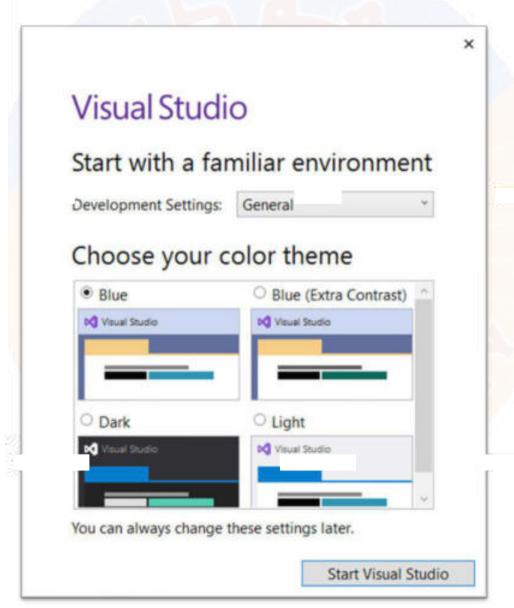
- تظهر نافذة الترحيب لمدة ثواني



- في أول تشغيل للغة تظهر نافذة تسجيل الدخول لحسابك على يكروسوفت ويمكن تخ خاله فيما بعد أو تسجيل الدخول



- في الاستخدام الأول تظهر نافذة ضبط بيئة العمل بحيث تكون بشكل مريح للاستخدام وتتكون من اختيارين أساسيين الأول لغة التطوير ويتم اختيارها من القائمة والثانى نمط الألوان المريح أكثر للعين كما بالشكل التالى



- ويتم اختيار لغة التطوير المستخدمة وهي #C بحيث عند كل تشغيل للبرنامج يبدا بمشاريع لغة #C مباشرة وذلك من خلال الفئة Development Settings. أما الاختيار الثاني الخاص بنمط الألوان فيرجع للمستخدم فله حرية الاختيار بين أنماط الألوان المختلفة التي توفرها بيئة التطوير.

Visual Studio

Start with a familiar — ironment

Development Settings: General

General

General

Visual Basic

Visual Studio

Visual C#

Web Development

Web Development (Code Only)

Dark

O Light

Visual Studio

You can always change these settings later.

Visual Studie

Start Visual Studio

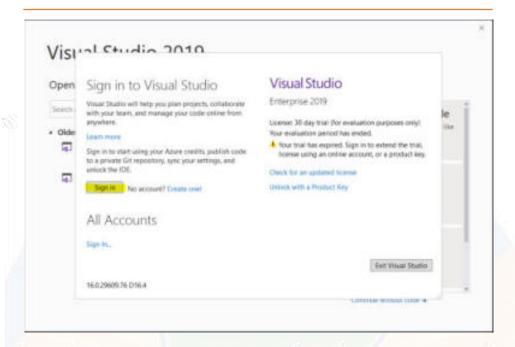
- سيأخذ بعض الوقت لحين فتح البرنامج للمرة الأولى وتطبيق الاعدادات السابقة

Visual Studio

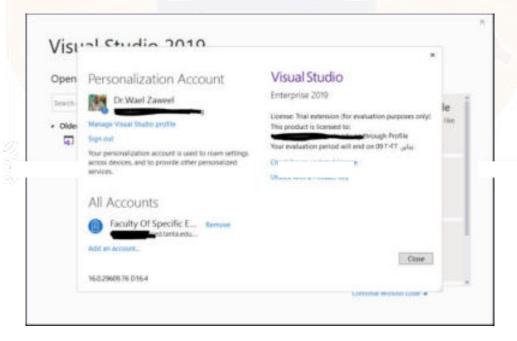
We're preparing for first use

This may take a few minutes.

- في حالة عدم إدخالك لحساب مايكروسوفت مسبقا وانتهاء الفترة التجريبية للبرنامج فإما استخدام Product Key لتفعيل البرنامج أو الدخول باستخدام البريد الجامعي والمنتهي بالدومين التالي لطلبة كلية التربية النوعية sed.tanta.edu.eg والذي يتيح لك عام مجاني للعمل أل البرنامج وللدخو جامعي يتم الضغط على in إدخال بيانات الحساب الخاص بك "البريد الإلكتروني الجامعي" ثم كلمة المرور الخاصة بك.

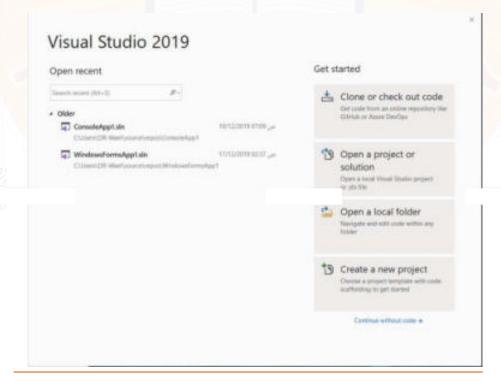


بعد إدخال البريد المسلم الجامعي وكلمة المرور بنجاح النافذة التالية والتي تفيد بنجاح تسجيل الدخول وستجد بها بيانات الحساب والترخيص والفترة الزمنية التي سينتهي فيها.

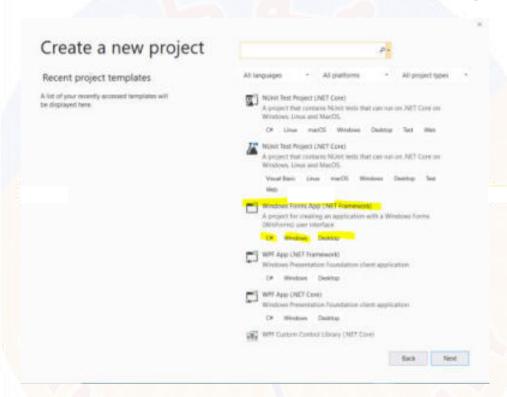


🐷 ثالثًا التعامل مع النافذة الافتتاحية:

- جميع النوافذ السابقة تظهر في أول تشغيل للبرنامج أما بدءا من النافذة التالية فستظهر عند كل تشغيل وهي ببساطة مسئوله عن تحديد ما اذا كنت ستتعامل مع برنامج جديد أو موجود مسبقا وتتكون مما يلى:
 - open Recent: ٥ عبارة عن آخر مجموعة ملفات تم تشغيلها.
 - Clone or check out code: 0 لفتح كود مسبقا محمل على Azure أو Azure ولك حق الدخول والمشاركة عليه.
 - Open Project or Solution: ٥ فتح مشروع محفوظ مسبقا
 - O تصفح وتحرير الأكواد بداخل أ
 - Create a new project: 0 إنشاء مشروع جديد



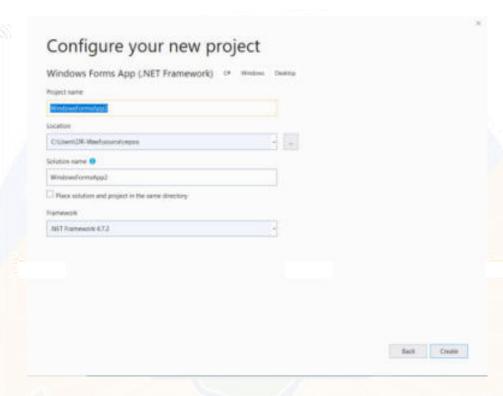
- لإنشاء مشروع جديد بحيث يعمل في بيئة النوافذ Windows يتم اختيار نوع المشروع الجديد App والتحقق من أن اللغة المستخدمة في البرمجة #C وستجدها مكتوبه في البيانات أسفل نوع التطبيق.



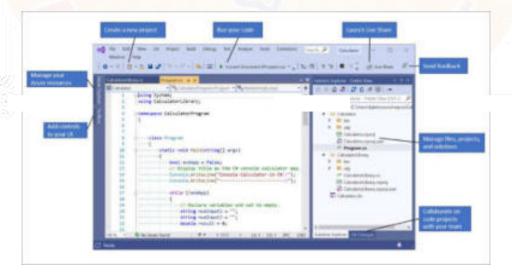
- بعد اختيار إنشاء مشروع جديد وتحديد لغة التطوير ونوع التطبيق راد برمجته يتم تحال الحفال الح

الحل Solution. وتحديد ما إذا كان سيتم حفظ كل من الحل والمشروع .NET Frame في نفس المجلد أم لا وكذلك تحديد إصدار إطار العمل work الذي سيبنى عليه المشروع وتحديد إصدار إطار العمل هام في

انه يجب أن يكون متوفر لدى المستخدم الذى سيتعامل مع التطبيق الجاري برمجته.



البعا الواجهة الرئيسية لبيئة التطوير IDE



Integrated Development Environment هي اختصار لـ IDE وتعنى بيئة التطوير المتكاملة من مايكروسوفت ومن خلالها يمكننا تطوير التطبيقات باستخدام العديد من لغات البرمجة مثل #C و كا وغيرها من لغات البرمجة التي تتيحها بيئة التطوير والمدمجة داخليا تحت المجموعة البرمجية مما يلى:

- شريط الأدوات القياسى:

P WindowsFormsApplication1

₩ Live Share R

عادة كل تطبيقات ويند ما نبدأ بشريط العنوان والقوائم والعدوي شريط العنوان م المشروع أو الحل، المستخدم ف تسجيل دخولك والقوائم الرئيسية. يلى ذلك شريط الأدوات القياسى ويحتوى على العديد من الأوامر أهمها أوامر الحفظ تشغيل أو تنفيذ المشروع وعمل مشاركة للمشروع مع احد الزملاء.

ile Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (ChileQ)

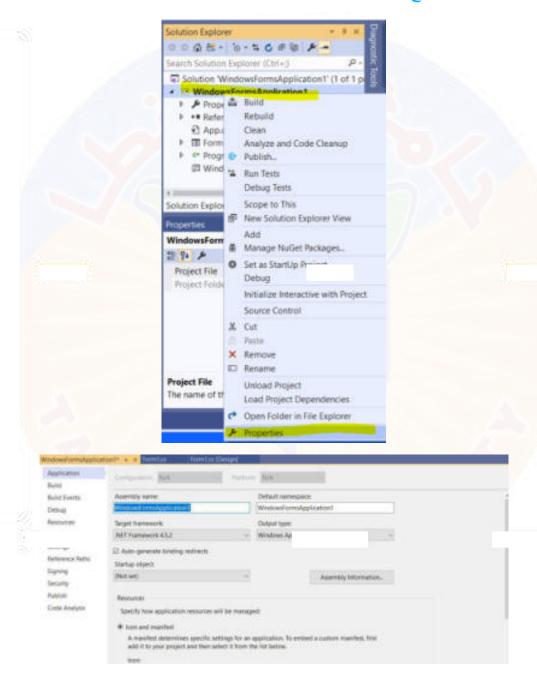
○ 世・益 編 よ ワ・マ・ Debug・ Any CPU ・ F Start・ 声 報 、 由信 また 単 リマツ 。

- متصفح الحل Solution Explorer:

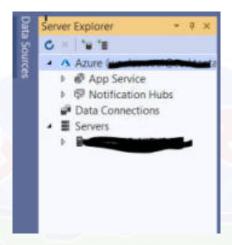
يستخدم #C في بناء حل Solution ولتصفح محتوياته نستخدم النافذة، وجدير باللسمال يتكون من مشروع roject

وكل مشروع يحتوى على العديد من الملفات والمصادر وتستخدم نافذة متصفح الحل في تصفح تلك المحتويات. ويمكن أيضا من خلالها الوصول لنافذة خصائص كل مشروع بالضغط بالزر الأيمن للفاره على

اسم المشروع واختيار خصائص Properties للتعديل في الخصائص العامة للمشروع

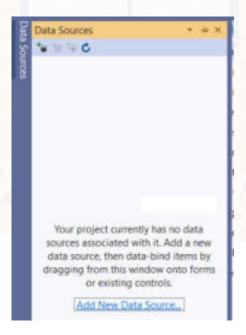


- متصفح الخادم Server Explorer



تصفح المشاريع المشاركة من خلال الخدمات السحابية المختلفة مثل GitHu

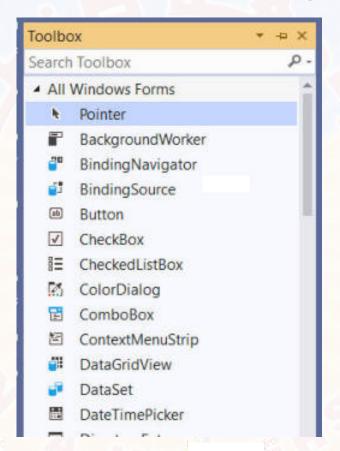
- مصدر قاعدة البيانات Data Source



إضافة وتصفح قاعدة البيانات المرتبطة بالمشروع الحالي

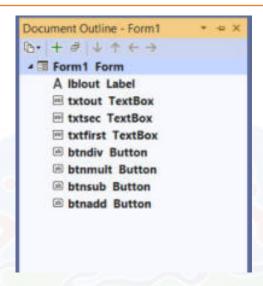
- صندوق الأدوات Toolbox:

يحتوى على العديد من أدوات التحكم مثل الأزرار مربعات النصوص والصور والعناوين وغيرها من الأدوات التي نستخدمها في برمجة المشاريع المختلفة.

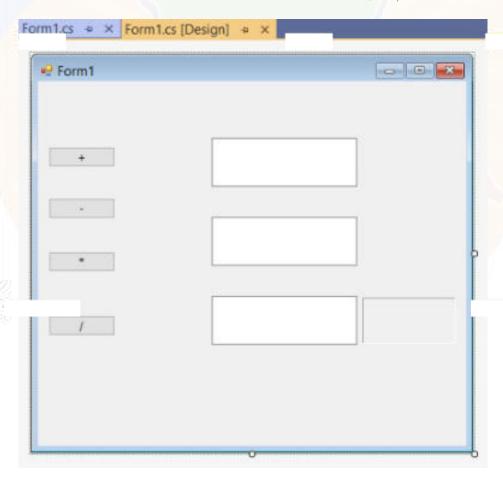


المخطط التفصيلي Document Outline:

من خلاله نحدد شجرة الأدوات فمثلا النموذج يندرج تحته أدوات وقد تحتوي بعض الأدوات على أدوات فرعية بداخلها مثل أداة الإطار Frame



نافذة التصميم Form Design:



وتحتوي على النموذج وبداخله أدوات التحكم المختلفة وتستخدم في تصميم وتنسيق واجهة المشروع الرسومية

- نافذة البرمجة Code:

تستخدم في كتابة التعليمات البرمجية وفقا لقواعد لغة #C والتي سيتم تنفيذها بعد التشغيل

```
| Provided | Provided
```

مع مراعاة أن كل بلوك برمجى يبدا وينتهى بقوس مجموعة "{}"، كل سطر كودى ينتهى بفاصلة منقوطة ";" Simi Colone ولا يشترط أن يكتب كله في نفس السطر حيث يمكن كتابته على اكثر من سطر هم أن ينتهى بفاصل

وبتشريح الكود نجد أنه يحتوي على:

- Methods: عبارة عن أكواد يتم تنفيذها لتؤدى مهمة محددة.
- Properties: مجموعة من الخصائص الخاصة بالـ Classes.
 - Class: أحد مكونات البرنامج في لغة #C والذي يتكون من Methods: مجموعة من الـ Properties.
 - Namespace: مجموعة مختلفة من الـ Classes منظمه.
 - Assembly أو EXE: مجموعة من الـ Assembly



وفيما يلى مثال لتقسيم الكود إلى مجموعه من المناطق التي تتنوع Namespace وحاوية البرنامج Reference بين منطقة المراجع Glasses وحاوية البرنامج والفصائل Classes بداخلها الطرق Methods والدوال وبداخلهم الإعلان عن المتغيرات والأكواد المختلفة.

```
□using System;
                                         Reference of
 using System.Collections.Generic;
                                         .Net Framework Namespaces
 using System.Ling;
 using System. Text;
 using System. Threading. Tasks;
                                         -Namespace name
      space CSharpTutorials *
                               class name
      lass Program 4
                                Method
                                              ___Variable 6
          static void Main(string[] args)
                                                   ____Value of variable
            string message = "Hello World!!"
             Console.WriteLine(message);
                                Method to display value on Console
               مثال لتوضيح م لكود (طباعة جملة llo World
        عند إنشاء تطبيق من النوع CONSOL App فتجد في متصفح
        الحل Solution Explorer العديد من الملفات والتي من بينها ملف
                   Program.cs عند فتحة تحد الكود مكتوب بالشكل التالي
        using System;
        namespace HelloWorld
         class Program
           static void Main(string[] args)
```

```
Console.WriteLine("Hello World!");
}
}
```

النتيجة

Hello World!

