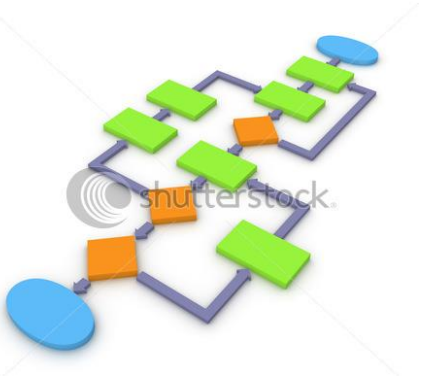


الفصل الأول: المقدمة

برنامج الحاسب

- البرنامج:

هو عبارة عن مجموعة من التعليمات تعطى للحاسب للقيام بعمل ما،

مثل : حساب مجموع قيم مختلفة ، حساب المتوسط الحسابي ، ....

البرنامج هو الذي يحدد كيفية التعامل مع البيانات للحصول على النتائج المطلوبة.

المبرمج

- المبرمج(computer programmer):

هو الذي يكتب البرنامج بعد أن يفهم المشكلة ويقترح الحل وينفذه لحل هذه المشكلة .

- ويجب أن يكون البرنامج صحيحاً وواضحاً وليس فيه غموض.

البرمجيات

- البرمجيات (Software) :

هي التي تسهل للمستخدم استخدام المكونات المادية (Hardware) بكفاءة وراحة ويمكن تقسيم البرمجيات إلى :

١) برامج التشغيل.

٢) برامج التطبيقات.

٣) لغات البرمجة.

برامج التشغيل

- برامج التشغيل (Operating System) :

هي عبارة عن برامج تقوم بدور الوسيط بين المستخدم والمكونات المادية.

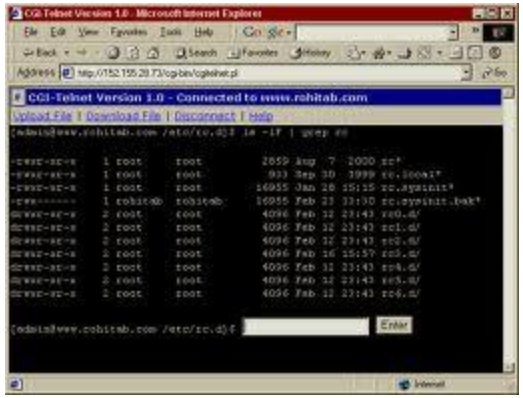
- من وظائفها :

١) تمكن المستخدم من استخدام المكونات المادية للحاسب بكفاءة وبراحة.

٢) تساعد المستخدم في إنشاء نظام الملفات وغيرها.

أمثلة لبرامج التشغيل

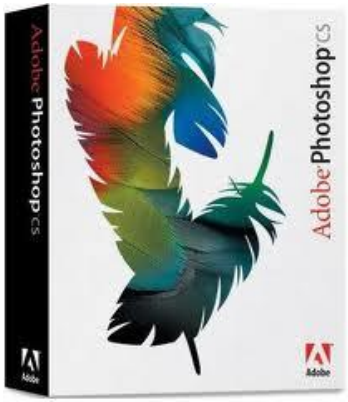
- من برامج التشغيل ما يصلح للعمل في الشبكات مثل , Unix Windows ، ومنها الذي يستخدم مع الحاسب فقط مثل Dos .

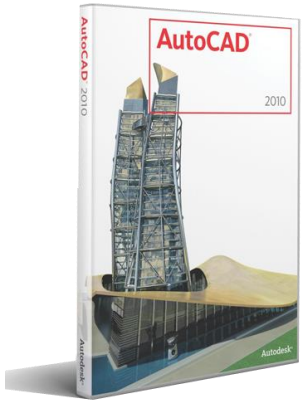


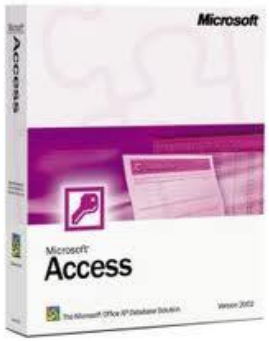
برامج التطبيقات

- برامج التطبيقات (Application Programs) :

هي برامج تساعد في إنشاء كثير من التطبيقات، مثل: إنشاء قاعدة البيانات والرسم باستخدام الحاسب وغيرها.







