بسم الله الرحمن الرحيم

الحمد لله رب العالمين وأفضل الصلاة وأتم التسليم على سيدنا محمد وعلى آله وصحبه أجمعين

سبحانك اللهم لا علم لنا إلا ما علمتنا إنك أنت العليم الحكيم

الأهداف التعليمية من هذا الدرس :

1- معرفة الأجزاء الأساسية للحاسوب من الناحية الوظيفية.

2- فهم آلية عمل الحاسوب بشكل مبسط بما يمهد لفهم أساسيات البرمجة.

3- التمييز بين مستويات لغات البرمجة وهي : لغة آلة ولغة مجمع ولغة عالية المستوى.

4- معرفة الأسباب التي أدت لظهور تقنية الدوت نت (.NET).

5- معرفة آلية عملية مترجم اللغات المشترك (common language runtime:CLR)

1.1 : مقدمة عامة عن البرمجة :

إن البرمجة بالمعنى الواسع هي آلية تمكننا كبشر أن نتعامل مع هذا التجمع المتنوع للمكونات والتقنيات والتي تسمى بالحاسوب (Computer).

فمن البديهي أننا عندما نريد من الحاسوب أن ينفذ عمل ما كحل مسألة ، أو رسم صورة ، أو إجراء اتصال ..... أن نقول أننا بحاجة إلى لغة نستطيع من خلالها التخاطب مع الحاسوب وإملاء الأوامر عليه لكي يقوم هو بدوره بتنفيذها.

لذلك إذا أردنا أن نتعلم هذه اللغة فلا بد لنا أولاً من فهم آلية عمل هذا الذي نريد أن نتخاطب معه (الحاسوب) وذلك بالتعرف على ماهية الأعمال الأساسية التي يقوم بها.

بداية سنقوم بالتعرف على الحاسوب بصورته المجردة والتي سوف تساعدنا على فهم كيفية التعامل معه.

رغم أن الحاسوب كما قلنا يتألف من تجمع مكونات وتقنيات متطورة عديدة إلا أن الطريقة الأساسية لعمل الحاسوب بسيطة جداً ولم تتغير كثيراً منذ الثورة الالكترونية التي ظهرت منتصف القرن العشرين.

2.1 : مقدمة عامة عن الحاسوب (Computer) :

ازداد استخدام الإنسان في عصرنا الحالي للحاسوب حيث أنه قد اعتمد عليه في كثير من المجالات ، ولعل أهم الأسباب التي أدت إلى هذا الازدياد هو الكلفة القليلة والحجم الصغير للحاسوب مقارنة لما كان عليه في العقود القليلة الماضية.

حيث كان الحاسوب يكلف ملايين الدولارات ويملأ العديد من الغرف بينما الآن فقد اختصر في رقاقة سليكون لا تتعدى حجم الإبهام وسعره لا يزيد عن مئات الدولارات.

لحسن الحظ أن عنصر السليكون هو ثاني العناصر وفرة على سطح الأرض بعد الأوكسجين مباشرة.

3.1 : ما هو الحاسوب:

الحاسوب هو جهاز قادر على إجراء العمليات الحسابية واتخاذ القرارات المنطقية بسرعة تفوق سرعة الإنسان بملايين أو حتى مليارات أو قل تريليونات المرات.

على سبيل المثال : العديد من الحواسيب الشخصية هذه الأيام يمكنها أن تنفذ مليون عملية جمع في الثانية.

قد يعيش الإنسان عمره كله بدون توقف (جدلاً) وهو يجري عمليات حسابية بينما يقوم أي حاسوب شخصي (Personal Computer) متواضع بالقيام بتلك العمليات في أقل من ثانية.

<u>دعوة إلى التأمل :</u>كيف يمكننا تحديد فيما إذا كان هذا الشخص قد قام بحسابات صحيحة؟

وكيف يمكننا تحديد فيما إذا كان هذا الحاسوب قد قام بحسابات صحيحة؟

تدعى أكثر الحواسيب سرعة بالحواسيب الفائقة (Supercomputer) والتي تستطيع أن تنجز تريليون عملية جمع بالثانية.

يعالج الحاسوب البيانات عن طريق مجموعة من التعليمات تدعى البرنامج (Program).

يرشد البرنامج الحاسوب عبر هذه التعليمات لإنجاز مهام معينة ويكتب هذه البرامج أناس مختصون يدعون المبرمجين (Programmers).