تشريح الكود السابق

التفسير	الكود
تعنى أننا سنستخدم فصيلة Class من .System المسمى Namespace	using System
يستخدم في تنظيم الكود وهو يحتوي بداخله على الفصائل Classes وأيضا كرى	namespace
الأقواس المعقوفة "أقواس المجموعة" تستخدم في بلوك من الكود.	{}
حاوية للبيانات والطرق Methods والدوال	
Functions ويجب أن تكون كل أسطر الكود داخل الفئة أو الفصيلة Class.	class

التفسير	الكود	
هي الطريقة Method الرئيسية داخل التطبيقات		
من النوع CONSOL App والتي تبدأ في التنفيذ		
بمجرد Start للكود لذا يجب كتابة أكواد مشروعك	Main	
بداخل أقواسها المعقوفة {} أو على الأقل		
استدعاء الكود بداخلها لكى يتم تنفيذه		
System Namespace هي فصيلة من الـ	Console	
لشاشة موجة الأوامر "الشاشة السمراء"		
تستخدم في طباعة القيم على سطر جديد	WriteLine()	
ت ي طباعة قيم على نفس السطر	Write()	

: WriteLine() و Write() مثال للفرق بين

Console. WriteLine("Hello World!");

Console.WriteLine("I will print on a new line.");

Console. Write("Hello World! ");

Console. Write("I will print on the same line.");

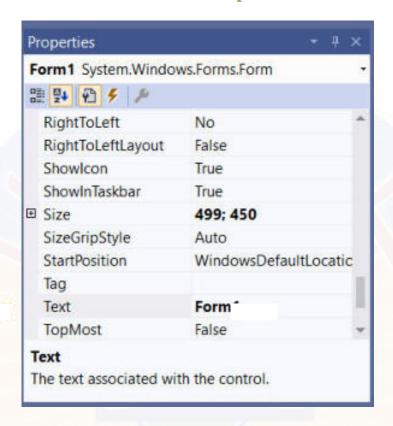
النتيجة

Hello World!

I will print on a new line.

Hello World! I will print on the same line.

- نافذة الخصائص Properties -



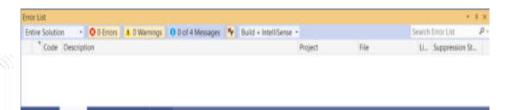
تستخدم في ضبط الخصائص المختلفة لعناصر المشروع من نماذج وأدوات تحكم.

- نافذة الإختبار Test:



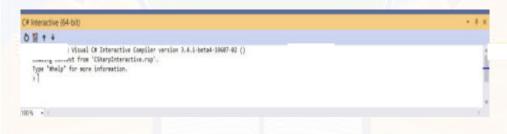
والتي تعطيك نتيجة فحص البرنامج أثناء بناء التطبيق Build وهِل توجد أخطاء أم لا.

- نافذة الإختبار Error List:



والتي تعطيك نتيجة بعدد التحذيرات والأخطاء البرمجية في الكود مع توضيح سبب الخطأ والسطر البرمجي المتسبب في ذلك الخطأ.

- نافذة التفاعل C# Interactive Compiler -



المترجم التفاعلى والذي يعطيك المزيد من التحكم ويمكنك معرفة جميع المعلومات عنه من خلال كتابة Help في النافذة.

- نافذة الخرج Output:

```
Show nated from Debug

**Consort remainification(.ese* (CLE vt.E.780)); stransformagelization(.ese) Lased *C. school-directly latteressy latt. Mil. System Design violence of substantial control of substanti
```

يظهر بها رسائل التقدم والتحميل أثناء الترجمة والتنفيذ للكود.

لمن يرغب في الاطلاع على معلومات إضافية يمكنه من خلال موقع شركة مايكروسوفت MSDN التعرف على باقى النوافذ وتفاصيل أكثر:

https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019

© خامسا تشریح ملفات net کامسا

أنوع التطبيقات الأكثر شيوعا والمنفذة بلغة #2:

- تطبيقات موجه الأوامر CONSOL Application:

والتي تعمل من الشاشة السوداء أو موجه الأوامر المشابهة لنظام التشغيل Dos قديما والتي لا تحتوي على واجهه رسوميه.

- تطبیقات سطح المکتب Desktop Application -

تطبيقات تعمل على أنظمة التشغيل Windows المختلفة ويمكن إنشاء هذا النوع بطريقتين إما من خلال Windows Form App أو "Windows Foundation Application" WFP

- تطبيقات الوبب Web Application -

"Active Server ASP.NET وتستخدم في هذا الغرض Web Site والمحالات المحالة العام "Pages" لبناء تطبيقات المحال المحال

- تطبیقات متجر وبندوز Windows Store Application

بدءا من نظام التشغيل Windows 8, 8.1, 10, 11 اعتمدت متجر خاص بتطبيقاتها مثل متجر جوجل وآبل ويمكنك برمجة تطبيقات شكلذلك المتجر باستخدام #C.

- تطبيقات الخدمات WFC Services -

وتستخدم في بناء تطبيقات الخدمات الموزعة تقنية وتستخدم في بناء تطبيقات الخدمات يمكن Foundation Communication من خلال الشبكات المحلية والإنترنت.

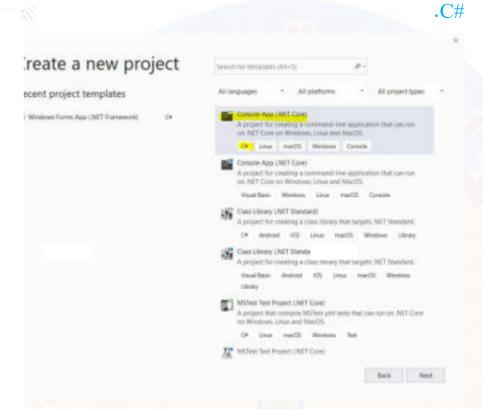
وهناك العديد من التطبيقات الأخرى، ولكن كانت تلك أكثرها

يقتصر هذا الكتاب على تطبيقات موجه الأوامر CONSOL App يقتصر هذا الكتاب على تطبيقات موجه الأوامر Windows Form وتطبيقات سطح المكتب Desktop App

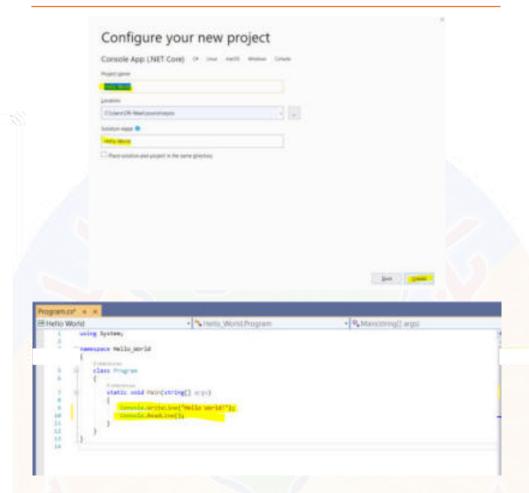
Hello World اول تطبيق

التطبيق الشهير بطباعة جملة Hello World ستجده دائما في جميات التناول البرمجة سواء بلغة #C غيرها من اللغات بمجموعة Visual Studio كبداية لاستخدام اللغة، ولكي نقوم بتكويد البرنامج سنستخدم البرمجة من خلال موجه الأوامر أو Console والتي يطلق عليها النافذة السوداء

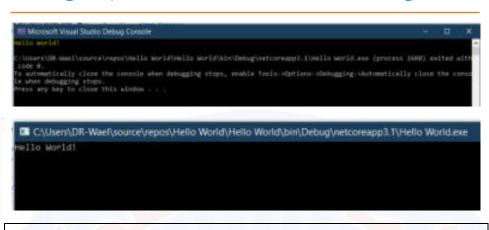
وليس تطبيقات سطح المكتب وللقيام بذلك نختار إنشاء تطبيق جديد ونحدد النوع Console App مع التأكد من لغة التطوير



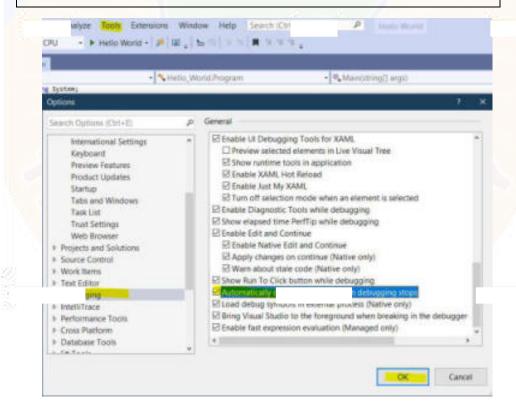
ثم نقوم بتحديد اسم للتطبيق Hello World ومكان الحفظ على الجهاز



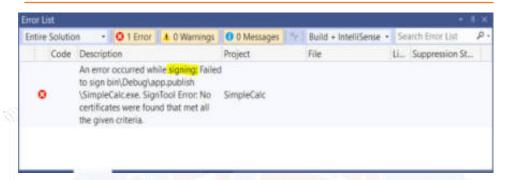
في المثال السابق استخدمنا الكائن Console وتم استخدام الـ WriteLine طي الشاشة وفي حالة كنا قد نشطنا إغلاق الشاشة عقب انتهاء التشغيل فلن تتمكن من مشاهدة عقب انتهاء التشغيل فلن تتمكن من مشاهدة يجة ويلزم في هذه لل التنفيذ من خلال كتابة وبمجرد هدفه ليس قراءة السطر وإنما التعطيل لمشاهدة القيم المطبوعة وبمجرد كتابة اي حرف يختفي.



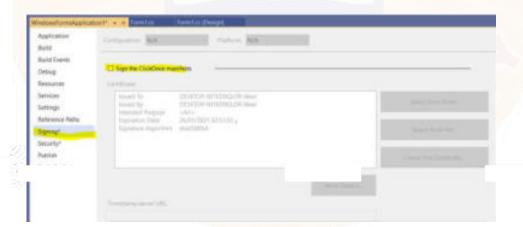
لتفعيل إغلاق نافذة Console عقب انتهاء الـ Debug نتبع الخطوات (Tools -Options -Debugging- Automatically التالية Close the console...)



عند التنفيذ في حالة ظهور خطأ يخص الـ Certification ، Signing



ويمكن حل المشكلة إما بتحميل شهادة Certification غير منتهية الصلاحية أو ببساطه تعطيل هذا الخيار وذلك من خلال نافذة متصفح الحل Solution Explorer والضغط بالفارة بالزر الأيمن على متصفح الحل Project و Signing ثم المشروع Project ثم الشهادة كما في الشكل التالي أو تحميل شهادة جديده.



تشريح المشروع وشرح ملفاته:

يتم تجميع ملفات العمل البرمجي تحت مسمى المشروع وغالبا ويمكن أن يجتمع أكثر من مشروع تحت مسمى الحل Solution وغالبا ما تكون تلك المشاريع متكاملة لضمها تحت حل واحد كمثال إحدى الشركات يتم برمجة مشروع للعمل على سطح المكتب ومشروع أخر عبارة عن موقع إلكتروني لنفس الشركة ومشروع ثالث كتطبيق للعمل من الموبايل فهذه المشاريع الثلاثة يمكن دمجها داخل حل Solution باسم الشركة. فالمشاريع الثلاثة الحاوية لجميع المشاريع الهشروع على المشاريع ا

ومن خلال نافذة متصفح الحل Solution Explorer يتم إضافة مشروع جديد بالضغط على اسم الـ Solution بالزر الأيمن ثم من القائمة الفرعية Add=>New Project ، سيتم إضافة مشروع جديد مع انية تغيير نوع الـ لـ وتغيير لغة برمجته. وكما

أوضحنا أن نافذة متصفح الحل Solution Explorer تشمل كل كافة عناصر ومكونات المشاريع ويعد ملف Program.cs أهم تلك المكونات

عند إنشاء مشروع جديد فيتم حفظ ملفاتِ المشروع على الـHDD في مسار محدد والذى تقوم باختياره أثناء إنشاء المشروع - وبفضل الابتعاد عن سطح المكتب Desktop ومستنداتي My Documents والبعد عن محرك الأقراص :C حيث أنه عرضه للحذف في حالة حدثت مشكلة لنظام التشغيل وتطلب الأمر عمل Format - داخل مكان الحفظ الذي قمت باختياره سيتم تلقائيا إنشاء مجلدا باسم المشروع يحتوي على لعديد من الملفات بامت ختلفة. فأي امتداد نقوم باختياره لـ كامل المشروع بكل محتوياته؟ لكي تقوم بفتح الملف المناسب من قائمة File => Open => Project/Solution، مجلد المشروع وتأكد من فتح الملف بامتداد sln. وهو ملف الحل Solution الذي يحتوى بداخله جميع المشاريع لهذا الحل (أو على الأقل مشروع د في حال لم ت متعددة) وبذلك سيظهر لك

بملفاته الرئيسية. ويمكن الاطلاع على هيكل الملف في حالة فتحه باى محرر نصوص ولكن يجب الحذر وعدم التعديل في محتواه، وأيضا نفس الشيء بالنسبة لملف المشروع Project الذي هو جزء من الحل ويكون

امتداده csproj وهو الملف الذي يحتوي على كافة إعدادات المشروع ويوجد بداخل مجلد فرعى من مجلد الحل Solution وباسم كل مشروع من المشاريع الفرعية.

