الجامعة التقنية الشمالية

المعهد التقني / نينوى

حقيبة تعليمية لمادة الخوارزميات

قسم الانظمة الحاسوب

المرحلة الاولى

❖ مقدمة:

تظهر الضرورة لعمل برنامج الحاسوب الالكتروني عند ظهور مشكلة لها صفة التكرار مثل (حساب رواتب الموظفين ، إدارة المخازن ، استخراج نتائج الطلاب ، .....إلخ) مما يساعد على توفير الوقت والجهد من الأعمال الروتينية التكرارية، أو عند الرغبة في حساب بعض العمليات الرياضية المعقدة والتي تتطلب دقة وسرعة للحصول على نتائجها مثل(حساب المساحات أو الأحجام لبعض الأشكال الهندسية وتحديد مسارات سفن الفضاء والصواريخ .....إلخ).

❖ البرنامج (Program):

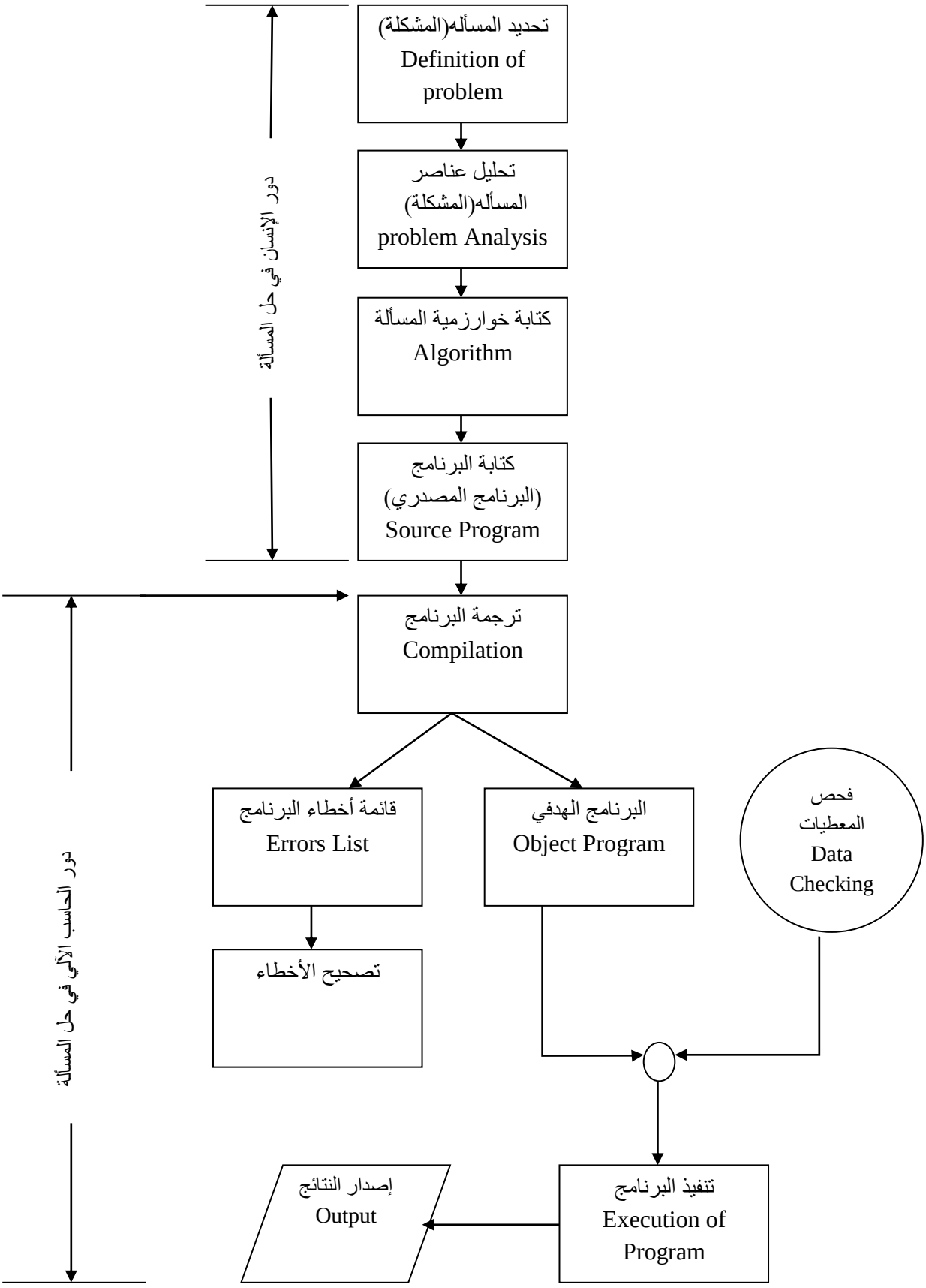
هو عبارة عن مجموعة من الأوامر (Instructions) المرتبة منطقيا التي تخبر الحاسوب بما يجب أن يقوم به للحصول على النتائج المطلوبة.