يتألف الحاسوب كما قلنا من مجموعة متنوعة من الأجهزة (المكونات) التي تدعى المكون المادي (hardware) <u>مثل:</u> لوحة المفاتيح (Keyboard) ، الشاشة (Screen) ، الفأرة (Mouse) ، القرص الصلب (Hard Disk) .

وتدعى البرامج التي تعمل على الحاسوب بالبرمجيات (Software) مثل: برنامج معالجة النصوص ، وبرامج البريد الالكتروني ، والألعاب الالكترونية.

تناقصت تكلفة المكون المادي للحواسيب كثيراً في السنوات الأخيرة بدليل ازدياد استخدام الحواسيب الشخصية بينما تزايدت تدريجياً كلفة تطوير البرمجيات كون أن مطوري البرمجيات أصبحوا يطورون برامج قوية ومعقدة وأنت في هذه السلسلة سوف تتعلم تقنية البرمجة غرضية التوجه التي من شأنها تقليل وقت وكلفة عملية التطوير.

4.1 : مكونات الحاسوب :

بغض النظر عن المظهر الفيزيائي ، بشكل عام يمكنك أن تتخيل أن كل حاسوب تنقسم بنيته المادية إلى ستة أقسام أو وحدات (Units) منطقية.

1. **وحدة الإدخال (Input Unit) :** وهي جزء "الاستقبال" في الحاسوب ، حيث أن هذه الوحدة تقوم بجلب المعلومات من أجهزة الإدخال (Input Devices) (مثل: لوحة المفاتيح والفأرة) وتضعها تحت تصرف الوحدات الأخرى والتي مهمتها معالجة هذه المعلومات. يمكننا إدخال المعلومات إلى الحاسوب بطرق أخرى بما في ذلك التحدث إلى الحاسوب ومسح الصور وأيضاً استقبال المعلومات من الشبكة.
2. **وحدة الإخراج (Output Unit) :** وهي جزء "الإرسال" في الحاسوب ، حيث أن هذه الوحدة تقوم بأخذ المعلومات المعالجة وتضعها على أجهزة الإخراج لتتيحها بدورها للوسط الخارجي ، معظم المعلومات المعالجة اليوم تعرض على الشاشة أو تطبع على الورق أو تستخدم للتحكم بالمكونات والأجهزة الأخرى.

   طبعاً يمكن للحاسوب أن يتيح المعلومات المعالجة عبر الشبكة أيضاً.
3. **وحدة الذاكرة الرئيسية (RAM: Random Access Memory) :** وهي جزء الوصول السريع والمستودع صغير الحجم نسبياً ، حيث تحتفظ هذه الوحدة بالمعلومات المدخلة من خلال وحدة الإدخال وتجعلها متوفرة للمعالجة عند الحاجة إليها.

   وحدة الذاكرة الرئيسية هذه تحتفظ أيضاً بالمعلومات المعالجة حتى يتاح وضعها على أجهزة الإخراج.

تضـيع المعلومـات المخزنـة فـي هـذه الوحـدة بانقطـاع التيـار الكهربـائي عنهـا وتـدعى هذه الوحدة أيضاً بذاكرة الوصول العشوائي.

4. **وحـدة الحسـاب والمنطـق (Arithmetic and Logic Unit: ALU) :** وهـي جـزء "الصـناعة التحويليـة" فـي الحاسـوب ، هـذه الوحـدة مسـؤولة عـن إجـراء الحسـابات مثـل الجمـع والطـرح والضـرب والقسـمة وتحتـوي أيضـاً علـى آليـة لاتخـاذ القـرارات المنطقيـة لتسـمح للحاسـوب مـثلاً مقارنـة عنصـرين مـن وحـدة الذاكرة الرئيسية وتحديد تساويهما من عدمه.
5. **وحـدة المعالجـة المركزيـة (Central Processing Unit: CPU) :** وهـي الجـزء الإداري فـي الحاسـوب ، وهـذه الوحـدة متسـاوية فـي الأهميـة مـع بـاقي الوحدات ولكن مهمتها هي الإشراف عليها.

فهـي تخبـر وحـدة الإخـراج أن هنـاك معلومـات يجـب أن تقـرأ مـن وحـدة الـذاكرة وتخبـر وحـدة الحسـاب والمنطـق أن المعلومـات فـي وحـدة الـذاكرة يجـب أن تحسـب ويبت في أمرها.