مجلد باسم Bin وهو اختصار لكلمة Binary ويوجد به جميع مخرجات المشروع بداخل مجلد فرعى آخر باسم Debug ، ويظهر هذا المجلد عند قيامك بتنفيذ التطبيق من داخل Visual Studio ولو مره واحدة، وتجد من ضمن تلك الملفات نسخه تنفيذية من تطبيقك الذى مت ببرمجته بامتداد E مل دون الحاجه لتثبيت وقد يتطل أن يحتوى الجهاز على Visual Studio ولكن في حالة رغبتك في توزيع البرنامج بدون اللغة كتطبيق احترافي يلزمك تحويله الى نسخه جاهزة للتوزيع البرنامج بدون اللغة كتطبيق احترافي يلزمك تحويله الى نسخه جاهزة للتوزيع . Release

Solution	الحل البرمجي وهو أعلى وأكبر حاوية للمشروع، ويمكن أن يتكون من أكثر من Project
Project	المشروع الذي نقوم بتطويره ويكون تحت Solution واحد
Longole	شاشة سوداء تطبع أسطر متتالية وتتميز بسهولتها وقوة أداءها وتشبه نظام التشغيل DOS لمن شاهده
Program.cs	هو الملف الرئيسي الذي نقوم بكتابة أسطرنا البرمجية الأولى عليه ويحتوي على الدالة الرئيسية Main

نافذة تقوم بعرض جميع مكونات الحل البرمجي، من مشاريع ومكتبات ودوال	Solution Explorer
هو امتداد الملف الخاص بالحل Solution	SLN
عملية استكشاف الأخطاء	Debug
نظام مخرجات ومعالجة يحتوي على معلومات إضافية من هدفها استكشاف الأخطاء وإعطاء تقارير	Debug Mode
نظام مخرجات نهائي يتم إرساله للزبائن مباشرة	Release Mode
ملف تنفيذي Executable File يعمل على نظام ويندوز	EXE

مشروع إنشاء حاسبة

أولا اختيار نوع التطبيق وضبط الأدوات والخصائص الأساسية

القيمة	الخاصية	الأداة
"عمليات حسابية لرقمين"	Text	Form1
btnNumAdd, btnSubt, btnMulti, btnDiv, btnTxtAdd	Name	٥ أدوات
+, -, *, /, &&	Text	Button
txtNo1, txtNo2, txtOut	Name	٣ أدوات
	xt	TextBo
الرقم الأول، الرقم الثان	Text	Labelوات
Windows Form App	نوع التطبيق	



- كود الضرب:

```
27
              private void BtnPulti_Click(object sender, EventArgs e)
2.8
29
                  double r:
30
                  r = Convert.ToDouble(txtNo1.Text) * Convert.ToDouble(txtNo2.Text);
31
                  txtOut.Text = Convert.ToString(r);
                                                                - كود القسمة:
              private void BtnDiv_Click(object sender, EventArgs e)
                  double n;
                  r = Convert.ToDouble(txtNo1.Text) / Convert.ToDouble(txtNo2.Text);
                  txtOut.Text - Convert.ToString(r);
                                                                - كود الجمع ال
                private void BtnTxtAdd_Click(object sender, EventArgs e)
                    txtOut.Text = txtNo1.Text+ txtNo2.Text;
```





الفصل الثالث المتغيرات والثوابت

Constants الثوابت

هو مكان في الذاكرة يتم الإعلان عنه داخل اللغة لتخزين قيمة من نوع محدد يتم أيضا تحديده عند الإعلان عن الثابت مع تحديد قيمة الثابت في الإعلان. ولا يجوز محاولة إعادة تعيين قيمة أخرى داخل البرنامج أي أن الثابت قيمته لا تتغير بدءا من الإعلان عنه حتى انتهاء البرنامج ومن هنا جاءت التسمية "ثابت Constant"

للإعلان عن الثواب دم الصورة التالية:

Const Datatype const_name=const_value;

حيث أن Const: كلمة محجوزة للدلالة على الإعلان عن ثابت.

Int, مثل البيانات مثل: Datatype

Double, String,etc

const_name

const_value: قيمة الثابت

مثال للإعلان عن ثابت باسم K من النوع integer وإعطاؤه قيمة

الفصل الثالث المتغيرات والثوابت

Const Int k=50;

Variables المتغيرات

هو مكان في الذاكرة يتم الإعلان عنه داخل اللغة لتخزين قيمة من نوع محدد يتم تحديده عند الإعلان عن المتغير (ويمكن تحديد قيمة الثابت في الإعلان أو تأجيل إعطاء المتغير قيمة داخل البرنامج). ويختلف المتغير عن الثابت في أنه يجوز إعادة تعيين قيم أخرى داخل البرنامج أي أن الثابت قيمته لا تهاء أمن الإعلان عنه حتى انتهاء البينما المتغير يجوز إعادة تعيين قيم له أكثر من مرة ومن هنا جاءت التسمية "متغير عجوز إعادة تعيين قيم له أكثر من مرة ومن هنا جاءت التسمية "متغير عتون الإعلان عنه حتى التسمية "متغير عمن هنا جاءت

للإعلان عن المتغيرات نستخدم الصورة التالية:

Datatype var_name=var_value;

Int, Double, String... تستبدل بأحد أنواع البيانات :Datatype

:var_na: اسم ا

var_value: قيمة المتغير

مثال للإعلان عن متغير k من النوع integer وإعطاؤه قيمة

٥٠ أثناء الإعلان.

الفصل الثالث المتغيرات والثوابت

Int k=50;

مثال للإعلان عن متغير k من النوع integer وعدم إعطاؤه قيمة

أثناء الاعلان وتحديد قيمة ٥٠ بعد الاعلان.

Int k; k=50;

الإعلان المتعدد للمتغيرات دفعه واحده من نفس نوع البيانات

Datatype var1 name=var1 value,

2 name=var2 value, var e=var3 value;

Double, String... ع البيانات :Datatyp

var1 name: اسم المتغير الاول

var1 value: قيمة المتغير الأول

var2 name: اسم المتغير الثاني

var2 value: قيمة المتغير الثاني

var3 name: اسم المتغير الثالث

var3 value: قيمة المتغير الثالث

كذا لأى عدد من المعلان عنهم فقط بد

قيم وتأجيل إعطاء القيم لداخل البرنامج في بعد.

مثال

int
$$x = 5$$
, $y = 6$, $z = 50$;
Console.WriteLine $(x + y + z)$;

Data Types أنواع البيانات

المدى	الوصف	المساحة	النوع
0 to 255	رقم صحيح +	8-bit	byte
-128 to 127	رقم صحيح + أو -	8-bit	sbyte
-32,768 to 32,767	رقم صحيح + أو -	16-bit	short
535	یح +	16-bit	ushort
-2,147,483,648 to 2,147,483,647	رقم صحيح + أو -	32-bit	int
0 to 4,294,967,295	رقم صحيح +	32-bit	uint
-9,223,372,036,854,775,808 : 9,223,372,036,854,775,807	رقم صحيح + أو -	64-bit	long
0 to 18,446,744,073,709,551,615	رقم صحيح +	64-bit	ulong
-3.402823e38 : 3.402823e38	عشرى مفرد الدقة	32-bit	float
-1.79769313486232e308 to 1.79769313486232e308	عشرى مزدوج الدقة	64-bit	double
x 10e-28 to 8	المالية	128-bit	deci
Any valid character, e.g. a,*, \x0058 (hex), or\u0058 (Unicode)	نصوص يونيكود مفرده	16-bit	char
True or False	منطقى	8-bit	bool
		كائن	object

المدى	الوصف	المساحة	النوع
	سلسلة حروف يونيكود		string
0:00:00am 1/1/01 to 11:59:59pm 12/31/9999		تاريخ وقت	DateTime

أنواع البيانات باللون الأزرق الأكثر شيوعا والأنواع باللون الرمادى للقراءة ويمكنك تجربتها عند احتياجك لها

🗸 مجال الإعلان عن المتغيرات والثوابت:

ويقصد به حياة المتغير أو الثابت أي المدى الذي يتم التعامل

معه داخله وهناك ثلاث أنواع رئيسية:

- المتغير المحلى Local أو على مستوى الإجراء الفرعي.
 - على مستوى النموذج أو الـ Class.
 - على مستوى المشروع Global.

• أولا الإعلان على المستوى المحلى Local:

يقصد بمجال الإعلان المحلى أي داخل الإجراء الفرعى أو الحدث

الدالة وهنا تكون المنابت داخل الإجرا

فقط أي لا يمكن أن يتم قراءته خارج هذا الإجراء وعند محاولة ذلك فإن اللغة تظهر خطأ أن المتغير أو الثابت لم يتم الإعلان عنه.

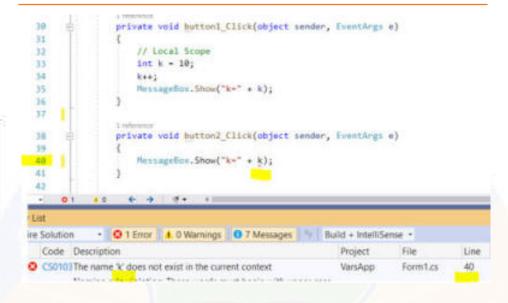
```
private void button1_Click(object sender, EventArgs e)
{
    // Local Scope
    int k = 10;
    k++;
    MessageBox.Show("k=" + k);
}
```

تم الإعلان عن المتغير k وإعطاءه قيمة 10 ثم إضافة رقم"١" إلى قيمة المتغير ليصبح ١١.

+K تعنى إضافة ١ إلى له السابقة وكانت تكتب قديما 1+

MessageBox.Show يقوم بإظهار رسالة حوارية للمستخدم ونص تلك الرسالة هو قيمة المتغير المكتوب بين الأقواس.

لاحظ عند تنفيذ الكود والضغط على الزر المكتوب فيه الكود ستظهر القيمة ١١ نظرا لأننا داخل نفس الحدث فبالتالى المتغير ما زال موجود ولكن عند محاولة استدعاء المتغير داخل زر أخر سيظهر خطأ انك لم تقم بالإعلان عن المتغير من الأساس.



على الرغم من أنه تم الإعلان عن المتغير k داخل الزر button ولكن عند م عة قيمته داخل button في الساعطى خطأ كما هو واضح من الشكل السابق أن المتغير غير موجود أي انه لم يتم الإعلان عنه.

• ثانيا الإعلان على مستوى النموذج أو Class:

يتم الإعلان عن هذا النوع في بداية النموذج أو الـ Class ويكون مرئى ومتاح سواء للقراءة أو تعديل القيمة داخل جميع الإجراءات الفرعية دوال والأحداث الم نفس النموذج أو الـ Class.

وكما في الكود التالى تم الإعلان عن المتغير k داخل النموذج وقبل الدخول في الأكواد الفرعية (حدث، دالة، إجراء فرعى) وبالتالي سيكون متاح للقراءة والتعديل داخل النموذج المعلن عنه بداخله

وللتحقق من مستوى الإعلان عن المتغير سنقوم بإضافة زرارى أمر وتعديل قيمة المتغير بكل منهما وإظهار الرقم ستجد أنه يزيد بمقدار "١" في الزر الأول عند كل ضغطه والزر الثاني يظل متذكر أخر قيمة.

```
private void button1_Click(object sender, EventArgs e)

At+;
MessageBox.Show("k=" + k);

limits void button2_Click(object sender, EventArgs e)

{
MessageBox.Show("k=" + k);
}
```

عند الضغط على الزر الأول سيتم إضافة ١ للمتغير وعند النقر على الزر الثاني سيظهر آخر قيمة للمتغير

شروع أو Project:

قالثا الإعلان على

يتم الإعلان عن هذا النوع في Class البرنامج الرئيسى Program.cs ويكون مرئى ومتاح سواء للقراءة أو تعديل القيمة داخل جميع الإجراءات الفرعية والدوال والأحداث المكتوبة داخل جميع نماذج المشروع أو الـ Classes.

```
# Variable | Francia | Promition | Promiti
```

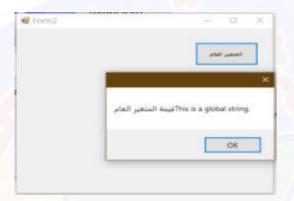
ولكى يتم التعامل مع هذا المتغير من داخل النماذج المختلفة أو الدين التعامل مع هذا المتغير من داخل النماذج المختلفة أو Classes الدين المتعامل يتم استدعاءه بالصيغة السابق للتعامل مع المتغير globalstring لكى يتم الوصول للمتغير وقراءة أو تعديل قيمته.

وفيما يلى إعادة يمة جديده للمتغير العام داخل نم عرضها ثم الدخول لنم وعرض قيمة المتغير.

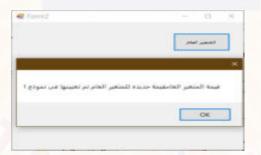
استدعاء قيمة المتغير في النموذج الثاني

```
Education | Programus | Formles [Design] | Programus | P
```

في حالة النقر على زر إظهار قيمة المتغير العام من النموذج الثانى أولا فتظهر القيمة قبل التعديل



في حالة النقر على زر إظهار قيمة المتغير العام من النموذج الثانى بعد الضغط على زر من النموذج الأول والذى تم في قيمة المتغير فتظهر رسالة بها قيمة المتغير بعد التعديل.



يتم إضافة نم مشروع بالضغط على قائمة ومنها Add Form. ولكى يظهر النموذج عند التشغيل يجب استدعاء النموذج ويتم ذلك من خلال الكود التالى ويمكن كتابته في زر أمر:

الفصل الثالث

```
private void BtnFrm2_Click(object sender, EventArgs e)
{
   Form2 frm=new Form2();
   frm.Show();
}
```

التحويل بين أنواع البيانات:

لا يمكن بأى حال من الأحوال أن تقوم داخل البرنامج بتعيين قيمة Assign لمتغير أو خاصية من نوع مختلف حتى وإن كانت تحتوى قيم ظاهريا مقبولة كمثال لا يمكن تعيين قيمة "١٢" لمتغير من النوع علمات حيث انه وعلى الرغم من أن ١٢ رقم إلا أن وجودها داخل علامات لتنصيص فقد حولها إلى ومن هنا يجب أن تراعى دائما تلك في أي تعيين لقيمة يجب أن يكون الطرفين من نفس نوع البيانات وفى حالة كان هناك اختلاف فى النوع يجب تحويله الى النوع المناسب.

ويتم تحويل نوع البيانات عندما نرغب بتعيين قيمة نوع بيانات إلى نوع آخر وهناك نوعان من التحويل:

التحويل الضمني Implicit Casting (التلقائي)

ويتم عند التح المعر إلى نوع أكبر وفقا التالي:

char -> int -> long -> float -> double

مثال:

```
int aInt = 10;
double aDouble = aInt; // التحويل الضمنى من
double
Console.WriteLine(aDouble);
```

التحويل الصريح Explicit Casting (اليدوي)

عند التحويل من نوع بيانات بحجم أكبر إلى نوع بيانات بحجم أصغر ويتم بكتابة النوع الجديد المراد التحويل إليه بين قوسين () وذلك في حالة كان النوع المراد التحويل إليه أقل حجما كما في التسلسل التالى:

double -> float -> long -> char

مثال:

```
double aDouble = 3.75;
int aInt = (int) aDouble; // التحويل الليدوى من الأعلى double; // التحويل اللؤقل
double التحويل الأقل int
Console.WriteLine(aInt);
```

```
يقصد بالحجم الأكبر والأقل في التحويل الضمنى أو اليدوى لنوع انات المساحة التخ ع والمذكورة في أول الفصل اع البيانات.
```

طرق تحويل أنواع البيانات

لنفرض أن لديك متغير ;int a=10 وترغب في عرضه في مربع نص فستواجه خطأ أن استخدمت الصيغة ;textBox1.Text=a حيث أن الطرفين غير متساويين احدهما رقم والآخر نص.

الطريقة الأولى بإضافة نص على المتغير: عند إضافة نص إلى متغير رقمى فيكون الناتج قيمة نصية حتى وإن كانت الإضافة مجرد علامة تنصيص وغلقها ""

textBox1.Text = a + "";

الطريقة الثانية var.ToType: وفيها يستخدم اسم الم نقطة والنوع المراد التحويل إلية وفى حالة المثال السابق مطلوب تحويل المتغير a إلى نص فنكتب الكود كالتالى:

textBox1.Text = a.ToString();

الطريقة الثالثة (Convert.ToString: وفيها يتم تحويل المتغير الذي يتم كتابته داخل الأقواس إلى نص.

xtBox1.Text = Convert.T

الطربقة الرابعة (OldDataType) الطربقة الرابعة

حيث يتم تحويل القيمة بين القوسين إلى نوع البيانات المكتوب قبل Parse مع ملاحظة أن تلك الطريقة تصلح للتحويل إلى أي نوع بيانات ماعدا التحويل إلى نص.