لغات البرمجة

- لغات البرمجة (Programming Languages) :

هي التي تستخدم في بناء البرامج المختلفة وهي تتراوح من اللغات التي تتعامل مباشرة مع المكونات المادية للحاسب والأخرى التي تتطلب تحويلها من صورتها التي تكتب بها إلى صورة أخرى يستطيع الحاسب التعامل معها.

- تقسم لغات البرمجة إلى:

١) لغة الآلة.

٢) لغات التجميع.

٣) لغات المستوى العالي

لغة الآلة

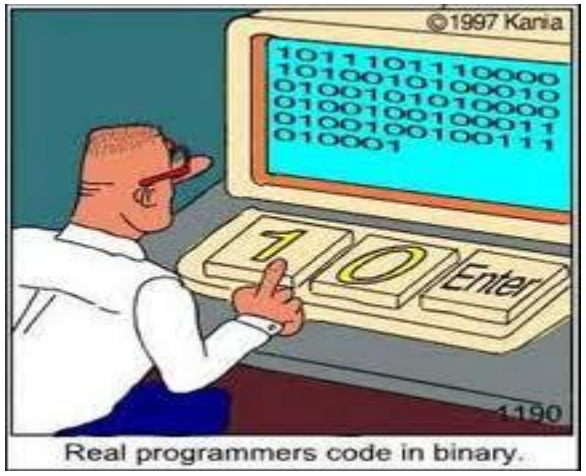
- لغة الآلة (Machine Languages) :

هي اللغة الوحيدة التي يفهمها الحاسب ويستطيع التعامل معها، وهي تعتبر لغة خاصة لكل حاسب وقد تختلف من حاسب لآخر لأنها تعتمد على المكونات المادية للحاسب نفسه.

- لغة الآلة تتكون من مجموعة أرقام من بين ١، ٠ التي تعطي تعليمات للحاسب للقيام بمعظم العمليات الأساسية واحدة بعد الأخر

لغة الآلة

- لغة الآلة من اللغات الصعبة في التعلم للإنسان حتى بالنسبة للمبرمجين لأنها مجموعة من الأرقام ١، ٠ فقط.
- للتغلب على هذه الصعوبة تم اقتراح لغة أخرى تعتمد على استخدام اختصارات معبرة من اللغة الانجليزية للتعبير عن العمليات الأولية التي يقوم بها الحاسب وهذه اللغة هي لغة التجميع.



لغة التجميع

- لغة التجميع (Assembly Languages) :

هي لغة تستخدم اختصارات معبرة من اللغة الانجليزية لتعبر بها عن العمليات الأولية التي يقوم بها الحاسب ، مثل إضافة (Add) وحفظ (Store) و طرح(Sub) وغيرها.

```
i = j + k;                1    ILOAD j    // i = j + k     0x15 0x02
if (i == 3)               2    ILOAD k                    0x15 0x03
   k = 0;                 3    IADD                       0x60
else                      4    ISTORE i                   0x36 0x01
   j = j - 1;             5    ILOAD i    // if (i < 3)    0x15 0x01
                          6    BIPUSH 3                   0x10 0x03
                          7    IF_ICMPEQ L1               0x9F 0x00 0x0D
                          8    ILOAD j    // j = j - 1     0x15 0x02
                          9    BIPUSH 1                   0x10 0x01
                         10    ISUB                       0x64
                         11    ISTORE j                   0x36 0x02
                         12    GOTO L2                    0xA7 0x00 0x07
                         13 L1:          BIPUSH 0         // k = 0 0x10 0x00
                         14    ISTORE k                   0x36 0x03
                         15 L2:
        (a)                          (b)                        (c)
```

لغة التجميع

- نظراً لان هذه اللغة تستخدم كلمات مختصرة من اللغة الانجليزية فإنها تحتاج محولاً لكي يحولها إلى لغة الآلة وهو ما يسمى المجمع (assembler) الذي يقوم بتحويل لغة التجميع إلى لغة الآلة كي يفهمها الحاسب ويستطيع تنفيذها.
- ولكن بالرغم من كل ذلك ولكن مازال هناك توجد مشقة عند حل ابسط المسائل لان ذلك يتطلب معرفة وكتابة العديد من التعليمات، وهذا ما دفع المبرمجين للتفكير في لغات أخرى تقلل المجهود الكبير اللازم لكتابة الكثير من التعليمات فكانت لغات البرمجة ذات المستوى العالي.

