

اساسيات لغات البرمجة

تعريف البرنامج :

البرنامج هو عبارة عن مجموعة من التعليمات المتسلسلة والمتراطة بغرض تحقيق وظيفة أو هدف

محدد . هذه التعليمات عبارة عن جمل مركبة يعتمد شكلها (تركيبها البنائي syntax) علي

لغة البرمجة المستخدمة وكذا الكلمات المكونة للجملة تعتمد علي محلل الكلمات (lexical

Analyzer) لهذه اللغة .

وتوجد برامج للتطبيقات العامة يستخدمها الطلاب والباحثون والعاملون في كثير من الأحيان

يحتاج المستخدم حل مسأله خاصة جدا وليست عامة وهذا ما يجعله يصمم برنامجا بنفسه طبقا

لهذا الغرض الخاص .

وتعتبر البرمجيات الجزء الرئيسي من منظومة الحاسب والمكمل للمكونات المادية التي تعطي

للمستخدم إمكانيات الاستخدام الأمثل لهذه المكونات لتحقيق هدفه في كافة التطبيقات

وتنقسم البرمجيات إلى :-

١- برمجيات نظم التشغيل (Operating system program)

٢- برمجيات للتطبيقات العامة (General application Programs)

٣- برمجيات للتطبيقات الخاصة بالمستخدمين التي يصممها المستخدم لتحقيق الغرض الخاص

حل المشاكل الرياضية أو الهندسية أو الطبية أو الكيميائية أو التعليمية إلخ.

أنواع اللغات البرمجية:-

١- لغات عالية المستوى (high level language):

وهي التي تكون جملها قريبة جدا من لغة الإنسان الواضحة أي أن الأمر في هذه اللغات يكون بسيطا ويسهل فهمه ومعرفة ما سوف يقوم الحاسب بتنفيذه مثل (كرر) (Repeat) و (لو حدث كذا) (If.....)

Begin	ابداء
End	وانتهى
Read	اقراء
Input	ادخل

وغير ذلك من الكلمات والجمل بتركيب معين يعتمد علي اللغة نفسها وتحتاج هذه اللغات إلى مترجم ونظرا لان معالج الحاسب (processor) لا يفهم هذه الجمل لذلك فهي تحتاج إلى مترجم يقوم بترجمتها إلى النبضات التي تكون شفرة المعالج (Processor) .

لغة الآلة

وهي اللغة التي تفهمها الآلة وتقوم بتحويل البرنامج كلة دفعة واحده من اللغة عالية المستوي إلى لغة الآلة إذا كان هناك أي أخطاء اثناء الترجمة فإن المترجم يعطي الرسائل الخاصة بالأخطاء والتحذيرات.

ومن أمثلة هذه اللغات

١- لغة ال(C)