وتتمثل وظيفة المبرمج (Programmer) الأساسية في كتابة البرامج التي توجه عمل الحاسوب. ويجب أن تكون هذه البرامج صحيحة وواضحة وقادرة على إنتاج معلومات ذات جدوى تخدم المستخدم النهائي.

وتنقسم المراحل التي يمر بها برنامج الحاسب الآلي إلى مرحلتين رئيسيتين هما دور الانسان في حل المسألة، ثم يليها دور الحاسوب في الحل.

وبالرغم من أن الحاسوب الالكتروني يتميز بقدرته على إنجاز العمليات الحسابية حسب الأوامر و التعليمات المعطاة له بسرعة فائقة و بدقة متناهية و كذلك بإمكانياته الكبيرة في حفظ المعلومات الواسعة و المختلفة التي يعجز الإنسان عن حفظها و استعادتها باستعمال ذاكرته العادية. فهو يعجز عن أن يقوم بشكل ذاتي بحل أي مسألة مهما كانت بسيطة، أي أن عمله ينحصر في إنجاز الحلول للمسائل التي تبرمج له بشكل صحيح يتوافق مع الأسس العلمية الصحيحة التي تعتد عليها هذه الحلول. لذا سوف نستعرض الخطوات الضرورية اللازمة لحل المسائل باستخدام الحاسوب الالكتروني وكذلك توضيحاً مفصلاً لمفهوم الخوارزميات و المخططات الانسيابية التي تشكل العنصر الأساسي لكيفية البرمجة.

الشكل التالي يوضح المراحل المختلفة التي يمر بها البرنامج:



الشكل رقم(١) يوضح المراحل المختلفة التي يمر بها البرنامج.

وعند استخدام الحاسوب الالكتروني في حل مسألة ما (مشكلة) ، فإن هناك عدداً من الخطوات التي ينبغي إتباعها، وفقاً لدرجة كفاءة تنفيذ هذه الخطوات، تتحدد كفاءة تشغيل المسألة على الحاسوب. والجدير بالذكر أن أهم هذه الخطوات يتم تنفيذها وإنجازها خارج الحاسوب وبدون استخدامه إذ أنها تمثل منطق حل المسألة، وفيما يأتي عرض لهذه الخطوات حسب ترتيبها المنطقي (خطوات حل المسألة):

١- تعريف المسألة وتحليلها.	**Problem Definition & Analysis**
٢- وضع خوارزمية الحل.	**Algorithm**
٣- كتابة البرنامج بإحدى لغات الحاسوب.	**Writing the Program**
٤- ترجمة البرنامج إلى لغة الآلة.	**Compilation**
٥- تنفيذ البرنامج.	**Execution**

١- تعريف المسألة وتحليلها:

إن تعريف المسألة هو عبارة عن دقة التعبير في تطبيق المسألة بحيث يجعلها معروفة ومفهومة بصورة واضحة وبدون أي غموض لجميع الأشخاص العاملين ضمن مجال الاختصاص الذي تخضع له المسألة.