العديـد مـن الحواسـيب اليـوم تمتلـك أكثـر مـن معـالج (processor) وهـذا مـا يسـمح بتنفيـذ أكثـر مـن عمليـة فـي نفـس الوقـت ، تـدعى هـذه الحواسـيب بالحواسـيب متعـددة المعالجات (multiprocessors).

6. **وحـدة الـذاكـرة الثـانوية (secondary storage unit) :** وهـي جـزء المسـتودع طويـل الأمـد والكبيـر الحجـم نسـبياً مثـل: القـرص الصـلب. حيـث أن هـذه الوحدة لا تفقد معلوماتها بانقطاع التيار الكهربائي عنها.

يحتـاج الوصـول إلـى معلومـات مخزنـة علـى وحـدة الـذاكرة الثانويـة وقتـاً أطـول بكثيـر ممـا يتطلبـه الوصـول إلـى وحـدة الـذاكرة الرئيسـية.

5.1 : مستويات لغات البرمجة :

يكتـب المبرمجـون تعليمـاتهم بالعديـد مـن لغـات البرمجـة ، بعـض هـذه التعليمـات يفهمهـا الحاسـوب مباشـرة والـبعض الآخـر يحتـاج متـرجم وسـيط بينهمـا. يوجـد حاليـاً المئـات مـن لغـات البرمجـة المسـتخدمة والتـي يمكـن تقسـيمها إلـى مسـتويات

ثلاث : لغات الآلة (machine language) ولغات التجميع(assembly language) ولغات عالية المستوى (high level language)[1].

1. **لغات الآلة :** يستطيع الحاسوب أن يفهم مباشرة لغة الآلة الخاصة به لأنها لغته الطبيعية التي تعلمها (عرفت له) من خلال التصميم المادي لمعالجه. تتألف لغة الآلة من سلاسل من الأعداد على شكل مجموعات من الأصفار والواحدات التي تعطي الأوامر للحاسوب بتنفيذ عملياته الأولية كل تعليمة على حده.

ترتبط لغات الآلة بالآلة (طراز معالج الحاسوب) ارتباطاً وثيقاً بمعنى أن لغة آلة ما يمكن أن تستخدم فقط لنوع (طراز) محدد من المعالجات أي الحواسيب. نذكر هنا أن لغات الآلة تعتبر ثقيلة ومرهقة للإنسان.

2. **لغات التجميع :** مع انتشار الحواسيب أصبح من الملاحظ أن عملية البرمجة بلغة الآلة ليست سهلة فهي بطيئة ومملة لمعظم المبرمجين وبدلاً من استخدام سلاسل الأعداد التي يفهمها الحاسوب مباشرة بدأ المبرمجون باستخدام مصطلحات إنكليزية مختصرة للتعبير عن العمليات الأولية للحاسوب.

تمثل هذه المصطلحات الأساس الذي انطلقت منه لغات التجميع حيث تم تطوير برامج مترجمة (compilers) تدعى هنا المجمعات (assemblers) تقوم بتحويل البرامج من لغة التجميع إلى لغة الآلة.

3. **لغات عالية المستوى :** تزايد استخدام الحواسيب واستخدمت لغات المجمع على نطاق واسع ولكن بقي إنجاز أبسط المهام يتطلب استخدام الكثير من التعليمات وبالتالي من أجل تسريع عمليات البرمجة تم تطوير لغات عالية المستوى والتي تقترب أكثر من طريقة تفكيرنا بحل المسائل ، حيث تساعد هذه اللغات المبرمجين على كتابة تعليمات تبدو قريبة من العبارات الإنكليزية وتحتوي على الرموز الرياضية الشائعة.

تصنف لغة السي شارب التي سوف نتكلم عنها في هذه السلسلة من اللغات العالية المستوى.

[1] العلو والانخفاض هنا هو من حيث قرب اللغة وبعدها عن لغة الآلة.

6.1 : المترجمات (compilers) :

تسمى البرامج التي تقوم بتحويل رماز (code) برنامج من لغة إلى أخرى بالمترجمات.

تأخذ عملية الترجمة لبرنامج من لغة عالية المستوى إلى لغة الآلة زمناً كبيراً نسبياً.

1.6.1 : آلية عمل المترجم :

يقوم المترجم بتخزين الرماز الناتج بلغة الآلة في ملف تنفيذي على وحدة الذاكرة الثانوية ، وهذا الأخير هو الملف الذي يجري شحنه إلى وحدة الذاكرة الرئيسية عند طلب تنفيذ البرنامج.

يمكن كتابة الرماز البرمجي للغات العالية المستوى باستخدام محرر نصوص.