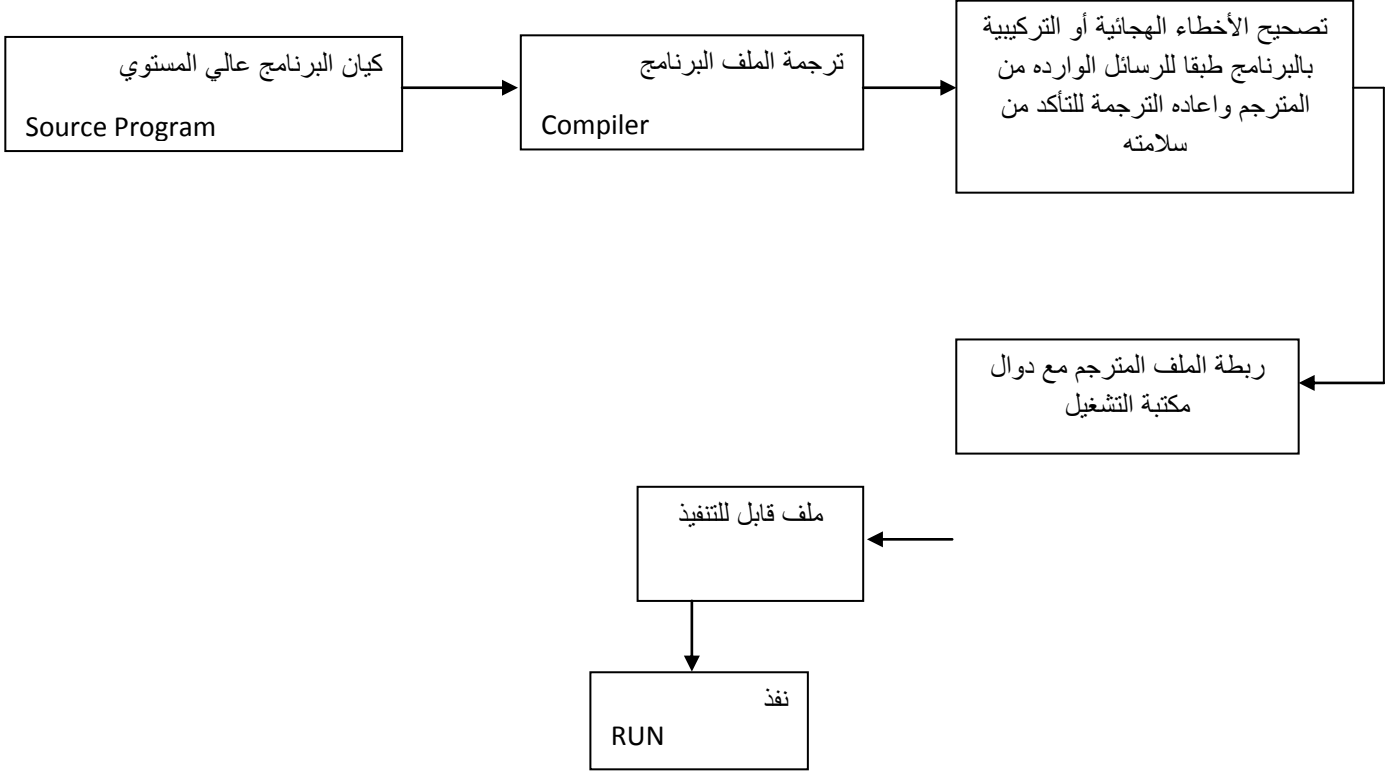
٢- لغة ال(c++)

٣- لغة ال(vc++)

٤- لغة ال(Basic)

٥- لغة ال(V Basic) ، وغيرها

ويمكن أن توضح خطوات تنفيذ البرنامج كآتي



وهناك نوع من المترجمات يقوم بترجمة البرنامج خطوة خطوة يبدأ من أول البرنامج من أعلي إلي أسفل ومن اليسار إلي اليمين وهو يسهل اكتشاف الأخطاء لكنه يستغرق وقتا كبيرا في ترجمة البرنامج

٢- لغة ذات مستوي منخفض (Low level Language)

لكل شريحة معالج (Processor IC) كلمات شفرية خاصة به يفهمها . كل كلمه تتكون من عدد من النبضات (صفر أو ١) علي حسب نوع المعالج يمكن أن تكون طول الكلمة ٨ أو ١٦ أو ٣٢ ولكل تبديله من النبضات معني معين يفهمه المعالج مثل حمل أو أسترجع أو أجمع أو قارن إلخ من العمليات البسيطة .

ويسمي التعامل المباشر بهذه التبدليات (صفر، واحد) بلغة الآله وهو بالطبع غاية في التعقيد إذا تم التعامل معه ويكبر احتمال الأخطاء وهناك لغة ذات مستوي أعلي نسبياً من لغة الآلة ولكنها من اللغات منخفضة المستوي وهي لغة التجميع وتسمي مجموعة كلماتها المكونة لها بـ (Op Code) كل كلمة منها تتكون من عدد بسيط من الأحرف مثل LD ومعناها بالتحميل أو STR ومعناها خزن وتختلف لغ التجميع من معالج لآخر ولا بد من وجود مجمع لها يقوم بترجمة (Op Code) إلي النظام الثنائي (صفر أو ١) وهو لغة الآلة كي يفهم المعالج الأوامر . ويمكن وصف كل من لغة الآلة ولغة التجميع كنوعين من اللغات منخفضة المستوي ببساطة كالآتي