أما تحليل المسألة ووضع طريقة الحل فهو أصعب المصاعب و أشق الخطوات، و من أجل الوصول إلى حل صحيح فإن كثير من القوانين والطرق الرياضية المناسبة لحل المسألة يجب أن تستعمل و لربما تحتاج أيضاً إلى تطوير هذه القوانين والطرق لنجعلها تناسب الحل في كثير من الأحيان ففي هذه المرحلة يتم تحديد أبعاد المسألة، وتحديد الهدف المطلوب الوصول إليه وذلك بتحليل مفردات المسألة ووصفها ومن ثم إنجاز المهام التالية:

١- تعريف المخرجات، وشكلها بدقة ... ونقصد بالمخرجات هنا، النتائج المراد تحقيقها من حل المسألة، وينبغي هنا أن نوضح أن المخرجات يجب أن يتم تعريفها أولاً لكي يمكن تحديد المدخلات والعمليات اللازمة لتحقيقها؛ فالنتائج تعرف وتحدد أولاً ونحصل عليها أخيراً.
٢- بناءً على المخرجات والنتائج المطلوبة يمكن تحديد المدخلات اللازمة للحصول على هذه المخرجات، وذلك من بيانات ومتغيرات وكذلك تحديد شكلها ومواصفاتها بدقة.
٣- حصر طرق الحل المختلفة من وجهة نظر الحاسوب، وتقييمها لاختيار أفضلها. ذلك أن أي مسألة قد يكون لها أكثر من طريقة للحل، ومن ثم يجب دراسة هذه الطرق واختيار

أكثرها ملائمة للتنفيذ باستخدام الحاسوب من حيث سهولتها، وسرعة تنفيذها، والمساحة التي تحتاجها من ذاكرة الحاسب.

والشكل الآتي يوضح العلاقة بين النقاط السابقة واللازمة لتعريف المسألة:

٢- وضع خوارزمية الحل:

بعد اختيار طريقة الحل المثالية و تحديد كل ما تشمله من علاقات رياضية، يتم التعبير عنها على شكل خطوات متسلسلة ومترابطة منطقياً، دقيقة الوصف تؤدي إلى الوصول إلى حل المسألة. وهذه الخطوات المتسلسلة تعرف بخوارزمية المسألة Algorithm of the Problem.

وتسمى هذه الخطوات بالخوارزمية Algorithm - وذلك نسبةً إلى العالم المشهور أبو عبد الله محمد بن موسى الخوارزمي القرطبلي عالم مسلم عراقي يكنى باسم الخوارزمي وأبو جعفر، يعتبر من أوائل علماء الرياضيات المسلمين، حيث ساهمت أعماله بدور كبير في تقدم الرياضيات في عصره. وهو الذي قام بوضع أسس حل المسائل بشكل تتابعي. – ويمكن تعريف الخوارزمية كالآتي:

" مجموعة متسلسلة من الخطوات اللازمة تحدد الأسلوب المستخدم لحل مسألة معينة"

و يمكن التعبير عن الخوارزمية السابقة بالرسم بدلاً من الكتابة وذلك باستخدام رموز وأشكال اصطلاحية عملاً بالحكمة الصينية القائلة: **"الصورة تغني عن ألف كلمة"** – ويطلق على الرسم التوضيحي الناتج باسم **"المخطط الانسيابي"**.

ويمكن **تعريف المخطط الانسيابي** على أنه رسم توضيحي يبين التسلسل المنطقي لسير العمليات اللازمة لحل مسألة محددة وذلك باستخدام رموز وأشكال هندسية متفق عليها حيث تصف هذه الرموز والأشكال العمليات المحددة.