الفصل الثالث المتغيرات والثوابت

a = double.Parse(textBox1.Text);

مثال مجمع لإستخدام الثوابت والمتغيرات والتحويل:

داخل الأكواد ستجد أسطر تم تحويلها إلى ملاحظه قم بتجربة الغاء الملاحظات بما يتناسب مع طبيعة مستوى الإعلان أو نوع التحويل المطلوب وتحويل الكود النشط إلى ملاحظه لكى تقوم بتجربة جميع ما سبق شرحة في الفصل.

أولا الأدوات وضبط الخصائص الأولية:

القيمة		الخاصية	الأداة
	VarsA	ext	Form1,2
btnAddVar, btn btnConvToStr, l	LastVal, otnConvToDubl,	Name	
btnClose, btnFrnbtnGenVar,btn	m2, btnGlobalVal Close		۷ أدوات Button و ۲
Exit ،۱۰ المتغير	زيادة المتغير، اخر قيمة للا لنص، تحويل نص لرقم * العام، نموذج٢، المتغير ال	Text	للنموذج الثاني
mm .	Unit	Text	TextBox
indovv	Form Ann		tation !!

indows Form App

(نموذجين ومجموعة من أزرار الأوامر ومربع نص كما في الصور التالية مع تغيير أسماء الأزرار والنص بما يتناسب مع كل زر منهم)



العود الـ Program.cs

للإعلان عن المتغير العام على مستوى المشروع

```
static class Program
{
   //Global Var
   public static string globalString="This is a global string.";
```

أكواد النموذج form1 بأدواته:

```
public partial class Form1 : Form
    // Class Scope
   int k = 10;
    1 reference
    public Form1() ...
    private void Formi_Load(object sender, EventArgs e) ...
   private void BtnAddVar_Click(object sender, EventArgs e)
        // Local Scope
        //int k - 10;
        k++;
        MessageBox.Show("k-" + k);
    1 reference
    private void BtnlastVal_Click(object sender, EventArgs e)
        NessageBox. Show("k=" + k);
    private void BtmConvToStr_Click(object sender, EventArgs e)
       int a - 200;
        terror Cod
        ************* a;
        //Cod 1 convert using var.ToType()
        //textBox1.Text = a.ToString();
       //cod 2 convert using add text "" only used for text
        //textBox1.Text = a+"";
        //Cod 3 convert using Convert.Totype(oldtype)
        textBox1.Text = Convert.ToString(a);
```

الفصل الثالث

أكواد النموذج form2 أدواته:

```
| Program.cs | Form1.cs | Form1.c
```





<u>Operator المؤثرات</u>

لا يخلو برنامج تقريبا من استخدام مؤثرات المقارنة أو المؤثرات الحسابية كالمقارنة أو إعطاء قيمة لمتغير الخ من المؤثرات المختلفة لذلك وجب التنبيه على أهمية استخدام تلك المؤثرات.

:Assignment Operator مؤثرات تخصيص القيم

وتستخدم عند إعطاء قيم للمتغيرات وفيما يلى أكثر مؤثرات تخصيص القيم للمتغيرات شيوعا:

مكافئ لـ .	مثال	الم	المؤثر
x = 5		التخصيص المباشر للقيمة	=
$\mathbf{x} = \mathbf{x} + 3$	x += 3	التخصيص بزيادة قيمة	+=
$\mathbf{x} = \mathbf{x} - 3$	x -= 3	التخصيص بإنقاص قيمة	-=
$\mathbf{x} = \mathbf{x} * 3$	x *= 3	التخصيص بضرب قيمة	*=
x = x / 3	x /= 3	التخصيص بالقسمة على	/=
x = x % 3	x %= 3	التخصيص بباقى القسمة	%=

مثال على تخصيص القيم للمتغيرات:

```
using System;
     Enamespace Operators
          //-----Operators App Example 0 wael E. Zaweel 2022-----//
5
7
          class Program
- 5
9
              static woid Main(string[] args)
10
11
12
13
                      1/ (
14
15
                     int a - 5;
15
                      a +- 9;
                      Console, WriteLine(a); // output: 14
17
18
19
                     a -- 4:
20
                      Console.WriteLine(a); // output: 18
                      a *= 2;
23
                     Console.WriteLine(a); // output: 28
                     a /= 4;
                      Console, WriteLine(a); // output: 5
27
28
                      a %+ 3;
                      Console, WriteLine(a); // output: 2
29
```

المؤثرات الحسابية Arithmetic Operators:

وتستخدم عند إجراء العمليات الحسابية بين الأرقام وكذلك بعضها يستخدم في زيادة أو إنقاص قيمة متغير كما هو الحال في دليل الحلقات رارية في الفصل ا

مثال	الوصف	المؤثر
a + b	عملية الجمع (أو إعطاء إشارة موجبه للرقم)	+
a - b	عملية الطرح (أو إعطاء إشارة سالبة للرقم)	1
a * b	عملية الضرب	*
a / b	عملية القسمة	/
a % b	$1 = \frac{5}{2}$ تعطى الباقى من عملية القسمة	%
a++	زيادة المتغير بمقدار ١	++
	نقصان المتغير بمقدار ١	

مثال:

```
using System;
      ≡namespace Operators
            //-----Operators App Example 0 wael E. Zaweel 2022-----//
 5
 6
 7
 ñ
            class Program
9
                static void Main(string[] arms)
18
11
12
31
32
                          المؤثرات الحمايية//
34
                         Console.WriteLine(5 + 4); // output: 9
                         Console.WriteLine(47 - 3); // output: 44
Console.WriteLine(13 / 5); // output: 2
35
36
                         Console.WriteLine(5 % 4); // output: 1
37
38
39
40
56
57
58
```

مؤثرات المقارنة Comparison Operators:

يجب استخدام مؤثرات المقارنة التالية عند اختبار القيم كما في حالات الشرط والتي سيتم شرحها في الفصل التالي

الاستخدام	مؤثر المقارنة
أقل من	a < b
أقل من أو أكبر	a <= b
أكبر من	a > b
أكبر من أو يساوى	a >= b
یساوی	a === b
لا يساوى	!= b

مثال:

```
using System;
      Enamespace Operators
             //-----Operators App Example @ wael E. Zaweel 2022-----//
 6
            Onference.
            class Program
                 static void Main(string[] args)
10
12
418
                          مؤثرات العقارنة//
41
42
                          int x + 5;
43
                          int y - 3;
44
                          Console.WriteLine(x >= y); // output True
                          Console.WriteLine(x == y); // output False
Console.WriteLine(x != y); // output True
45
46
47
48
56
57
58
```

المؤثرات المنطقية Logical Operators

وتستخدم في الربط بين المتغيرات والقيم المختلفة (مع، أو، ليس)

مثال	الوصف	المعنى	المؤثر
a<3 && b<6	تعطى true في حالة تحقق القيمتين	and	&&
a<3 b<9	تعطى true في حالة تحقق قيمة على الأقل من القيمتين وكذلك تحقق الاثنين	or	Ш
!(a<3 && b<9)	تعكس القيمة المنطقية لو true تتحول لــ false والمكن	not	I

مثال:

48 49

50) 51

57

الفصيلة Math وإستخداماتها الرباضية:

تستخدم الفصيلة Class التي تسمى Math في إجراء العمليات الحسابية الأساسية عن طريق مجموعة من الطرق Methods لكل طريقة منهم وظيفة تقوم بها على الأرقام والصورة العامة لتنفيذها.

Math.Method(numeric vars);

Method: الطريقة أو الأمر الذي سيتم تنفيذه numeric_vars: المتغيرات الرقمية (أو قيم رقمية مباشرة) التي سيتم التطبيق عليها.

وفيما يلى توضيح لأهم تلك الطرق Methods واستخدامها:

مثال	الوصف	الطريقة Method
Math.Max(5, 10)	القيمة العظمى	Max(x,y)
Math.Min(5, 10)	القيمة الصغرى	Min(x,y)
Math.Sqrt(25)	الجذر التربيعي	Sqrt(x)
s(-5)	طلقة "الموجبة"	Abs(x)
Math.Round(5.65)	تقريب الأرقام العشرية	Round(x)

مثال على استخدام فصيلة Math وطرقها المختلفة:

```
saing System;
      Enamespace Operators
                 ---Openetors App Example 0 wael E. Zaweel 2022-----//
           Class Program
1
18
               static void Main(string[] args)
11
12
21
39
44
56
57
                       //applead objlical alice Math
                       Communic.NriteLine("Max No (5 & 10) is: " + Math.Max(5, 10));
99
                                                                                      //output Hax No (5 & 18) 1s: 18
                       Consuls.Nritmine("Min No (5 & 10) is: " + Math.Min(5, 10));
68
                                                                                      //output Min No (5 & 10) is: 5
                       Communication ("The Squart Of 25 is: " * Math. Sqrt(25));
                                                                                      //output The Squert Of 25 is: 5
                      CommunicateLine["The Absolute Of -5 is: " * Math.Abs(-5));
62
                                                                                      //output The Absolute OF -5 is: 5
                       Console.WriteLine("The Round Of 5.65 is: " + Math.Round(5.56)); //output The Round Of 5.65 is: 6
63
55
```

String الطرق Methods النصية المتغيرات النصية

عند الإعلان عن متغير نصى من النوع String فإنه يتمتع بالعديد من الطرق Methods التي تسهل عليك إجراء العديد من العمليات على قيم هذا المتغير.

للإعلان عن متغير من النوع String نستخدم الصيغة التالية:

string var name = value;

حيث أن:

string: كلمة محجوزة للإعلان عن نوع المتغير النصى.

var name: اسم المتغير

var_value: قيمة نصية للمتغير وتكتب بين علامات تنصيص.

وفيما يلى أهم تلك الطرق وأكثرها استخداما:

الوصف	الطريقة Method
تسمى Concatenate وهي تضم القيم النصية "ولا يتم إضافة مسافات بين تلك القيم فيجب مراعاة ذلك عند استخدامها بإضافة المسافات	string.Concat(x,y)
یدویا" تسمی interpolation وهی بعکس عملیة	
Concatenate حيث يتم ضم القيم النصية المتغيرات المذكورة داخل أقواس المجموعة {} مسافة بين كل قيمتين	=\$ "{} {} {}"
يرجع بالحرف رقم "no" من المتغير النصى var مع العلم أن البداية بالرقم صفر	var[no]
تعود برقم يدل على ترتيب الحرف "chr" داخل المتغير النصى.	var.IndexOf(chr)
تقص مجموعة من الحروف من المتغير var بدءا من الحرف رقم no وتعود بتلك السلسلة	var.Substring(no)
تحويل المتغير النصى إلى حروف كبيره Capital	var.ToUpper()
تحويل المتغير النصى إلى حروف صغيرة Small	var.ToLower()

مثال على استخدام الطرق المختلفة للمتغير النصى من النوع String وتأثيرها على النص:

العلامات الخاصة:

في بعض الأحيان يتطلب الأمر كتابة بعض العلامات الخاصة داخل النص وتلك العلامات لا تقبلها لغة #C مباشرة مثل علامة صيص وإدراج سط مسافة Tab "تكافئ ٤ مساف وغيرها من العلامات التي لا تقبلها اللغة عند كتابتها تعطيك خطأ لذلك يجب التحايل على هذا الخطأ باستخدام رموز أخرى تفهمها #C وتقوم بترجمتها إلى العلامة التي ترغب في كتابتها من الأساس. وفيما يلى اشهرها:

النتيجة	الوصف	البديل
•	تستخدم لكتابة علامة تنصيص مفردة	\'
"	تستخدم لكتابة علامة تنصيص مزدوجة	\"
1	تستخدم لكتابة شرطة عكسية	\\
تكافى الضغط على مفتاح Enter		\ n
تكافئ الضغط على مفتاح Tab		\t
	تكافئ الضغط على مفتاح Backspace	\ b

مثال على استخدام علامات بديلة للتغلب على مشكلة عدم كتابة عض الحروف والعلاما على عض الحروف والعلاما على على عض الحروف العلاما على عض الحروف العلاما على عض الحروف العلاما على على العلاما على العلام على العلى العل





و جملة الشرط if:

وتستخدم في التفرع المشروط داخل المشروع بمعنى تختبر شرط محدد وبناء عليه يتم تنفيذ كود محدد في حالة تحقق الشرط وكود آخر في حالة عدم تحقق الشرط.

يجب ملاحظة أن if تكتب بحروف صغيرة في حالة كتابتها بحروف كابيتال (IF أو IF) يعطى خطأ.

الحالة الأولى لجملة if وفيها نكتفى بحالة التحقق فقط.

```
dition)
{

الكود الذي سيتم تنفيذه عند التحقق //
}
```

مثال:

30)

Console.WriteLine("60 is greater");

if else الثانية جملة

if (condition)

```
الكود الذي سيتم تنفيذه عند تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي سيتم تنفيذه عند عدم تحقق الشرط // الكود الذي الذي الكود الذي الذي الكود الكود الذي الكود الذي الكود ال
```

```
مثال:
```

```
if (60 > 30)
{
    sole.WriteLine("60 is gre
}
else
{
    Console.WriteLine("30 is greater");
}
```

:if elseif الثالثة

الصورة العامة:

```
if (condition I)
{

بلوك الكود الذي سيتم تنفيذه في حالة تحقق الشرط //
}
else if (condition 2)
```

```
إلوك الكود الذي سيتم تنفيذه في حال عدم تحقق الشرط الأول //
وتحقق الشرط الثاني

else
{
// بلوك الكود الذي سيتم تنفيذه في حالة عدم تحقق كل الشروط السابقة //
}
```

مثال:

```
int a = 30;
if (a >=25)
{
    sole.WriteLine("Pass.");
}
else if (a >= 20)
{
    Console.WriteLine("Pass");
}
else
{
    Console.WriteLine("Good evening.");
}
```

if else باستخدام علامة الا

الصيغة المخت

variable = (condition) ? expTrue : expFalse;

مثال

```
int a = 20;
string result = (a < 25) ? "fail" : "pass";
Console.WriteLine(result);</pre>
```

أمثلة على جملة الشرط if البسيطة:

```
static void Main(string[] هروه) {

//وهمروات رادهال المعمورات والمعالية المعالية المعال
```

يمكن كتابة جملة if على سطر واحد في حالة تنفيذ حالة واحده فقط ي التحقق وأخرى في قق "سطر برمجى واحد" كما سب

أمثلة على جملة الشرط المركبة else if

```
string deg;
double total;
Console, WriteLine("Welcome please enter total degree..(max 600),:");
deg=Console.ReadLine();
total - double.Parse(deg);
    //if else if "Grade" : remove /* */ to rum cod
      string grad;
      double percent;
      percent = total / 600 * 100;
      if (percent >= 85)
          grad = "Excelent";
      else if (percent >= 75)
          grad = "Very Good";
      else if (percent >= 65)
          grad = "Good";
      else if (percent >= 50)
          grad = "Pass";
          grad = "fail";
      Console.Writeline("Student with Total: " + deg + " His Grad: " + grad);
      Console, ReadLine();
```

switch case جملة

وتستخدم في انتقاء بلوك كود من مجموعة بلوكات وتنفيذه بناء على قيمة متغير هي التي تحدد البلوك المراد تنفيذه.

Switch: كلمة مفتاحية لبداية الشرط.

exp: التعبير أو المتغير الذي بناء عليه يتم اختيار الحالات.

case: كلمة مفتاحية يتم تحديد احد قيم الحالات التي سيتم التبديل بينها بناء على التعبير الموجود مع select.