لغة الآلة :-

تسمى اللغة الثنائية لأنها تتكون من كلمات ذات طول (٨ أو ١٦ أو ٣٢) طبقاً لنوع المعالج ليقيم بفهمها الحاسب الآلي ولذلك لابد من تحويل أي لغة آخر إلى هذه اللغة حتى تتمكن المعدة من تفسيرها وتنفيذ أوامرها . ومن الواضح إنها في منتهي الصعوبة في التعامل معها نظراً لما تتطلبه من دقة متابعة أثناء الكتابة لسطور كبيرة في احتمال الخطأ ويلزم أن يكون المبرمج في هذه الحالة علي دراية بتركيب الحاسب ومجموعة الأكواد الخاصة بالمعالج ومعناها والتوزيع الداخلي للذاكرة ومواقع تخزين الأوامر والبيانات المختلفة

ونظراً لأن لكل معالج لغة مختلفة عن الآخر فإن تنفيذ البرنامج علي جهاز لا يصلح علي الإطلاق في تنفيذه علي جهاز آخر ونتيجة لهذه الصعوبات كانت الدافع الرئيسي لعلم المترجمات (Compilers) للغات عالية المستوى لتنفيذ الترجمة التي كان علي المبرمج عملها وتحل مشقة صعوبة هذه المهمة وإستغراقها لوقت طويل جداً

لغة التجميع :-

ظهرت لغة التجميع بعد لغة الآلة ولكنها تتميز بأنها أسهل منها بكثير حيث إنها ليست مجموعة إصفار ووحايد وإنما تكتب من خلال مجموعة من تعبيرات بسيطة قصيرة في شكل رموز مختصرة كما أوضحنا من قبل ولتنفيذها علي الحاسب لا بد لها من مجمع (Assembles) كي يترجمها إلي لغة الآلة إلا إنها تحتاج إلي مبرمجين لهم خبرة عالية بالمعالج وطريقة برمجته والمكونات المادية داخله والملفات المسموح بربطها به.

أمثلة علي لغات البرمجة

١- لغة الكوبول

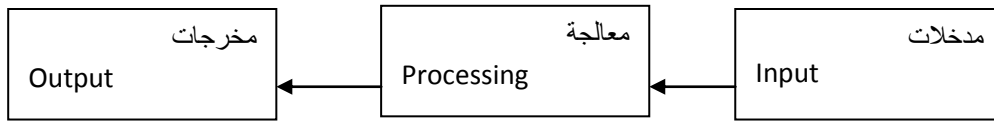
وهي إختصار للعبارة (Common Business Oriented Language) وقد طورت وصممت هذه اللغة بواسطة لجنة Codasyl إختصارا لكلمة Conference of Data System Language وقد قام المعهد القومي الأمريكي للقياس والتوحيد ANSI بتطوير نسخة معيارية من الكوبول عام ١٩٦٨ وتم تسميتها ANSI ثم ظهر بعدها ANSI1974 ، ANSI1980 وينتشر إستخدام لغة الكوبول علي نطاق واسع عالميا حيث تستخدم في البنوك وفي المنظمات الحكومية وتستخدم علي حاسبات كبيرة أو علي حاسبات شخصية وتتميز هذه اللغة بقدرتها علي التعامل مع الملفات لذا اشتهرت بإنها لغة أعمال

خطوات كتابة البرنامج

لكتابة برنامج ما لابد من إتباع خطوات منطقية متسلسلة ومتفق عليها طبقا للتوحيد القياسي العالمي وهي:-

١- تحليل المشكلة :

والمقصود بها تحديد عناصر المسألة المطلوب برمجتها بتحديد وتوصيف شكل المدخلات وعمليات المعالجة المطلوبة لهذه المدخلات (حسابية ومنطقية) ومن ثم إنتاج المخرجات وتحديد أشكالها المطلوبة وكيفية إخراجها ويمكن التعبير عن ذلك بالآتي