٣- كتابة البرنامج بإحدى لغات الحاسوب:

إن المخطط الانسيابي هو عبارة عن تخطيط تصوري مساعد سهل الملاحظة بالنسبة للمبرمج و لكنه غير مفهوم عند الحاسوب الالكتروني، لذلك فإن طريقة الحل الممثلة بمخطط سير العمليات يجب أن تكتب بإحدى لغات الحاسوب التي يفهمها و التي تتلاءم مع حل المسألة. و يلي ذلك كتابة البرنامج على الوسط الخارجي المناسب و إدخال البرنامج إلى الحاسوب و البرنامج الناتج من هذه الخطوة و المكتوبة بإحدى اللغات الرمزية أو الراقية يسمى بالبرنامج المصدري Source Program.

٤- ترجمة البرنامج إلى لغة الآلة:

بعد الانتهاء من كتابة البرنامج المصدري يتعين إدخاله إلى الحاسوب للتأكد من صحة كتابته من جهة، ثم لترجمته إلى لغة الآلة بواسطة برنامج الترجمة الخاص في حالة عدم وجود أخطاء في البرنامج المصدري. و تمر عملية الترجمة في المراحل الآتية:

١- **مرحلة التحليل المعجمي Lexical analysis**:
في هذه المرحلة يتم مطابقة مفردات برنامج المصدر والعلاقات و الأسماء مع تلك المسموح بها في اللغة و اكتشاف أي أخطاء فيها.

٢- **مرحلة التحليل اللغوي والنحوي Syntax analysis:**
في هذه المرحلة تجري عملية مطابقة تعليمات البرنامج المصدري مع القواعد اللغوية المستخدمة، و اكتشاف أي أخطاء فيها، بالإضافة إلى عملية تحويل البرنامج المصدري إلى تعليمات و أوامر رمزية بلغة التجميع.

٣- **مرحلة ترجمة البرنامج إلى لغة الآلة:**
في هذه المرحلة نحصل على البرنامج الهدف Object Program و الذي بموجبه يمكن البدء في عملية التنفيذ.

٥- تنفيذ البرنامج.

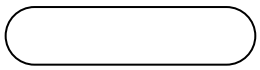
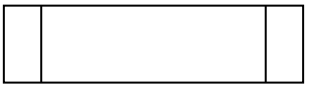
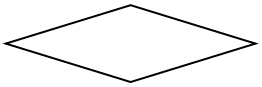
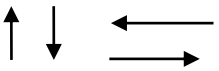
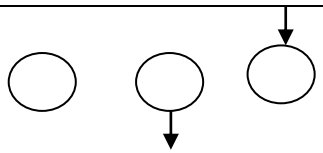
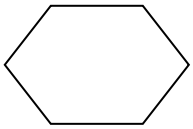
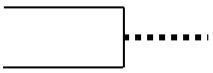
بعد الحصول على البرنامج الهدف، تتم تجربته للتأكد من صحته منطقياً وذلك باستخدام عينة من المعطيات الاختبارية Test Data فإذا ثبت صحة طريقة الحل بمطابقة النتائج الخارجة من الحاسوب مع النتائج التي تم الحصول عليها يدوياً على سبيل المثال، يمكن تنفيذ البرنامج على المعطيات الحقيقية.

إذن **الخوارزمية** هي عبارة عن مجموعة من الخطوات المتسلسلة التي تصف بصورة مضبوطة وبدون أي غموض جميع الخطوات الرياضية والمنطقية اللازمة لحل مسألة ما. ولكن هذا الوصف في كثير من الأحيان يكون معقداً وصعب الملاحظة والتتبع لذلك فإن **المخطط الانسيابي** الذي يمثل وصفاً تصويرياً لخطوات الخوارزمية يكون أكثر وضوحاً. والمخطط الانسيابي يقوم مقام الخوارزمية ويمكن بواسطته ملاحظة تتبع التسلسل المنطقي لحل المسألة بكل سهولة، وغالباً ما تكون استخراج الخوارزمية من المخطط الانسيابي أسهل بكثير من كتابة الخوارزمية مباشرة.