.... القيم التي سيتم تعينها لكل حاله بحيث يتم الاختيار بينهم.

break: لإنهاء البلوك الخاص بأى حالة case حتى لا يدخل للحالة التالية ويقوم بتنفيذ بلوكات الكود الموجودة بها

استخدام التفرع إلى نقطة "اسم داخل الكود" باستخدام goto:

ame:

... بلوكات أو اسطر كود } goto lableName:

lableName: أي اسم يختاره المبرمج ليتم الانتقال لتلك النقطة فيما بعد.

goto: الانتقال إلى نقطة "اسم" سبق تعينه ويكتب بعد كلمة goto مثال:

```
8//------
// --- SwitchApp Example 0 woel E. Zaweel 2019-2020---//
Enamespace SwitchApp
 1
    class Program
        static void Main(string[] args)
           string deg:
           double total;
            Console,WriteLine("Welcome please enter total degree..(max 600).:");
            deg - Console.Readline();
            total = double.Parse(deg);
            string nn;
            int na
 choise:
            Console.WriteLine("Enter 1 : for state & 2 : For Grad ?");
            nn-Console.ReadLine();
            n = int.Parse(nn);
switch(n)
      break;
   case 1:
       //simpl if "Pass and Fail"
       string stat;
       If (total >= 300)
          stat - "Pass";
       else
          stat = "Fail";
       Console.WriteLine("The student with total deg: " * deg * " his stat : " * stat);
       goto case 0;
       //break;
   case 21
```

```
case 2:
//if else if "Grade"
           string grad;
           double percent;
           percent = total / 600*100;
           if (percent >= 85)
               grad - "Excelent";
           else if (percent >= 75)
              grad = "Very Good";
           else if (percent >= 65)
               grad + "600d";
           else if (percent >- 50)
              grad = "Pass";
           else
              grad - "Fail";
           Console.Writeline("Student with Total : " + deg + " His Grad : " + grad);
           goto case 0;
           //breaki
   Consols.WriteLine("Value Must be 1 Or 2");
   goto choise;
   //breaks
```

Ctrl+F5 للتنفيذ

Ctr للتنفيذ

🐠 ثانيا جمل التكرار

ملة التكرار for

```
for (Start; Condition; Incremental/ Decremental)
{

الكود الذى سيتم تنفيذه "مره على الأقل حتى لو "لم يتحقق الشرط"
}
```

Start: متغير عداد الحلقة التكرارية وإعطاءه قيمة البداية "يجوز الإعلان المكان".

Condition: شرط استمرار الحلقة التكرارية مع مراعاة أن الحلقة سيتم تتفيذها مره على الأقل حتى لو لم يتحقق الشرط.

Incremental/Decremental: نمط الخطوة في الحلقة تصاعدي أو تنازلي.

```
++i تستخدم للزیادة بمقدار ۱
--i تستخدم للنقصان بمقدار ۱
i+=5 للزیادة بمقدار ٥ ویمکن استبداله بای رقم آخر
i-=5 للنقصان بمقدار ٥ ویمکن استبداله بای رقم آخر
```

مثال لجملة for البسيطة "بدون بلوك للكود":

Ctrl+F5 للتنفيذ

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

1
2
3
4
5
6
7
8
10
Press any key to continue . . .
```

في Console App لمشاهدة الناتج المطبوع وعدم إنهاء البرنامج إما Debug أو نقوم بتشغيل البرنامج من قائمة Debug نختار Start Without Debugging وهي نفسها تكافئ Ctrl+F5

مثال لجملة for تناقصية

```
// use Ctrl + F5 not Start To run and see results
int i;
...
{
    // cod 2 incremental for
    for (i = 10; i >= 0; i-=2)
    Console.WriteLine(i);
}
```

Ctrl+F5 للتنفيذ

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

10
8
6
4
2
0
Press any key to continue . . .
```

لة التكرار المتداخل رب

Ctrl+F5 للتنفيذ

```
C:\WINDOWS\system32\cmd.exe
10 * 8 - 80
10 * 9 = 90
10 * 10 - 100
10 * 11 - 110
     12 - 120
      - 11
    2 = 22
   * 4 - 44
   * 5 = 55
   * 6 = 66
  * 7 = 77
  * 8 = 88
  * 9 = 99
  * 18 = 118
  * 11 - 121
   * 12 - 132
   * 1 - 12
  * 2 - 24
  * 3 = 36
   * 4 = 48
     5 = 60
        72
         84
   * 8 = 96
   * 9 = 108
12 * 10 = 120
12 * 11 - 132
12 * 12 = 144
Press any key to continue
```

Quiz: قم بتعديل جدول الضرب السابق بحيث كل جدول يطبع في بدايته رقم الجدول (1) Table of ويبدأ كل جدول من الرقم في نفسه

جملة if لتنفيذ Block من الكود

Ctrl+F5 للتنفيذ

الشرط والتكرار الفصل الخامس

```
C:\WINDOWS\system32\cmd.exe
6 * 11 = 66
6 * 12 = 72
   ***Table Of ( 7 ) ***
 * 7 = 49
 * 8 = 56
 * 9 = 63
 * 10 = 70
 * 11 = 77
  * 12 = 84
  ***Table Of ( 8 ) ***
8 * 8 = 64
 * 9 = 72
8 * 10 = 80
 * 11 = 88
8 * 12 = 96
   ***Table Of ( 9 ) ***
9 * 9 = 81
9 * 10 = 90
9 * 11 = 99
9 * 12 = 108
   ***Table Of ( 10 ) ***
10 * 10 = 100
10 * 11 = 110
10 * 12 = 120
***Table Of ( 11 ) ***
11 * 11 = 121
11 * 12 - 132
   ***Table Of ( 12 ) ***
12 * 12 = 144
Press any key to continue
```

for each جملة التكرار

تستخدم لعمل حلق رر بناء على عدد معلوم من

والذي يكون على شكل متغير من النوع مصفوفه array.

```
foreach (type varName in arrayName)
الكود الذى سيتم تنفيذة //
```

type: نوع المتغير "دليل الحلقة"

varName: اسم المتغير "دليل الحلقة التكرارية"

arrayName: المتغير المنظومي "المصفوفه" والذي سيتم ابحث بداخله.

```
للإعلان عن متغير منظومي وإعطاءه قيم.

int [] numbers = new int[] { 3, 14, 15, 92, 6,7 };

بدون استخدام new قبل نوع البيانات عند تعيين قيم المصفوفة سيظهر خطأ
```

مثال: برنامج لحساب عدد الأرقام الزوجية والفردية داخل نطاق رقمى الأرقام ۱۵٬۹۲٬۲٬۷ دورا

لك العناصر:

Ctrl+F5 للتنفيذ

while جملة

نظرا لكتابة الشرط في رأس الحلقة التكرارية فإن الحلقة لن يتم تنفيذها إلا في حال تحقق الشرط حيث يتم اختبار الشرط أولا وبناء عليه يدخل الحلقة أم لا.

مثال

Ctrl+F5

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

Var I = 1

Var I = 2

Var I = 3

Var I = 4

Var I = 5

Var I = 6

Var I = 7

Var I = 8

Var I = 9

Var I = 10

Press any key to continue . . .
```

do while جملة

```
do
{

// وسيتم تنفيذه مره على //

الكود المراد تنفيذه (وسيتم تنفيذه مره على الأقل حتى لو لم يتحقق الشرط نظرا لكتابته في نهاية

الحلقة) ثم بعدهالن يتم تنفيذه الا عند تحقق الشرط

الموجود في نهاية الحلقة

while (condition)
```

نظرا لكتابة الشرط في نهاية الحلقة التكرارية فإن الحلقة يتم تنفيذها مره لل الأقل حتى ولو لم الشرط حيث يتم اختبار الشرط ف الحلقة وبناء عليه يدخل الحلقة مره ثانية أم لا.

مثال

Ctrl+F5

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

Var I = 0

Var I = 1

Var I = 2

Var I = 3

Var I = 4

Var I = 5

Var I = 6

I = 7

Var I = 8

Var I = 9

Press any key to continue . . .
```

تستخدم Continue في الحلقة التكرارية لأخذ قيمة جديده لمتغير الحلقة التكرارية والذهاب للدورة التالية دون تنفيذ باقى كود الدورة الحالية

array المصفوفات

المصفوفات Array هي تشبه المتغيرات العادية، ولكنها تحتوى على أكثر من قيمة وتستدعى تلك القيم جميعا بنفس اسم متغير المنظومة ويتم التمييز فيما بينهم عن طريق موضع القيمة داخل المتغير. والمصفوفة قد تكون من بعد واحد على شكل صف أو عمود، وقد تكون من بعدين أفقى ورأسى على شكل جدول من عدة أعمدة وصفوف، أو قد تكون ثلاثية الأبعاد على شكل مكعب أفقى ورأسى وعمق.

المصفوفات أحادية البعد:



ويتم تمثيلها على شكل صف أو عمود مكون من عدة عناصر، لنفرض أننا قمنا بإنشاء متغير مصفوفه باسم [3] A من بعد واحد كما في الشكل السابق فسيصبح على شكل عمود وسيتم تخزين الأسماء بداخله عنامل معها عن ط

رقم صفر وآخر قيمة الموضع رقم ٢ فحينما نريد عرض اسم زياد نستدعيه من خلال الموضع ١ ويكتب [1] A ونفس الوضع عند تخصيص قيمة للمتغير. وعادة ما يتم استخدام حلقة تكرارية لاستخراج القيم أو تخصيص القيم للمتغير أحادى البعد.

للإعلان عن مصفوفة من بعد واحد

DataType [] array_name=new DataType[no];

حيث أن:

DataType: نوع البيانات للمتغير

[] : دلیل علی انه متغیر مصفوفه من بعد ۱

Array_name: اسم المتغير

new DataType: نفس نوع البيانات المذكور في بداية الإعلان

no]: عدد العناصر دا تغير

مثال للإعلان عن متغير مصفوفه نصى مكون من ٣ عناصر

string[] data = new string[3];

لتخصيص قيم المصفوفة أثناء الإعلان:

DataType [] array_name={val1,val2,.....};

حيث أن:

DataType: نوع البيانات للمتغير

: دلیل علی انه م

Array name: اسم المتغير

(val1,val2,....) عناصر المتغير "القيم" ويجب أن تكتب من نفس نوع الإعلان عن المتغير ويتم الفصل بين تلك القيم بفاصله ,

مثال لإعطاء القيم لمتغير المصفوفة أحادي البعد أثناء الإعلان عنه:

string[] data = {"Cairo", "Tanta", "Alex"};

المصفوفات ذات البعدين:

	•	1	
•	احمد	طنطا و	ag
١	محمد	المحلة	نوفة ب
۲	إبراهيم	كفر الزيات	عرين
	A	[2,3]	£.

ويتم تمثيلها على شكل جدول من بعدين أفقى ورأسى، لنفرض أننا قمنا بإنشاء متغير مصفوفه من بعدين باسم [2,3] كما في الشكل السابق فسيصبح على شكل جدول مكون من عمودين وثلاثة صفوف وسيتم تخزين الأسماء بداخله والتعامل معها عن طريق موضعها أفقى ورأسى داخله حيث أول قيمة تأخذ الموضع رقم (صفر، صفر) وآخر قيمة الموضع رقم (1,0) فحينما نريد عرض اسم محمد مثلا نستدعيه خلال الموضع عند

قيمة للمتغير. وعادة ما يتم استخدام حلقتان تكراريتان متداخلتان لاستخراج القيم أو تخصيص القيم للمتغيرات ثنائية الأبعاد.

للإعلان عن مصفوفة من بعدين

DataType [,] array_name=new DataType[no1,no2];

حيث أن:

DataType: نوع البيانات للمتغير

[,] : دليل على انه متغير مصفوفه من بعدين نظرا لوجود الفاصلة

Array_name: اسم المتغير

new DataType: نفس نوع البيانات المذكور في بداية الإعلان

[no1, no2]: الرقم ألسد العناصر الأفقى والرقم الثانا

مثال على المصفوفات ثنائية الأبعاد:

```
overArray_class_interitance App Example 8 worl E. Zauenl 1012-----//
         manuspace Arraylas
ia
                class Progres
11
12 12 14 15 16 17
                    static wold Main(string[] arms)
                         //puts to adjust party or tite?:
string[,] data = new string[3, 2];
string field = "";
1H
1H
                          Property of the passed passed companies for (let 1 = 0; 1 < 3; 1++)
                               for (bet x = 0; x < 2; x++)
22 2 2 2 2 2 2 2 2 2 2
                                     switch (x)
                                               field + "18";
                                               bresk;
                                          cese Ir.
                                               field + "New";
                                               break;
```

```
Consula.Britatine("Enter Student Number : " + 1 + " " + field);
                           data[i, s] - Consule.Resultine();
23
34
33
36
37
                    Consols.NriteLine("Fress Ray To Frint Out The Array Values....");
38
                   Dennis Assettine();
40
                   استرجاع أو شباعة البياليات من المستوفة //
                    Community intime ("id; Hame");
42
43
                    for (int 1 = 0; 1 < 3; 1++)
44
                        for (int x = 0; x \in 2; x \mapsto)
45
AL
47
                           Community (data[1, x] + "(1");
41
49
                       Conssin.WriteLine();
54
51
52
53
```

للتنفيذ Ctrl+F5

```
Enter Student Number : 0 id
Enter Student Number : 0
                          Name
Ahmed
  er Student Number : 1
                          id
Enter Student Number : 1
                          Namo
Enter Student Number : 2
Enter Student Number : 2 Name
Press Key To Print Out The Array Values....
       Name
       Ahmed
       Adel
       Mohammed
Press any key to continue . . .
```

الطرق Methods مة للتعامل مع المصفوفات:

وتعمل مع المصفوفات الرقمية وقبل إستخدامها يجب التحقق من وجود مكتبة System.Ling في بداية الكود كالتالي:

```
using System.Linq;
```

وذلك لتفعيل تلك الطرق داخل محرر الكود وعدم إعطاء أخطاء، وهي تتيح كل من Max للقيمة العظمي داخل المصفوفة، Min للقيمة الصغري داخل عناصر المصفوفة، Sum لحاصل جمع عناصر المصفوفة.