هناك نوع آخر من المترجمات ندعوها المفسرات (interpreter) حيث تستطيع ترجمة البرامج المكتوبة بلغة عالية المستوى مباشرة دون الحاجة إلى تخزين الملف المترجم على وحدة الذاكرة الثانوية.

لكن عيب هذه الأخيرة أنها أبطأ من النوع الأول (compiler) ، وبشكل عام تستخدم المفسرات في مرحلة تطوير البرامج حيث نقوم بترجمة البرنامج عند أي إضافة جديدة أو عند تصحيح خطأ ما وعند الانتهاء من البرنامج نقوم بإنشاء نسخة مترجمة عنه إلى لغة الآلة من أجل تنفيذه بفعالية أكبر.

يبين الشكل التالي مقارنة بين مستويات اللغات الثلاث وطريقة عمل كل منها :

	نموذج تعليمات	برنامج ترجمة	منظور المبرمج	منظور الحاسوب
لغة آلة	00010100 00010010 10100010	لا يحتاج	بطيئة ومجهدة واحتمال الخطأ كبير	لغتي الطبيعية وأستطيع أن أفهمها مباشرة
لغة تجميع	LOAD A ADD B STORE C	مجمع (assembler)	قريبة من المصطلحات الإنكليزية	يقوم المجمع بتحويل الرماز إلى لغتي الأم
لغة عالية المستوى	C=A+B	مترجم (compiler) أو (interpreter)	التعليمات تشبه الإنكليزية المتداولة والعبارة الواحدة تنفذ عدة مهام أساسية	يقوم المترجم بتحويل الرماز إلى لغتي الأم

7.1 : لمحة عامة عن لغة السي شارب (C#) :

إن تطـــور أدوات البرمجـــة والأجهـــزة الالكترونيـــة المختلفـــة مثـــل الأجهـــزة الجوالـــة (مثل: mobile phone و PDA: Personal Digital Assistant) تطلبت العديد من المتطلبات.

فتكامل المكونات البرمجية للبرامج الاحترافية المكتوبة بعدة لغات ومصممة وفق أكثر من تقنية ليس بالأمر السهل ، ناهيك عن مشاكل التنصيب وعدم التوافق التي أصبحت شائعة بسبب الإصدارات الجديدة للبرامج.

تطلبت شعبية وانتشار استخدام الأجهزة الجوالة أن يتم تطوير تطبيقات لا تلزم العميل بالوصول إليها عبر حاسوبه الشخصي بل يجب أن توفر هذه التطبيقات الوصول إليها بغض النظر عن نوع الجهاز المستخدم لهذا الوصول.

استجابة لهذه المتطلبات وفي عام 2000 أعلنت شركة مايكروسوفت (Microsoft) عن لغة السي شارب (C#) التي هي الوجه المطور من لغتي (C++ , C) والتي طورتها خصيصاً لتعمل متوافقة مع منصة العمل دوت نت (.NET) التي سوف نتكلم عليها في الفقرة القادمة.

طُورت C# بجهود فريق من النخبة بقيادة كلاً من : Anders Hejlsberg و Scott Wiltamuth.

استخلص هذان العبقريان أقوى الميزات الموجودة في لغتي (C,C++) و (JAVA) وأضافا ميزات تنفرد بها لغة السي شارب (C#).

8.1 : لمحة عامة عن منصة العمل دوت نت (.NET Framework) :

الفكرة الأساسية من تطوير هذه المنصة من قبل شركة مايكروسوفت كانت هي الاستقلالية. والاستقلالية هنا تعني استقلالية اللغة البرمجية المستخدمة و استقلالية منصة التشغيل (platform) التي سوف يعمل عليها التطبيق المطور.

بالنسبة لاستقلالية اللغة أصبح بإمكان المبرمجين عبر هذه التقنية الرائعة أن يتشاركوا في المشروع البرمجي الواحد وكتابة رمازهم البرمجي كلاً بلغته البرمجية التي ألفها وتمرس فيها شرط أن تكون متوافقة مع هذه المنصة (.NET) مثل : C#,VB.net

أما بالنسبة لاستقلالية منصة التشغيل فلقد اكتمل مؤخراً مشروعين مفتوحي المصدر

مـن شـأنهما تشـغيل التطبيقـات المطـورة وفـق تقنيـة .NET. علـى أنظمـة التشـغيل الحاليـة المختلفة (Windows , Linux , Mac , iPhone) وهما مشروعي :

MONO Project : http://mono-project.com/

DotGnu Project : http://www.gnu.org/software/dotgnu/

9.1 : منصة العمل .NET والمترجم اللغات المشترك (CLR) :

إن منصـة .NET Framework تـدير وتنفـذ التطبيقـات وخـدمات الويـب وتحتـوي أيضـاً علـى مكتبـة صـفوف[2] منصـة الـدوت نـت مسـبقة الصـنع (.NET Framework Class Library: FCL) حيـث تـوفر هـذه المكتبـة الضـخمة إمكانيـات عديـدة ومتنوعـة مثـل الأمـن (security) والتشـغيل البينـي (interoperability) والكثير الكثير.