لغات البرمجة ذات المستوى العالي

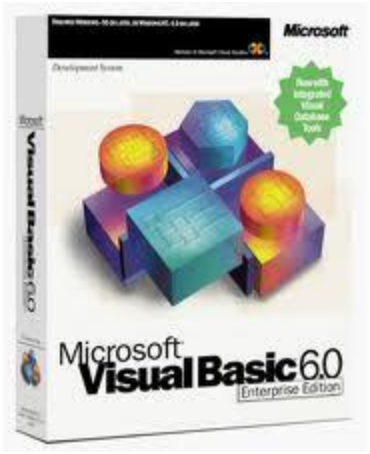
- لغة البرمجة ذات المستوى العالي (High level Languages) :

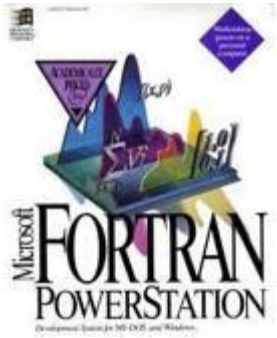
هذه اللغات كتبت بحيث تستخدم بعض الكلمات الانجليزية العادية بنفس معانيها حيث يقوم كل أمر منها بتنفيذ العديد من الواجبات ، وهذه اللغه كسابقاتها تحتاج الى (مترجماتcompilers) التي تقوم بتحويل التعليمات (الأوامر) إلى لغة الآلة ، وهذه اللغات تستخدم العلاقات والعوامل الرياضية المتعارف عليها مثال:

Sum = A + B +C

لغات البرمجة ذات المستوى العالي

- هذه اللغات تعتبر سهلة ومرغوبة من وجهة نظر المبرمجين بالمقارنة مع لغة التجميع ولغة الآلة





لغات البرمجة ذات المستوى العالي

- من المعلوم إن عملية تحويل البرنامج من لغة ذات مستوى عال إلى لغة الآلة تستهلك وقتاً ولذلك تم تطوير نسخ من لغات المستوى العالي بحيث تستخدم برنامج مفسر(interpreter)وهو يقوم بترجمة الكود سطراً سطراً أثناء التنفيذ.

المترجم compiler	المفسر interpreter
سريع	بطئ
يترجم البرنامج كاملاً	يترجم البرنامج سطراً سطراً
صعب التعديل عليه ومن ثم اعاده الترجمة لأنه يعمل على البرنامج كاملاً	سهولة التعديل عليه (حذف ،إضافة ، تغيير أو تصحيح)
صعوبة اكتشاف مكان الأخطاء في بعض الأحيان	سهولة اكتشاف مكان الأخطاء

أهمية مهنة البرمجة

- في الوقت الحالي لا يمكننا الاستغناء عن المبرمجون لأن دورهم مهم وحيوي وتكثر الحاجة لهم في شتى المجالات وذلك لعمل الآتي:

١) كتابة برامج وبناء الأنظمة المختلفة لحل المشاكل وتبسيط التعامل مع الحاسب.
٢) المسئولية الكاملة عن إصلاح ما يحدث من أعطال أو حل المشاكل التي تحدث في الأنظمة المختلفة.
٣) بناء واجهة المستخدم المختلفة في كثير من اللغات والتطبيقات.
٤) بناء نظم التشغيل المختلفة مثل Windows, Unix وغيرها من النظم. فمثلاً تستخدم لغة C في بناء نظام التشغيل Unix .
٥) برامج المواجهة المختلفة في الأنظمة المختلطة الرقمية والتماثلية.

صناعة البرمجيات

- تعتبر صناعة البرمجيات في عصرنا الحالي من الصناعات المهمة جداً والتي تتطور باستمرار نتيجة التطور الهائل في صناعة الحاسبات الآلية ، لذلك فإن هذه الصناعة تتطلب مبرمجين مهرة ولديهم القدرة على تحليل وحل المشاكل بالإضافة إلى إلمام بكل المستجدات والعلوم والتطوير المتعلق بالحاسب وصناعة الحاسبات وذلك حتى يستطيعوا مواكبة تطوير البرامج والنظم المختلفة للاستفادة العظمى من التقدم في الحاسبات.

تصميم برامج الحاسبات

حل المشكلات

سبب نشأة البرامج

- سبب نشأة البرامج :

البرامج تنشأ بسبب وجود مشكلة لدى المستخدم ويحتاج لبرنامج يساعده.