مثال:

```
moding Systems
          onling System.Lines.
          unleg System Personing, Tooling
         //-----drray_class_inheritance App Frample 6 usel E. Cassel 2022------//
               class Progress
                     static sold Main(string[] soul)
ŝ
                                 Wash stable! System Line
10
                                int[]e = new lot[5];
fur (int 1 = 0; 1+5; 1++)
18
                                     (4(1) - (*)4)
11
                                 Convenie, Writes live("No of Array | a) | Elements - " + a.Langth);
                                (mosale, N-1;miles | "Max Value of Array \'s\" Elements - " + s.Non());
fameds, Arithiles | "Non of Array \'s\" Elements - " + s.Nin());
Emmals.N-1;miles | "Non of Array \'s\" Elements - " + s.Sin());
8.7
84
a i
```

للتنفيذ Ctrl+F5

```
No of Array 'a' Elements = 5
Max Value of Array 'a' Elements = 12
Min Value of Array 'a' Elements = 0
Sum of Array 'a' Elements = 30
y key to continue . . .
```





🛭 الخصائص العامة

يقصد بها تلك الخصائص المشتركة بين أغلب أدوات التحكم التي يتم إضافتها على النموذج ومنها:

المعنى	الخاصية	
اسم أداة التحكم الذي سيتم استخدامه		
للإشارة إليها داخل الكود من خلاله	Name	
وكذلك في الأحداث المختلفة.		
النص الظاهر داخل الأداة.	Text	
غيير حجم الأداة تلقائيا	toSize	
تحديد ما إذا كان تغيير الحجم في اتجاه	AutoSizeMode	
الزيادة فقط أم في اتجاه الزيادة والنقصان	AutoSizeWioue	
لون خلفية الأداة	BackColor	
لون النص داخل الأداة	ForeColor	
صورة خلفية الأداة	BackGroundImage	
تغيير شكل مؤشر الفأرة "أيقونة الفاره" عند	Comment	
المرور فوق الأداة	Cursor	
أة نشطه ومتاح التفاعل		
إيقاف تنشيط الأداة ومنع التفاعل معها	Enabled	
من الواجهة الرسومية.		
خصائص الخط وتشمل فرعيا:	The state of the s	
Name -	Font	

المعنى	الخاصية
Size -	
Bold -	
Italic -	
Underline -	
تضبط موضع الأدوات على النموذج	
وتشتمل على الخصائص الفرعية Y ،X	
علما بأن النقطة صفر، صفر هي الركن	
الأيسر العلوي من النموذج وهي للقراءة	Location
فقط أثناء التشغيل ولكن لإعطاء قيم	
لموضع وقت التنفيذ نستخدم ft	
Right	
اتجاه الفقرة في الأدوات النصية من	
اليمين إلى اليسار	RightToLeft
تضبط حجم أداة التحكم وتشتمل على	a•
Width, Height الخصائص الفرعية	Size
ترتيب أدوات التحكم مع مفتاح الانتقال	TabIndex
Tab	Tabinuex
ة الانتقال للاداه بالضغط عل	
Tab من عدمه	TabStop
محاذاة النص داخل أداة التحكم	TextAlign
إظهار وإخفاء الأداة على النموذج	Visible
السابقة بطريقتين:	وبتم تغيير أي من الخصائص

الطريقة الأولى من خلال نافذة الخصائص Properties وإعطاء قيم تتناسب مع نوع كل خاصية. وعند ضبط الخصائص في هذه الحالة فإنها تكون بمثابة القيم الابتدائية للخاصية بمعنى أن المشروع عند بدء التشغيل ستكون تلك قيم الخصائص مالم يتم تغييرها في كود بداية المشروع أيا كان نوعه.

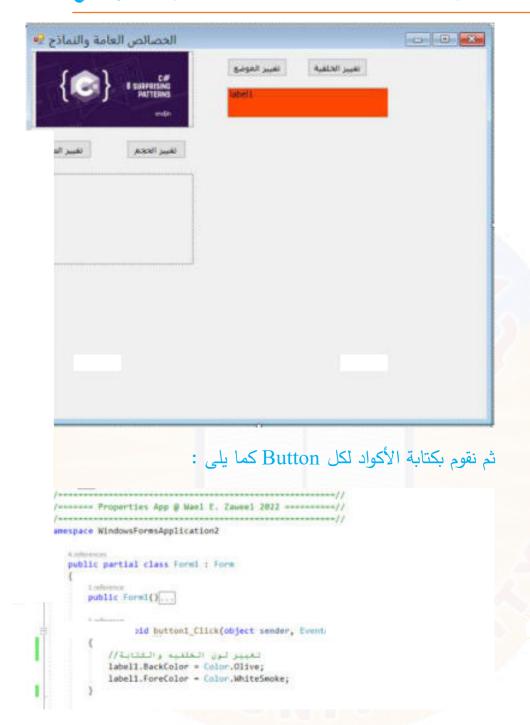
الطريقة الثانية من خلال نافذة محرر الكود وهنا عند التغيير فإنه يتم وقت التنفيذ ولن يتم التغيير إلا بتنفيذ الكود الذي يحتوى على تغيير القيم لتلك الخصائص وما لم يتم استدعاء ذلك الكود فإن الخاصية لن تم تغييرها. ويتم تغير المن الكود كما يلى:

object.Property = value

حيث أن Object أداة التحكم مثل الأزرة مربعات النصوص وغيرها، و Property الخاصية المراد تغييرها، و Property الخاصية المراد تغييرها، و تغيير الخاصية إليها مع مراعاة أن القيمة يجب أن تتناسب مع الخاصية فمثلا الخصائص الرقمية يتم إعطاءها قيم رقميه، الخصائص المنطقية يتم إعطائها قيم منطقيه، خصائص الألوان يتم إعطائها قيم لونيه وهكذا.

مثال على تغي صائص من خلال الكود:

الأدوات المستخدمة أداتى صور أحدها تم تحميل صوره بداخلها من الخصائص وأدوات Button للضغط عليها لتنفيذ الاكواد وأداة Label



```
private void button2 Click(object sender, EventArgs e)

{

// الموتع المعين الموتع الموتع المعالي المهاللة المهاللة المعالي ا
```

التنفيذ:



و ثانيا التعامل مع النماذج

الخصائص Properties المميزة للنموذج:

المعنى	الخاصية
المفتاح الافتراضي على النموذج، وهنا	
نختار أحد الـ Button المضافة للنموذج	
وجعله المفتاح الإفتراضي أي يتم تنفيذ الكود	AcceptButton
الخاص به عند الضغط على مفتاح Enter	
من لوحة المفاتيح	
اح الهروب الافتراضي على الذ	
ا نختار أحد الـ Button الـ	
للنموذج ونجعله مفتاح الهروب الإفتراضي	CancelButton
أي يتم تنفيذ الكود الخاص به عند الضغط	
على مفتاح Escape من لوحة المفاتيح	
تتحكم في طريقة إظهار نافذة النموذج	
(نافذة أدوات، قابلة للتحجيم، بدون إطار	FormBorderStyle
وغيرها)	
النموذج يستقبل المفاتيح	
الضغط عليها من لوحة المفاتيح قبل أداة	KeyPreview
التحكم النشطة.	
اختيار أيقونة للنموذج	Icon
التحكم في درجة شفافية النموذج	Opacity

المعنى	الخاصية
تأخذ القيم True, False لعرض وإخفاء	ControlBox MinimizeBox
صندوق التحكم، زر التصغير، زر التكبير.	MaxmizeBox
موضع ظهور النموذج عند بدء التشغيل	StartPosition
تجعل النموذج يظهر في مقدمة النماذج	
الأخرى " يصبح في الواجهة الأعلى فوق	TonMost
جميع النماذج المفتوحة" في حالة اختيار	TopMost
القيمة True	

الأحداث vents يزة للنموذج

الحدث هو المسبب لتنفيذ كود معين أي أن الكود المكتوب بداخل الحدث لن يتم تنفيذه إلا بوقوع الحدث ومن أهم أحداث النماذج:

وقت وقوع الحدث	الحدث
يقع عند تحميل النموذج وقبل ظهوره	Load
يقع عند انتقال التنشيط للنموذج أو أحد أدوات التحكم الموجودة عليه	Activate
ند انتقال التنشيط من النموذ إلى نموذج آخر أو نافذه أخرى	Deactivate
يقع عند تغيير حجم النموذج	Resize
يقع أثناء إغلاق النموذج	Closing
يقع عند إغلاق النموذج	Closed

يقع عند إظهار النموذج ويعنى رسم النموذج أو إظهار الواجهة الخاصة به

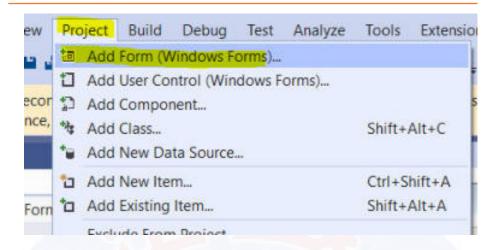
Paint

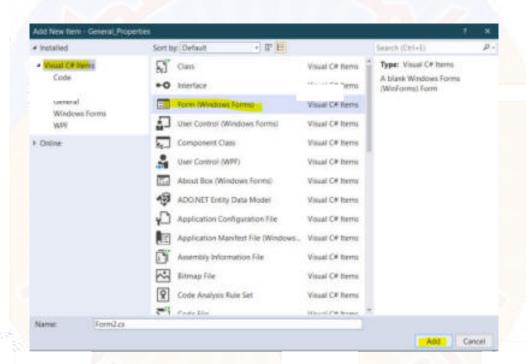
الطرق Methods المميزة للنموذج:

الطرق تختلف عن الخصائص في أنه لا يمكن تخصيص قيم لها فلن تجد بعدها علامة التساوى وتعتبر بمثابة أوامر يتم تنفيذها.

المعنى	الطريقة Method
إظهار النموذج	Show()
إظهار النموذج كنافذة حوارية أعلى جميع اذج ومنع التعامل مع النماذج الأ	ShowDialog()
إخفاء النموذج ولكنه مازال في الذاكرة ومحتفظ بجميع المتغيرات على مستوى النموذج والتعديلات التي تمت عليه لحين عرضه مره أخرى أو إغلاقه نهائيا.	Hide()
إغلاق النموذج وإزالته من الذاكرة	Close()

ولإضافة نموذ وع يتم ذلك من خلال قائمة ولإضافة نموذ وع يتم ذلك من خلال قائمة مم Windows ثم Add Form ويجب أن يكون نوع المشروع من الأساس Form App





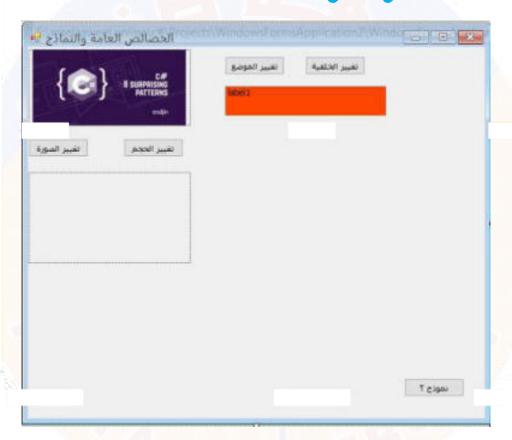
عند إضافة أي عدد من النماذج فإن المشروع سيظل يبدأ التنفيذ من النموذج الأول ولكى يتم تغيير ذلك إلى نموذج مختلف نكتب ذلك صراحة داخل كود Program Main والموجود بداخل ملف Solution Explorer داخل نافذة متصفح الحل Program.Cs

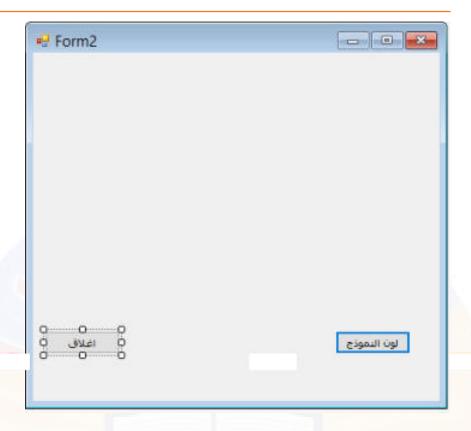
Application.Run(new Form1());

ويستبدل Form1 باسم النموذج الذي ترغب في بدأ المشروع به.

ملاحظة: داخل الكود عند الرغبه في التعامل مع النموذج الحالي فيتم الإشارة إلية من خلال كلمة This

مثال للتعامل مع النماذج المتعددة:





نفس المثال المذكور في بداية الفصل الذي يتحدث عن ضبط الخصائص العامة سنقوم بإضافة كل من نموذج ثان و Button للنموذج الأول نستخدمه في عرض النموذج الثاني، مع إضافة زرى Button على النموذج الثاني احدهما لتغير خلفية النموذج والثاني لإغلاقه وجعل على النموذج الثاني احدهما لتغير خلفية النموذج والثاني لإغلاقه وجعل الزر الخاص بالتلوين مفتاح أساسي AcceptButton والخاص بالخروج الثاني المروب المروب من خصائص النموذج الثاني

في النموذج الأول كود استدعاء النموذج الثاني:

الكود التالي:

```
private void button5_Click(object sender, EventArgs e)

(

//المحتى تفيية المحتى المحتى من كالن فنه واصلعدامها فعه Show فعد المحتى الم
```

ثم في النموذج الثاني نكتب أكواد زرى الأمر الخاص بالتلوين والإغلاق:

```
public Form2()[...]

private void buttoni Click(object sender, EventArgs e)

{

//كنيماه بميد ما AcceptButton چه عليدا بايز بيها الزر وبها الزر بيها بايد المود عامله المود ا
```

التنفيذ



عند الضغط على زر نموذج٢ سيظهر النموذج الثانى وعند الضغط على زر نموذج٢ سيظهر النموذج الثانى بينما عند على لون النموذج أو مفتاح Esc من لوحة المفاتيح سيتم إغلاق النموذج الثانى.

ق ثالثا الأحداث العامة

وهى الأحداث المشتركة بين أغلب أدوات التحكم واشهرها أحداث لوحة المفاتيح والفأره:

وقت وقوع الحدث	الحدث
عند النقر بالزر الأيسر للفأرة	lick
يقع عند النقر المزدوج بالزر الأيسر للفأرة	DoubleClick
يقع عند بدأ الضغط على زر الفأرة وقبل اكتمال Click.	MouseDown
بقع عند ترك زر الفأرة أثناء الـ Click	MouseUp
يقع لحظة دخول الفأرة لحدود الأداة	MouseEnter
يقع عند مرور الفأرة فوق الأداة	MouseHover
ند مغادرة الفأرة للأداة	useLeave
يقع عند بدأ الضغط على أحد مفاتيح لوحة المفاتيح	KeyDown
يقع عند ترك أحد مفاتيح لوحة المفاتيح	KeyUp

يلى:

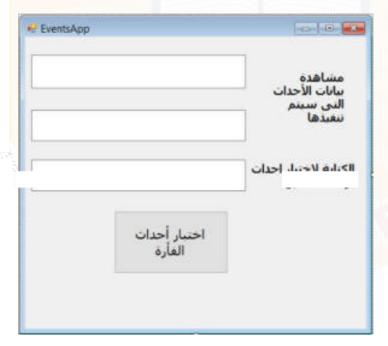
وقت وقوع الحدث	الحدث
يقع عند اكتمال ضغطة كامله على أحد	
مفاتيح لوحة المفاتيح إما بترك المفتاح بعد	
الضغط أو بمرور الزمن المسجل في	KeyPress
الحاسب للضغطة الواحدة قبل تكرار	
الحرف.	
يقع عن سحب وإفلات أداة تحكم فوق الأداة	DuagDuan
الحالية	DragDrop

وبالنقر المزدوج على المناه في نافذة التصميم فيتم فتح نافذ البرمجة على الحدث الافتراضي لتلك الأداة، وعند الرغبة في اختيار حدث آخر من الأحداث 🗴 🖚 🕶 Properties button1 System.Windows.Forms.Button الأسبق ذكرها يتم ذلك 盟里至 باختيار الأداة في نافذة button1_Click Click ClientSizeChanged التصميم ثم اختيار ContextMenuStripCha نافذة الخصائص ControlAdded ControlRemoved خول للجزء الخا CursorChanged DockChanged بالأحداث واختيار DragDrop الحدث المناسب كما DragEnter DragDrop Occurs when a drag-and-drop operation is completed.