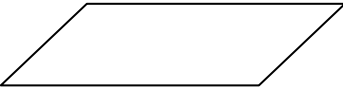

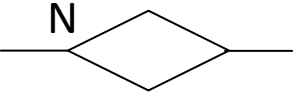





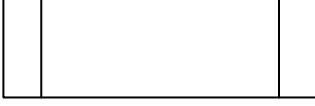
٢- وضع الحل التخطيطي :-

القيام بسرد خطوات الحل أو الحلول في شكل متسلسل مترابط واضح للوصول إلي حل المشكلة ويسمي ذلك الخواريزم (Algorithm)

٣- رسم خريطة تدفق العمليات من تفاصيل الخطوة السابقة :

يتم رسم الخطوات المتسلسلة برموز قياسية تعبر عما سبق من عمليات مطلوبة وهي البدء والمدخلات والمعالجة بعمليات مختلفة والنتائج المخرجة ثم الإنتهاء ويسمي ذلك مخطط سير العمليات وخريطة التدفق للعمليات ما هي إلا مجموعة من الأشكال أو الرموز المتفق عليها عالميا والتي تعتبر من أنجح الأساليب في تحليل النظم وتوضيحها في شكل خرائط . كل رمز يمثل وظيفة معينة في الخريطة والأشكال الآتية مرفق أمامها المعنى المستخدم لها في خرائط التدفق

الشكل	المعنى
	بدء - إنتهاء - توقف Start - End - Stop
	مدخلات - مخرجات Input -Output
	عمليات Processing
	قرار - شرط Decision
	نقط إتحال Connector Point

	حلقات تكرار For Loop
	تحديد إتجاه سير البرنامج
	إستدعاء برنامج فرع 

٤ - بعد رسم الخريطة كتابة أوامر البرنامج :

ليتمكن الحاسب من فهم الحل يتم تحويل الحل التخطيطي بعد التحليل وبناء خريطة التدفق إلى أوامر لغة برمجة لأحدي اللغات وفي هذه الحالة يسمى البرنامج **Source Program**

٥- الترجمة :

يتم ترجمة البرنامج بواسطة المترجم (Compiler) لمعرفة الأخطاء في تركيب الجمل أو الهجاء وكذا التحذيرات المنتجة بواسطة المترجم ثم تصحيحها وإعادة ترجمتها والتأكد من صحتها .

وتتم الترجمة بإعطاء أمر الترجمة للمترجم ليقوم بترجمة الملف المصدر (Source Program) وتحويله إلى لغة الآلة التي يمكن أن تفهمه في هذه الحالة ويسمي البرنامج الهدف (Object Program) ثم يتم التنفيذ للبرنامج RUN.

٦- الأختبار Test:-

بعد تنفيذ البرنامج يتم الأطلاع علي النتائج للتحقق من صحته منطقيا وذلك بإستخدام عينة من البيانات الأختبارية فإذا ثبت صحتها تكون قد حصلنا علي البرنامج السليم بصورة كاملة للتطبيق العملي علي بيانات حقسقة والمستخدمه عمليا .

٧- توثيق البرنامج :-

عمل توصيف وشرح ووضع التعليقات أمام الإجراءات والخطوات لسهولة متابعة البرنامج عند تطويره أو صيانتة بطريقة تساعد المستخدم علي فهم كيفية عمل البرنامج وحفظ البرنامج المصدر ورقيا وعلي اسطوانات احتياطية (Buck Up)

مثال توضيحي علي خطوات كتابة البرنامج

المسألة :-

أحسب كل من مساحة ومحيط الدائرة التي نصف قطرها = ١٠ سم مع العلم بأن القيمة التقريبية

$$\pi = 3,14 \text{ تقريبا}$$

خطوات الحل :

١- تحديد عناصر المشكلة :

• المدخلات

$$R = 10$$

$$\pi = 3,14$$

• العمليات :

مساحة الدائرة

$$A = \pi \times R \times R$$

محيط الدائرة

$$C = 2 \times \pi \times R$$

• المخرجات

$$A =$$

$$C =$$

٢- رسم خريطة التدفق