توجـد تفاصـيل توصـيف منصـة العمـل .net ضـمن مـا يسـمى توصـيف البنيـة التحتيـة لهـذه المنصـة (Common Language Infrastructure: CLI) والـذي يحتـوي علـى معلومـات حـول التخـزين (storage) وأنمـاط المعطيـات (data type) والأغـراض (object).

لقـد وصّـفت هـذه المنصـة ووحـدت المقـاييس الخاضـعة لهـا المنظمـة الأوربيـة لصـناعة الحاسـوب ECMA : http://www.ecma-international.org/publications/standards/Ecma-334.htm

وهـذا التوصـيف نسـتطيع تشـبيهه بمخطـط عمـل هـذه المنصـة والـذي يمكـن بنـاؤه علـى أي منصـة تشـغيل وفقـاً لهـذا التوصـيف وهـذا مـا تـم بالفعـل مـن قبـل المشـروعين الـذين تكلمنـا عنهمـا سـابقاً (MONO , dotgnu).

إن متـرجم اللغـات المشـترك (CLR) هـو جـزء مركـزي وأساسـي مـن أجـزاء منصـة net. وهـو مسـؤول عـن تنفيـذ البـرامج ، كمـا قلنـا سـابقاً لتنفيـذ أي برنـامج مكتـوب بلغـة عاليـة المسـتوى مثـل لغـة (C#) يجـب ترجمتـه إلـى لغـة الآلـة وتـتم هـذه الترجمـة هنـا عبـر خطوتين:

[2] الصف هو نموذج مستخدم في البرمجة الغرضية التوجه بهدف تمثيل الأشياء الموجودة في العالم الحقيقي.

الخطـــوة الأولـــى : البرنـــامج يتـــرجم بوســـاطة متـــرجم اللغـــة نفســـه (language compiler) إلـى لغـة وسـيطة (intermediate language: IL) تـــــــــدعى لغـــــــــة مايكروســـــــــوفت الوســـــــــيطة (Microsoft intermediate language: MSIL) وهـي اللغـة الوحيـدة التـي يسـتطيع أن يفهمهـا (CLR) وهـذه الخطـوة بصـراحة هـي التـي تضـمن للـدوت نـت اسـتقلالية اللغـة حيـث تتـرجم كافـة الرمـازات (source code) المكتوبـة باللغـات المتوافقـة مـع net. إلـى لغـة (MSIL) ويخـزن هـذا الملـف علـى شـكل ملـف (exe. أو dll.) ويـدعى هـذا الملـف (assembly file) علـى وحـدة الـذاكرة الثانوية ومن ثم يقوم (CLR) بدمجها مع بعضها في المشروع المتعدد اللغات.