- تصميم البرمجيات ينطلق من الخوارزميات والمخططات التدفقية حيث يشكلان بداية الطريق لخلق أي برنامج.

المراحل الأساسية لحل المشكلة

- لحل أي مشكلة لابد من المرور بخمس مراحل:

١) تعريف وتحليل المشكلة.
   - فهم المشكلة.
   - تقسيم المشكلة.
   - عملية حل المشكلة
٢) وضع الحل التخطيطي.
٣) كتابة الكود البرمجي.
٤) ترجمة البرنامج إلى لغة الآلة.
٥) تنفيذ البرنامج وتجربته.

المرحلة الأولى: تعريف وتحليل المشكلة (فهم المشكلة)

- في هذه المرحلة نفهم المشكلة العارضة لدينا ونستنتج الطرق التي تناسب حل المشكلة وإمكانية تطبيقها.
- في هذه المرحلة يجب تحديد ما يلي:

١) طبيعة النتائج المستخرجة(Output) والتي احتاجها من برنامجي.

٢) معرفة المدخلات(Input) والمعطيات المطلوب إدخالها للبرنامج.

٣) طرق الحل المناسبة وتقييمها بما يتلاءم مع طريقة تنفيذها وفي ضوء ذلك نختار الأفضل .

المرحلة الأولى: تعريف وتحليل المشكلة (فهم المشكلة)

- فهم المشكلة وهي الخطوة الرئيسية للشروع في خلق أي برنامج ومنها يتم تنفيذ وفهم باقي الخطوات.
- هناك بعض القواعد التي ذكرها الفيلسوف رين ديكارت التي تساعد في حل المشكلة:

١) لا يمكن قبول أي شي إلا بالتجربة والمشاهدة.

٢) كل مشكلة يمكن تبسيطها وتجزيئها إلى أجزاء عدة.

٣) دائمًا نبدأ بالأجزاء السهلة البسيطة ومنها إلى الأصعب.

٤) المراجعة لجميع الأجزاء ليكتمل الحل.

المرحلة الأولى: تعريف وتحليل المشكلة (فهم المشكلة)

- المشاكل دائماً تظهر أكثر تعقيداً من الحقيقة بسبب عدم فهم المشكلة بالشكل الصحيح.
- وهنا نحصل على:

القاعدة الأولى: وهي تحليل المشكلة بعناية فائقة وفهم كل جزئياتها والتي نلخصها بقول أن فهم المشكلة يمثل نصف الحل.

- في هذه الجزئية المفروض نعرف الأهداف المطلوبة ، والوسائل اللازمة لتحقيق الحل الصحيح.

المرحلة الأولى: تعريف وتحليل المشكلة (تقسيم المشكلة)

- مع زيادة فهمنا للمشكلة يزداد تبعاً له وضوح تفصيلات وأبعاد أخرى للمشكلة، فتصبح المشكلة أكثر ثباتاً ووضوحاً وتفصيلاً مما يجعل من الصعب التعامل كل هذه التفاصيل في نفس الوقت.
- القاعدة الثانية :

حاول تقسم المشكلة إلى أجزاء بسيطة وغير معتمدة على بعضها البعض ثم ركز على كل جزء على حده.

ومن هذا الأسلوب أستطيع إتباع طرق مختلفة للتقسيم لذلك نجد فروع لهذه القاعدة .

المرحلة الأولى: تعريف وتحليل المشكلة (تقسيم المشكلة)

- قاعدة ٢ (أ):

حاول تقسيم المشكلة إلى مجموعة مشاكل (أجزاء) بسيطة متتابعة، حتى نحصل على الحل الكامل للمشكلة الأصلية بحل المشاكل الفرعية البسيطة الواحدة تلو الأخرى.

- قاعدة ٢ (ب):

إذا كانت العملية تتضمن بعض العمليات التي يعاد تكرارها حاول عزل التي تطلب إعادة من التي لا تتطلب الإعادة.

المرحلة الأولى: تعريف وتحليل المشكلة (تقسيم المشكلة)

- قاعدة ٢ (ج):

حاول في البداية إيجاد حل للمشاكل في الحالات البسيطة أو المشهورة وعند الوصول إلى حل مرض وصحيح يمكن تطوير هذا الحل ليشمل الحالات الخاصة والمعقدة ،بحيث نبدأ بالتعامل مع الحالات البسيطة فالأصعب والأصعب.