كما في الصورة السابقة تم تنشيط أداة Button1 ثم من نافذة الخصائص تم اختيار أيقونة الأحداث ثم النثر المزدوج على حدث DragDrop فيتم مباشرة فتح نافذة البرمجة على هذا الحدث.

```
1 reference
Eprivate void button1_DragDrop(object sender, DragEventArgs e)
{
}
```

مثال يستخدم Button وأدوات TextBOX لعرض الحدث الذي تم وقوعه وكتابة كود في كل حدث عام من أحداث الفاره ولوحة المفاتيح قوم بعرض رساله دا النصوص "مع مراعاة تداخ بعض الأحداث مع بعض لذلك يجب تحويل أكواد بعض الأحداث إلى ملاحظة"



تصميم الواجهة:

الأكواد:

```
// =====EventsApp Example @ wael E. Zaweel 2019-2022=====//
  //------//
Imamespace EventsApp
  1
      public partial class Formt : Form
          public ForwI()...
          private void button1 Click(object sender, EventArgs e)
              كتابة داخل مربع النص اسم الحدث الذي لم لنفيذ: ١/١
              textBox1.Text = "Button1 Clicked";
          1
          private void button1 MouseHover(object sender, EventArgs e)
              كتابة داخل مزيع النص اصم الحدث الذي ثم تنفيذه//
              textBox1.Text = "Button1 Mouse Hover";
          private void button1 MouseDown(object sender, MouseEventArgs e)
              كتابة داخل مربع النص اسم الحدث الذيرتم تنفيذه//
              textBox1.Text = "Button1 House Down";
         private void button! MouseUp(object sender, MouseEventArgs e)
             كتابة دامل سربع النص اسم الحدث الذي تم تنسيت://
             textBox1.Text = "Button1 Mouse Up";
             لمعرفة موضع المؤشر داخل زر الامر e يمكن الاستفادة من المتغير//
             textBox1.Text += Button: " + e.Button + " X= " + e.X + " Y= " + e.Y;
         private void textBoxi_TextChanged(object sender, EventArgs e)
             كتابة في مربع النص! عند وقوع حدث تغيير المحتوى في مربع النص الاول!//
textBox2.Text = " Text on Text Box1 Changed";
         private void buttoni MouseLeave(objec. ....., ......Args e)
             //will oalstonl oalstonl
             textBox1.Text = "Left Button1";
         لمقاهدة بيانات الحدث textbox3 اعدات لوحة العفائيج قو بالكتابة داعل أداة//
         private void textBox1 KeyGown(object sender, KeyEventArgs e)...
         private void textBox1_KeyOosm(object sender, KeyEventArgs e)
             textBox2.Text = "KeyDown: e.KeyData: " + e.KeyData;
```

```
private void textBox3_KeyPress(object sender, KeyPressEventArgs e)

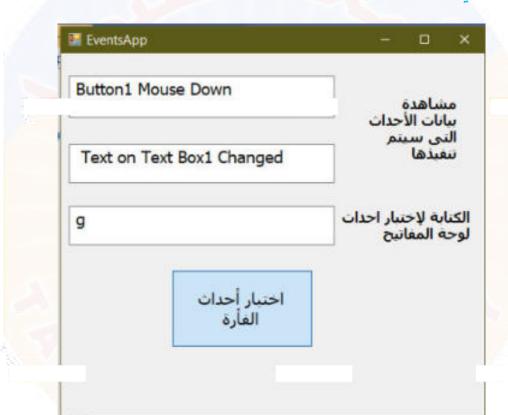
{
    textBox1.Text = "KeyPress: e.KeyChar: " + e.KeyChar;
}

private void button1_Leave(object sender, EventArgs e)...

private void Form1_KeyPress(object sender, KeyPressEventArgs e)

{
    // المعلى المع
```

التنفيذ:



لجعل النموذج يتلقى أحداث لوحة المفاتيح قبل الأدوات يجب ضبط خاصية KeyPreview الى True





الفصل السابع الأدوات الأساسية

Label, TextBox, Button الأدوات العامة

أداة العناوين Label تستخدم في كتابة بيانات توضيحيه على النموذج للأدوات، أغلب خصائصها هي الخصائص العامة المعتادة ويمكن استخدام خاصية Autosize للتحجيم التلقائى أو تركها بحجمها الذى يحدده المستخدم.

أداة أزرار الأوامر Command Button هدفها الضغط عليها التنفيذ كود معين وخصائصها وأحداثها الهامة هي نفس الخصائص والأحداث العامة.

أداة مربعات الذ TextBox النصوص وبذلك يمكن الاستفادة منها كوسيله لإدخال البيانات للمشروع، الخاصية المميزة لها هي Text، وكذلك يمكن استخدام خاصية الخاصية المميزة لها هي PasswordChar لتحويل الأداة لتلقى الباسورد بمعنى ألا تظهر بها الحروف المكتوبة وتستبدل بأحد الحروف كالنجوم مثلا **** كما هو معتاد عند إدخال الباسورد، وخاصية MultiLine والتي تتيح كتابة أسطر متعددة داخل مربع النص أو الكتابة على سطر واحد.

ومن الطرق ا مربع النص () ومن الطرق ا مربع النص () ومن الطرق ا وتستخدم في نقل التنشيط مره أخرى لمربع النص. أما الحدث المميز TextChanged ويقع عند تغير محتوى الأداة "النص المكتوب" إلى أي نص آخر سواء بزيادة أو نقصان حرف أو أكثر

الفصل السابع الأدوات الأساسية

H & V Scroll Bars أشرطة التمرير



شريط التمرير الأفقى والرأسى في حد ذاته لا يقوم بتمرير أي Scroll شيء أو تحريكه أو تغييره، ولكن عند برمجة الحدث المميز الخاصية لتنفيذ كود محدد عند تحريك منزلق شريط التمرير معتمد على الخاصية المميزة للشريط كالتي تحدد موضع المنزلق على شريط التمرير إنه بإمكانك تغيير مو روض من مكان لآخر أو تغيير لون قيمة نصية وهكذا.

المعنى	الحدث/ الخاصية
الحدث الرئيسى المميز لأداتى Scroll ويقع	
عندما نقوم بتحريك المنزلق الصغير الموجود	
على شريط التمرير سواء مباشرة أو من خلال	_Scroll
الضغط على الأسهم الموجودة على جانبي	
الخاصية الأساسية للأداة والتي تعنى قيمة أو	Value
موضع المنزلق على الشريط وتكون قيمته رقمية	value
قيمة التغير الكبير في شريط التمرير وهي الزيادة	LargChange
التي تتم على قيمة الشريط عند الضغط على	Largenange

المعنى	الحدث/ الخاصية
جسم الشريط في المسافة ما بين المنزلق وأسهم	
الزيادة والنقصان.	
قيمة التغير الصغير في شريط التمرير وهي	
الزيادة التي تتم على قيمة الشريط عند الضغط	SmallChange
على أسهم الزيادة والنقصان في طرفى الشريط.	
بداية التدريج للشريط (القيمة الصغرى)	Minimum
نهاية التدريج (القيمة العظمى)	Maximum
سريط التمرير الأفقى:	مثال لاستخدام ش



الهدف من المشروع: استخدام قيمة أشرطة التمرير الثلاثة لتغيير خلفية أداة العناوين من خلال ال Method الخاصة بخلط الألوان RGB.

الأدوات المستخدمة:

تستخدم في إظهار قيمة كل شريط تمرير في الثلاثة، "Label : " تستخدم كعنوان لكل شريط تحدد اللون الذي يتحكم به R,G,B " تستخدم كعنوان لكل شريط تحدد اللون الذي الذي الذي الذي الدي الدي الثان الذي الدي الثان الذي الثان الثان الذي الثان ال

والأخيرة نستخدمها في تغيير لون الخلفية بناء على أشرطة التمرير. مع تغيير خاصية Text إلى القيم الموجودة في الشكل السابق.

RGB: ٣ واحد لكل لون من الألوان الأساسية HScrollBar: مع تغيير قيمة Maximum إلى ٢٥٥ وهي أقصى درجة تستخدمها دالة خلط الألوان لكل لون.

الأدوات الأساسية

```
private void hScrollBar3_Scroll(object sender,
ScrollEventArgs e)
{
         Color_mix();
     }
}
```

نتيجة التنفيذ:



استخدام (,,) Color.FromArgb تعود بلون مخلوط من درجاته الثلاثة Red و Green و Blue بناء على الأرقام الثلاثة التي سيتم إدخالها والتي تتراوح بين ٠: ٢٥٥

:Option, Check أدوات

وتستخدم في اختيار قيمة من بين مجموعة عناصر واستخدامها دائما مرتبط بأداة Frame حيث يتم إضافة أكتر من أداة Check دائما مرتبط بأداة Opt حيث يتم إضافة أكتر من أداة داخل الإطار والفارق هنا أن أداة الداة الإطار عنصر واحد فقط داخل الإطار Frame.

كمثال لتوضيح الفارق بين استخدام Option, Check تخيير المستخدم بين مجموعه من العناصر الحالة الأولى هي الهوايات فمتاح للمستخدم اختيار اكتر من هواية بوضع علامة صح بجانب الهواية التي يرغب بها في هذه الحالة سنستخدم أدوات Check لتلك الهوايات، وفي حالة اختيار التخصص الدراسي بين مجموعة تخصصات فالمستخدم لا يمكنه اختيار إلا تخصص واحد فقط وعند اختيار غيره سيلغى السابق في هذه الحالة نستخدم Option.



الأدوات الأساسية

```
// ------Check Example @ wael E. Zawee12022-----//
//========
Enamespace check
     public partial class Form1 : Form
         public Forwi() ...
         private void Forel_Load(object sender, EventArgs e)...
         private void button1_Click(object sender, EventArgs e)
             الأعلان عن صلفيز نصي واعطاء؛ فيمة بدائية//
             string txt="";
             علامه"-" تستخدم لاعظاء المتغير قيمة ولا يجوز استخدامها في المقارنة//
             ونستخدم للمقارنة علامتي "==" بدلا منها//
            if (checkBox1.Checked == true) txt += "زسم";

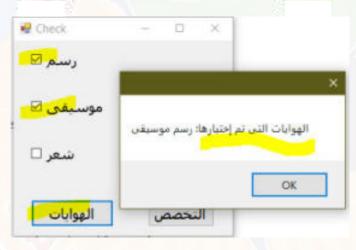
if (checkBox2.Checked == true) txt += ";

if (checkBox3.Checked == true) txt += ";
             HessageBox.Show("اللي لم إسليارها") +txt);
          private void button2_Click(object senger, EventArgs e)
              الاعلان عن متغير من النوع لموذع ١/١/
              Form2 f2 = new Form2();
              عرض العتفير المعلق عله كلموذج؟ على فيثة صلدوق حواري//
              f2.ShowDialog();
```

كود النموذج الثاني:

التنفيذ:

عند اختيار الهوايات المناسبة ثم الضغط على زر الهوايات تظهر الاختيارات التي قمنا بها في الرسالة.





عند الضغط على التخصص يفتح النموذج الثانى نختار منه احد التخصصات ثم الضغط على زر التخصص فيظهر هذا التخصص في الرسالة.

ListBox, CheckedListBox, ComboBox قوائم العناصر

الأدوات الثلاثة تستخدم في عمل قائمة من العناصر المختلفة كقائمة ن المدن مثلا أو قائمة سبوع وهكذا. وتختلف في طبيعة تلك العناصر فقائمة X ListBox تعرض عناصر مغلقه على نفسها لا يستطيع المستخدم كتابة عنصر خارج عناصر القائمة ونفس الشيء في يستطيع المستخدم كتابة عنصر نوجود صندوق اختيار بجانب كل عنصر يتم وضع علامة صح به دليل على الاختيار، بينما عنصر يتم وضع علامة صح به دليل على الاختيار، بينما ComboBox عبارة عن قائمة من العناصر مع إتاحة للمستخدم كتابة قيم من خارج القائمة.

فمثلا لو عناصر القائمة المدينة وتم تحديد مدن محافظة الغربية على اعتبار أن الكلية إقليمي ولكن لسبب ما تم قبول طالب من محافظة الإسكندرية في حالة ListBox و CheckedListBox غير مسموح للمستخدم أن يكتب قيم جديده فلن نتمكن من إضافة الإسكندرية بعد

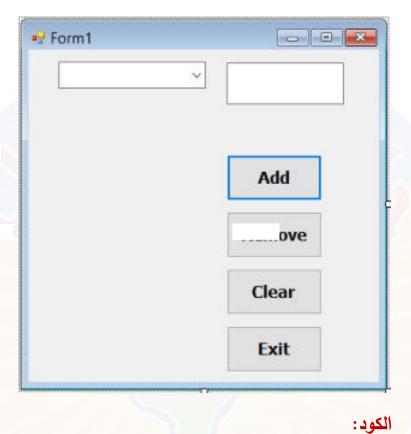
التنفيذ بينما في ComboBox يتيح مربع لكتابة القيم الجديدة بالإضافة للاختيار من عناصر القائمة فمتاح كتابة الإسكندرية في هذه الحالة.

وفيما يلى قائمة بأهم الخصائص Properties والطرق الخسائص Methods والأحداث Events المميزة لأدوات القوائم الثلاثة فهى مشتركه بين الثلاث أدوات الاختلاف فقط في طريقة تعامل الأداة مع العناصر كما سبق توضيحه في الفقرات السابقة.

المعنى	الحدث/ الخاصية	
ترجع بالقيمة النصية التي تم اختيارها داخل	Text	
<u>قوائم</u>	TOAT	
رقم العنصر المختار بين باقى عناصر القائمة		
علما بأن العناصر تبدأ العد من الرقم صفر	SelectedIndex	
هو أول عنصر		
إضافة عنصر جديد للقائمة	Items.Add()	
حذف العنصر المحدد من القائمة	Items.Remove()	
مسح أو تفريغ القائمة من جميع العناصر	Clear()	

غالبا نستخدم إضافة عناصر القوائم ()Items.Add في الحدث عناصر لنموذج. Load وذلك لتحميل العناصر داخل القائمة في بداية ظهور النموذج.

مثال على استخدام أداة ComboBox بإدخال مجموعة من المدن داخل أداة النصوص TextBox وقت التنفيذ وحذفها وتفريغ القائمة كما هو موضح بالواجهة الرسومية للمشروع:



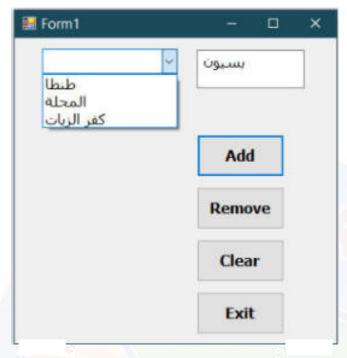
```
-----ListBoxFrmApp # Wael E. Zaweel 2022-----//

⊟e WindowsFormsApplication4

 Bic partial class Form1 : Form
 -public Form1()...
 private void button2 Click(object sender, EventArgs e)
      اعلاق المتموذع//
      this.Close();
  }
  1 reference
=private void buttonl_Click(object sender, (ventArgs e)
    CoeboBox1 Dated Service Services (Line CoeboBox1
    comboBox1.Items.Add(textBox1.Text);
    معج النص من مربع النص ونقل التنقيط إليه استعداد للثنابة قيم جديده//
    -wiffox1.Text - "";
     xtBox1.Focus();
aprivate void combufuel SelectedIndesChanged(object sender, fountArgs a)...
private void buttund_Click(object sender, EventArgs e)
    //pelix lif this ComboSon
    comboBox1.Items.Clear();
 private void buttoni_Click(object sender, EventArgs e)
    // المنصر العلم العالى ولئن بعد التائد من أن فناك منصر بالفعل باحتمار 1/(10000) f (coeboScx1.SelectedIndex >>0)
    comboBox1.Items.RemoveAt(comboBox1.SelectedIndex);
```

التنفيذ

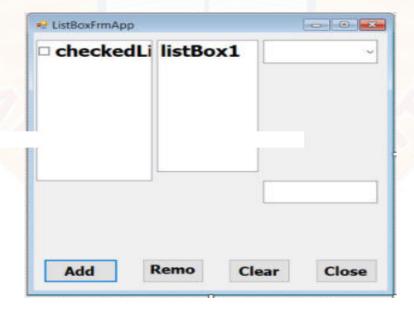
الأدوات الأساسية الفصل السابع



يتم كتابة مدينة داخل مربع النص والضغط على زر Add فتضاف تلقائي للقائمة والضغط على Remove يحذف المدينة المحددة في Clear ،قائمة