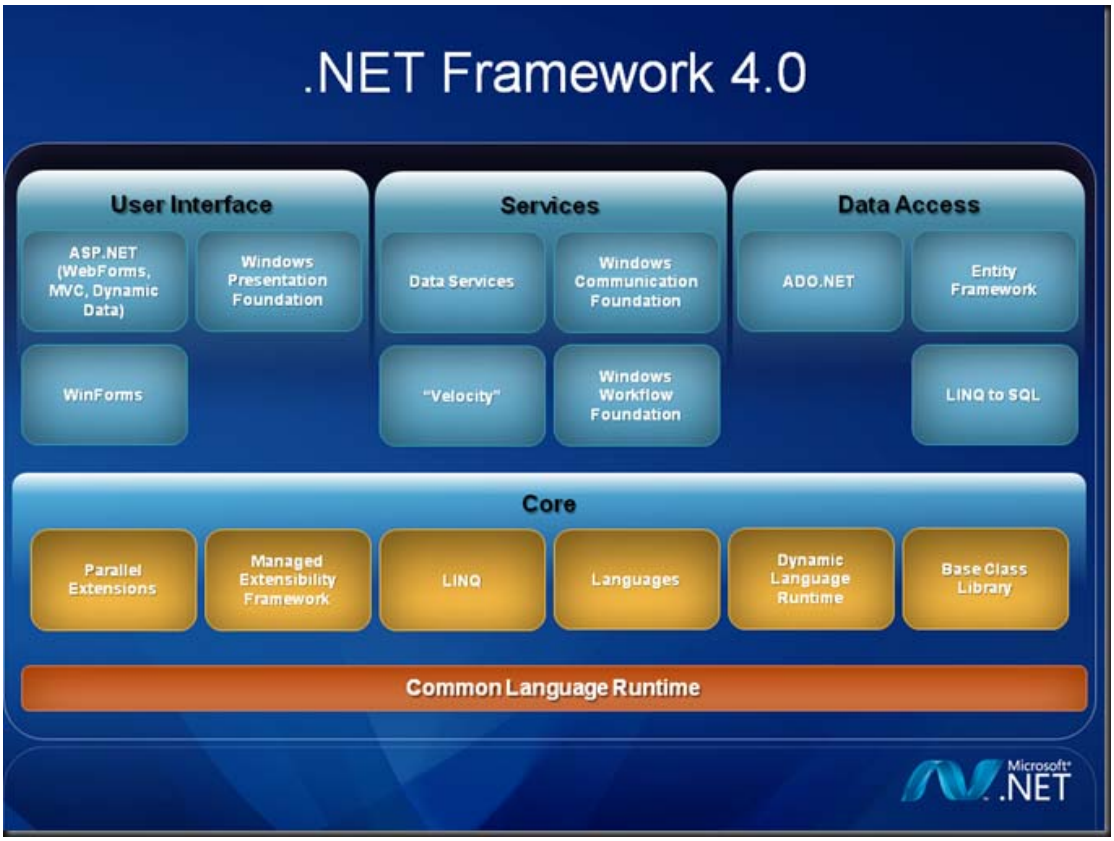
الخطـوة الثانيـة : عنـدما يـتم طلـب تنفيـذ البرنـامج يقـوم (CLR) بتحريـر متـرجم آخـر موجـود ضـمنه يـدعى المتـرجم الآنـي (.just-in-time compiler:JIT or Jitter) ويقـوم (JIT) بتحويـل البرنـامج مـن لغـة (MSIL) إلـى لغـة الآلـة ومـن ثـم ينفـذ البرنامج.

حيثمـا تكـون منصـة العمـل net. موجـودة (منصـبة) فمنصـة التشـغيل (platform) الموجـودة عليهـا قـادرة علـى تنفيـذ أي برنـامج مكتـوب وفـق هـذه التقنيـة بـدون أي تعـديل وهـذا مـا عرفنـاه سـابقاً باسـتقلالية منصـة التشـغيل ، وهـذه الاسـتقلالية مـن شـأنها تقليـل الوقـت والمـال المصـروف علـى تطـوير وتشـغيل التطبيقـات التـي يمكـن الوصـول إليهـا والتخاطـب معهـا بغـض النظـر عـن الجهـاز ومنصـة التشـغيل التـي يعمـل وفقهـا والتـي تزداد الحاجة إليها يوماً بعد يوم.

ويسـتفاد أيضـاً مـن اسـتقلالية المنصـة بتوسـيع طيـف العمـلاء الـذي سيسـتخدمون هـذا التطبيق المطور.

نـذكر أخيـراً أن مـن بعـض وظـائف متـرجم اللغـات المشـترك (CLR) إدارة الـذاكرة (memory management) ومعالجـة النياسـب المتعـددة (multithreading)[3] والكثير من المهام التي لا يتسع المجال لذكرها.

[3] النياسب المتعددة هي مفهوم يستخدم في لغات البرمجة لتطبيق مفهوم تعدد المهام (Multi-tasking) الذي تستخدمه أنظمة التشغيل للسماح بتنفيذ أكثر من عملية بشكل متزامن.


.NET Framework 4.0
User Interface
ASP.NET (WebForms, MVC, Dynamic Data)
Windows Presentation Foundation
WinForms
Services
Data Services
Windows Communication Foundation
"Velocity"
Windows Workflow Foundation
Data Access
ADO.NET
Entity Framework
LINQ to SQL
Core
Parallel Extensions
Managed Extensibility Framework
LINQ
Languages
Dynamic Language Runtime
Base Class Library
Common Language Runtime
Microsoft .NET