المرحلة الأولى: تعريف وتحليل المشكلة (عملية حل المشكلة)

- القاعدة الثالثة:

عند تقسيم المشكلة إلى أقسام صغيرة يجب أن يكون التقسيم على خطوات متعددة بحيث تستخدم القواعد العامة في المراحل الأولى ثم يتم الانتقال إلى المراحل الخاصة بعد ذلك.

- المراحل الأولى في الحل تتطلب اعتبارات عامة وواسعة بينما المراحل المتأخرة تتطلب التركيز على التفاصيل والانتقال من العام إلى الخاص يعرف بطريقة من الأعلى إلى الأسفل (Top Down Design)

المرحلة الأولى: تعريف وتحليل المشكلة (عملية حل المشكلة)

- يقترح أن لا يتجاوز عدد الأجزاء المقسمة في كل خطوة ٥ أجزاء.
- القاعدة الأساسية في عملية التقسيم:

١) أن يستمر التقسيم حتى يمكن عزل الأجزاء عن بعضها البعض.

٢) أن يكون حل هذه الأجزاء سهلاً.

- عملية التقسيم تتطلب مهارة وخبرة يتم تنميتها واكتسابها مع الوقت .

المرحلة الأولى: تعريف وتحليل المشكلة (عملية حل المشكلة)

- القاعدة الرابعة:

في كل مرحلة من المراحل يجب مراجعة الحل المقترح ليتم التأكد من انه كامل وصحيح.

- تتم المراجعة لكل فرع أو جزء على حده ومن ثم على التوافق بين الفروع كلها والتأكد من أنها تحقق المطلوب وأنها تأخذ في الحسبان كل الحالات الخاصة.

المرحلة الثانية: وضع الحل التخطيطي

- في هذه المرحلة نقوم هنا بالتعبير عن الحل التي استنتجت سابقاً على شكل خطوات متسلسلة ومترابطة منطقياً للوصول إلى الحل وهي ما تدعى بالخوارزمية .

- وبعد ذلك نقوم بوضع هذه الخوارزمية في مخطط بياني مستخدمين مجموعة من الأشكال والرموز ونكون بذلك حصلنا على المخطط التدفقي (مخطط سير العمليات أو المخطط المنهجي)

المرحلة الثالثة: كتابة الكود البرمجي

- لكي يفهم الحاسب الحل المقترح ويتم تنفيذه يجب تحويله إلى لغة يفهمها فيتم تحويله إلى كود برمجي باستخدام لغات البرمجة المعروفة ويسمى الحل المقترح هنا بالبرنامج المصدر

المرحلة الرابعة: ترجمة البرنامج المصدري

- في هذه المرحلة يتم إدخال البرنامج إلى الحاسب وترجمته إلى لغة الآلة بواسطة برنامج الترجمة الخاص بلغة البرمجة المستخدمة.
- تمر عملية الترجمة بالمراحل التالية:

١) مرحلة التحليل المعجمي (مطابقة مفردات)

٢) مرحلة التحليل اللغوي والنحوي (مطابقة تعليمات)

٣) مرحلة ترجمة البرنامج إلى لغة الآلة

المرحلة الخامسة: تنفيذ البرنامج وتجربته

- في هذه المرحلة يتم تجربة البرنامج للتأكد من صحته، باستخدام عينة من البيانات الاختبارية .

مراحل حل المشكلة

- بعد أن استعرضنا خطوات التفكير لحل أيه مسألة برمجيه وقبل أن ندخل في تفاصيل كتابة حل المشكلة نقول أن الحل يمر بمرحلتين :
- المرحلة الأولى: هذه المرحلة تمثل دور الإنسان في حل المشكلة :

١)تحديد معالم المشكلة

٢)تحليل عناصرها

٣)البحث والتفكير في طريقة حل المسأله

٤)تدوين الحل في خطوات متسلسله متعاقبه ، يعبر عنا باللغة العادية محكومة بالمنطق الرياضي . هذه الخطوات في مجموعها تسمى بالخوارزم Algorithmوتمثيل هذه الخطوات بخريطة التدفق لتساعد في تسلسل المنطق العام حل المشكلة.

٥)كتابة البرنامج.

مراحل حل المشكلة

- المرحلة الثانية: وهذه المرحلة تمثل دور الحاسب نفسه في حل المشكلة:
  ١) ترجمة البرنامج.
  ٢) التحقق من خلوه من الأخطاء.

مراحل حل المشكلة باستخدام الحاسب