حذف كل عناصر القاد عط على Exit يخرج من البرنام

مثال ٢ لتعبئة عناصر القائمة عند بدء التشغيل من خلال الحدث Form Load لأدوات القوائم الثلاثة:



الكود:

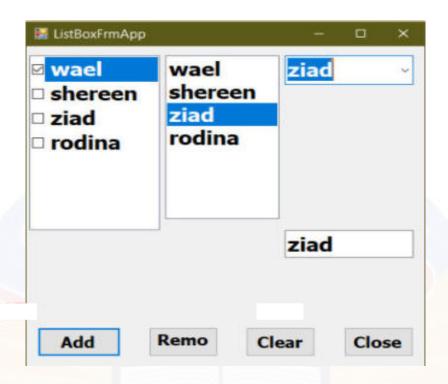
```
-//-----//
 //----ListBoxFrmApp Example © wael E. Zaweel 2022----//
□namespace ListBoxFrmApp
     3 references
     public partial class Form1 : Form
         1 reference
        public Form1()...
         private void Form1_Load(object sender, EventArgs e)
            إضافة العناصر عند بده التشغيل لكل القوائم//
            //adding items to list box1
            listBox1.Items.Add("wael");
            listBox1.Items.Add("shereen");
            listBox1.Items.Add("ziad");
            listBox1.Items.Add("rodina"
            //adding items to checked list box1
            checkedListBox1.Items.Add("wael");
            checkedListBox1.Items.Add("shereen");
            checkedListBox1.Items.Add("ziad");
            checkedListBox1.Items.Add("rodina");
            //adding items to combo box1
            comboBox1.Items.Add("wael");
            comboBox1.Items.Add("shereen");
            comboBox1.Items.Add("ziad");
             comboBox1.Items.Add("rodina");
```

```
private void button4 Click(object sender, frentArgs e)
   //exit form or class code
     Inte-Class();
    //if u need to end app use this line
    // Application.fait();
الأكواد البالية تظهر النص الديايتم الحنباره بأداة اللائمة ليتم مرضه داخل (١١١ النموم//
private void checked[istflox] SelectedIndexChanged(object sender, fromtArgs e)
    textfox1.Text - checkedListBox1.Text;
private void listBool_SelectedIndexChanged(object sender, EventArgs e)
    textBox1.Text - listBox1.Text;
private void comboflox1_SelectedIndexChanged(object sender, (ventArgs e)
    textBox1.Text = comboBox1.Text;
grivate void buttoni Click(object sander, (vent/ogs e)
    //solding item to lists from test book
    listBox1.Stems.Add(textSex1.Text);
    comboBox1. [tems.Add(textBox1.Text);
    checked_istBox1.Items_Add/textBox1.Text);
    // clear text from text boxt and set focus
    textBox1. Text = "";
    textlox1.focus();
private void buttoni (lick(object sender, frenthegs e)
    //itek if any item selected on the list before delete this item "selectindes 0 selected -1 out selected
    //in this cod we use remove propereties that removes with text value
    //If u want to remove by values index use this line
    //if (listNox).SelectedIndex !- -[) listNox1.Trms.RemoveAt(listNox).SelectedIndex);
    i+ (listBox1.SelectedIndex != -1) listBox1. Items. Nemove(listBox1. Fext);
    if (checkedListBox1.SelectedIndex 1+ -1) checkedListBox1.Items.Remove(checkedListBox1.Text);
```

مع ملاحظة: (this.Close) تغلق النموذج الحالي ولا تنهى البرنامج الا لو كان هذا النموذج الوحيد المفتوح بينما (Application.Exit)

if (comboBox1.5electedIndex !- -1) comboBox1.Items, Armove(comboBox1.Text);

التنفيذ:



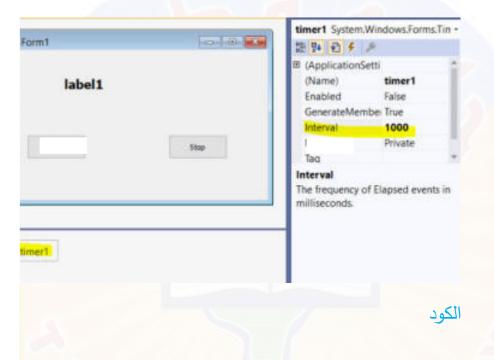
وعند الضغط على اى عنصر في اى قائمة من الثلاثة يظهر النص داخل مربع TextBox وبالضغط على Add يتم إضافة النص المكتوب داخل مربع TextBox الى القوائم الثلاثة، بينما Remove يحذف العنصر النشط في كل القوائم وClear يمسح الكل.

🥌 أداة عداد الزمن

أداة عداد الزمن ليس لها واجهة رسومية مثل باقى الداة عداد الزمن ليس لها واجهة رسومية مثل باقى الأدوات وليس لها سوى حدث مميز Tick والذى يقع بشكل دورى عند مرور الزمن المحدد في الخاصية المميزة للأداة والتي تسمى Interval وتعبر عن مرور الزمن علما بأن القيمة ١٠٠٠ تساوى

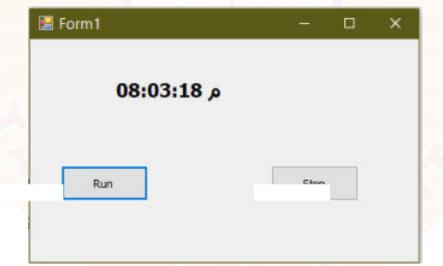
ثانيه واحدة. ويمكن تشغيل الـ Timer وإيقافه من خلال خاصية .Enabled

مثال على استخدام Timer بعداد زمنى كل ١ ثانية "يجب ضبط Interval=1000" بحيث يعرض الوقت المسجل على الحاسب مما يعطيك إيحاء بتحرك الثواني كأنها ساعه حقيقية.



```
//----TimerApp Example 0 wael E. Zaweel 2022----//
Enamespace TimerApp
     public partial class Form! : Form
         public Form1()...
         private void timer1_Tick(object sender, EventArgs e)
         الكود الذي ينقد بعد سزور كل ١ لنانية//}
             label1.Text = DateTime.Now.ToLongTimeString() +"";
         private void button1_Click(object sender, EventArgs e)
         تشغيل التايمر//}
             timer1.Enabled - true;
         private void button2_Click(object sender, EventArgs e)
         ايقاف التايمز//}
            timer1.Enabled = false;
         private void Form1_Load(object sender, EventArgs e)
         عند بد٠ النقعيل عرض الوقت الحالي تشيعة ابتدائية//}
               bell.Text = DateTime.Now,ToLongTimeString(
```









Function الدوال

الدوال هي أكواد فرعيه تشبه إجراءات الأحداث إلى حد ما إلا أنها غير مقترنه بحدث، والدوال نوعان رئيسيان الأول void لا يعود بقيمه والثانى Method يعود بقيمة.

الإعلان عن الدوال "إنشائها":

في حالة الإعلان عن دالة void لا تعود بقيمة:

```
void function_name(type1 var1, type2 var2,....)
{
}
```

حيث:

void: تشير إلى أن الدالة لا تعود بقيمة.

function_name: اسم الدالة والذي سيتم استخدامه عند استدعائها.

type1: نوع بيانات معامل الدالة الأول.

varl: معامل الدالة الأول.

type2: نوع بيانات معامل الدالة الثاني.

var2: معامل الدالة الثاني. وهكذا لأى عدد من المعاملات التي سيتم تمريرها للدالة عند استدعائها.

{} الكود داخل أقواس المجموعة المطلوب تنفيذه للدالة مع العلم أنه لن يتم استخدام Return داخل الكود لأنها لا تعود بقيمة.

في حالة الإعلان عن الدوال التي تعود بقيمة Method:

```
type function_name(type1 var1, type2 var2,....)
{
Code
    value;
}
```

حيث:

type: نوع البيانات التي ستعود به الدالة أو الـ Method.

function_name: اسم الدالة والذي سيتم استخدامه عند استدعائها.

type1: نوع بيانات معامل الدالة الأول.

var1: معامل الدالة

type2: نوع بيانات معامل الدالة الثاني.

var2: معامل الدالة الثاني. وهكذا لأى عدد من المعاملات التي سيتم تمريرها للدالة عند استدعائها.

{} الكود داخل أقواس المجموعة المطلوب تنفيذه للدالة مع العلم أنه يجب استخدام return داخل الكود وتمرير قيمة من نفس نوع البيانات type المذكور في الإعلان عن الدالة.

استدعاء - تنفيذ - الدوال:

ولتنفيذ "استدعاء" دالة void لا تعود بقيمة يكتب الكود كالتالي:

function name(prameters);

حيث

function_name

prameters المعاملات التي سيتم تمريرها للدالة وفقا لما ورد في الإعلان عنها

ولتنفيذ "استدعاء" دالة Method لا تعود بقيمة يكتب الكود كالتالى:

var = function_name(prameters);

يث

var متغير من نفس نوع الإعلان عن الدالة لكى يحتفظ بالقيمة التي تعود من الدالة

function_name اسم الدالة.

prameters المعاملات التي سيتم تمريرها للدالة وفقا لما ورد في الإعلان عنها

مثال تطبيق الدوال التي تعود بقيمه ولا تعود بقيمة من خلال برنامج أله حاسبة بسيطة معتمد على داله يتم تمرير القيم لها بالإضافة للعملية وبناء عليه تعود بناتج العملية:

₱ FunctionA	рр	0 8 2
Num 1	0	+
Num 2	0	
Num 3		*
		1
		End

الكود

```
//======FunctionApp Example @ wael E. Zaweel 2022======//
 //------//
Enamespace FunctionApp
 1
     3 references
     public partial class Form1 : Form
         3 reference
         public Formi()
            InitializeComponent();
        //function without return value
         void End_App(string msg)
            MessageBox, Show(msg);
            Application.Exit();
      //function return a value
      double Colc(string num1, string num2, str
                                             etion)
         double a, b,r=0;
         a = double.Parse(num1);
         b = double.Parse(num2);
         if (operation=="+")
            r = a + b;
         else if (operation == "-")
            r = a - b;
         else if (operation == """)
            r = a * b;
         else if (operation == "/")
            r = a / b;
         return c;
```

```
private void buttom1 Click(object sender, franthrys e)
    double #1
    a = Calc(textBox1.Text, textBox2.Text, "4");
textBox3.Text = a + "";
 private void buttom2_Click(object sender, Eventarge e)
    deuble a;
    a - Calc(textBox1.Text, textBox2.Text, "-");
    testBox1.Text * a + "";
private void buttoml_Click(object sender, (ventArgs e)
    11 "420
    double a;
    a - Calc(textBox1.Text, textBox2.Text, "*");
    textSox3.Text = a + "";
 private void buttom4_Click(object sender, Eventargs e)
    double aj
    a = Calc(textBox1.Text, textBox2.Text, "/");
    textBox3.Text - a + "";
  1 reference
  private void button5_Click(object sender, EventArgs e)
       ("صيتم اغلاق البرنامج");
FunctionApp
                                                                                  التنفيذ
Num 1
             40
             3
Num 2
             43
Num 3
                        سيتم اعلاق البرنامج
                                  OK
                                                 End
```

Classes الفصائل

هي الحاوية لعدد من الخصائص والدوال والطرق بداخل كل منهم الكود الخاص به، ولا يتم التعامل مع الفصيلة بشكل مباشر وإنما من خلال إنشاء كائن من نفس نوع الفصيلة.

لإنشاء فصيلة:

```
Public class Class_name
{
....
}
```

حيث أن:

Public اختيارى وتستخدم في حالة الرغبة غى عمل instance من الفصيلة. ويمكن الاستغناء عنها في حالة عدم الرغبة في عمل Instance من الفصيلة.

class للإعلان عن فصيلة.

Class_nam اسم_____ی ترغب ف إنشاؤه.

{} اكواد الفصيلة "خصائص، دوال، طرق.." تكتب بداخل الأقواس المعقوفة للفصيلة.

ويجب مراعاة أن الفصيلة لا يتم التعامل معها مباشرة ولكن من خلال انشاء كائن من الفصيلة كالتالي:

```
Class_name object = new Class_name();
```

حيث:

Class_name الفصيلة التي قمنا بإنشائها.

object الكائن الجديد الذي ننشأه من الفصيلة لنتعامل معها من خلاله.

في حالة إنشاء فصيلة جديده وتحتاج لان تحتوى على Methods موجوده في فصيله أخر المشروع لا داعى لتكرارها ممكن الفصيلة الجديدة تلك الأكواد من الفصيلة القديمة وكأنها مكتوبه بداخلها ويتم ذلك في الإعلان عن الفصيلة الجديدة كما يلى:

```
class New_class_name : Old_class_name
{
.....
}
```

lass name

: نقطتين بمعنى أن ترث الفصيلة السابقة عناصر الفصيلة التالية Old class name الفصيلة السابقة المطلوب وراثة عناصرها.

دالة البناء Constructor Function

دالة داخل الفصيلة تسمى بنفس اسم الفصيلة وهى أول ما يتم تنفيذه عند التعامل معها. وغالبا يتم بداخلها إعطاء القيم الابتدائية للمتغيرات.

دالة الهدم Destructor Function

هي أيضا تكتب باسم الفصيلة، ولكن قبل الاسم علامة ~ وتستخدم كآخر كود في الفصيلة وغالبا يتم فيه التخلص من المتغيرات وجميع العناصر التي تستخدمها الفصيلة.

مثال على إنشاء الفصائل، والوراثة، ودوال البناء والهدم.

نوع المشروع CONSOL App

```
mespace Clasess

/pupilic Liass Myclass

{

public class Myclass

{

public string[,] a = new string[3, 2];

string field;

//Classiania = new string[3, 2];

string field = new string[3, 2];

field = new string[3, 2];

//Classiania =
```

```
public void read()
     for (int 1 = 0; 1 < 3; i++)
         for (int b = 0; b < 2; b++)
            switch (b)
                case 0:
                   field - "id";
                   beeaks
                case I:
                   field - "Total";
                   break;
            Console.Writeline("Enter Student Num : " + i + " " + field);
            a[1, b] - Console.ReadLine();
          //myclass-----//
class Student : Myclass
   public void print()
       Console.WriteLine("id\ttotal");
       for (int 1 - 0; 1 < 3; 1++)
           for (int b = 0; b < 2; b++)
              Console.Write(a[i, b] + "\t");
           Console.WriteLine();
   دالة الهدو//
   ~Student()
   ()
                                                     -11
```



المراجع

ANTHONY GRANT "High- and Low-Level Programming Languages,

Computer Concepts - Programming Languages"

Elshoff, James L.; Marcotty, Michael (1982).

"Improving computer program readability to aid modification". Communications of the ACM. 25 (8)

Fuegi, J.; Francis, J. (2003). "Lovelace & Babbage

e Creation of the 1843 'n IEEE Annals of

the History of Computing. 25 (4): 16

Herman Hollerith "Computing History ". Columbia University

Juergen Haas (2020), "The Difference Between Compiled and Interpreted Languages

Lonny Meinecke, "What is Programming Language? -

Types & Examples".

oft MSDN (2019)

Mitchell, Robert (2012). "The Cobol Brain Drain".

Computer World

Shaun Bebbington (2014). "What is coding."

يحظر توزيع الكتاب خارج منصة الكتب الجامعية لجامعة طنطا كما يحظر طباعة الكتاب وتداوله مطبوعا كما يحظر نسخ أو اقتباس أجزاء من الكتاب دون إذن كتابى

ل السيد زعويل

http://tdb2.tanta.edu.eg/staff/waelzaweel

https://www.facebook.com/wael.zaweel/