خصائص الخوارزمية :

١- يجب ان يكون لها بداية ونهاية.
٢- يجب ان تكون كل خطوة من خطوات الخوارزمية واضحة وبدون أي غموض عن العملية المقصودة من تلك الخطوات.
٣- يجب أن تكون خطوات الخوارزمية متسلسلة ومحددة بعددها بحيث تشكل وحدة متكاملة تؤدي بمجموعها إلى إنجاز عمل معين أو التوصل إلى نتيجة أو نهاية .
٤- يجب أن تكون الخوارزمية كاملة بحيث تأخذ بنظر الاعتبار جميع الظروف والاحتمالات التي يمكن أن تجابه طريق التنفيذ .

وعند رسم المخطط الانسيابي لمسألة معينة فإننا نستخدم مجموعة من الأشكال الرمزية الاصطلاحية المبينة في الجدول التالي:

الشكل الاصطلاحي(الرمز)	معنى الرمز
	يستخدم هذا الرمز للدلالة على بداية البرنامج ونهايته (Start / END)
	يمثل هذا الرمز كل من عمليتي الإدخال (قراءة البيانات تمهيدا لمعالجتها) والإخراج (عرض النتائج على الشاشة، طباعة، ...) (Input / Output)
	رمز المعالجة، وقد يحتوي هذا الرمز على عملية حسابية أو عملية تخزين (Calculate and Store)
	عملية استدعاء (نداء) لبرنامج فرعي (Call Sub-Routine)
	رمز اتخاذ قرار، ويستخدم هذا الرمز في خطوات المعالجة التي تتطلب إجراء عملية منطقية كالمقارنة أو عملية اختبار Decision
	اتجاه سير البرنامج. تستخدم الأسهم لبيان حركة واتجاه خريطة التدفق. Directions
	نقطة توصيل وربط (Connector).
	تكرار أو دوران (Looping)
	تعليق وإيضاح (Comment)

جدول رقم(١): اشكال الرموز في المخطط الانسيابي.

من أهم فوائد استخدام المخطط الانسيابي قبل كتابة البرنامج لمسألة ما، التالي:

١- يمكن المبرمج من الإلمام الكامل بالمسألة المراد حلها و السيطرة على كل أجزائها بحيث يساعده على اكتشاف الأخطاء المنطقية (Logic Error) و التي تعتبر من أهم الأخطاء التي تجهد المبرمج.
٢- يساعد بيسر و سهولة على تعديل البرامج الموضوعة بمجرد النظر.
٣- يعتبر الاحتفاظ برسوم المخطط الانسيابي لحلول مسائل معينة أمراً مهماً إذ يكون مرجعاً عند إجراء تعديلات عليها أو استخدامها لحل مسائل أخرى مشابهة دون الحاجة إلى الرجوع إلى المبرمج الأول باعتبار أن الحلول الأولى قد صيغت في خطوات واضحة بسيطة و مفهومة.
٤- توفير وسيلة مناسبة ومساعدة في كتابة البرامج ذات التفرعات الكثيرة.

إذن المخططات الانسيابية تستخدم:

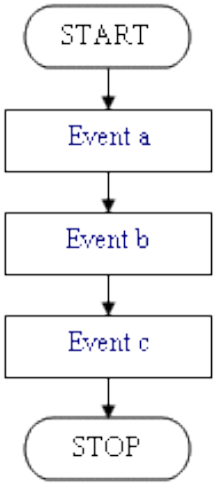
١- لتوضيح سير العمليات .
٢- تسهل على المبرمج إيجاد وتتبع الأخطاء في الخوارزمية .
٣- تسهيل عملية مراجعة البرنامج لتعديله أو لاكتشاف الأخطاء الموجودة فيه.
٤- تسهيل عملية دراسة البرنامج من قبل المستفيدين.
٥- يستخدم كوسيلة لتوثيق البرامج حيث يعكس المخطط كافة العمليات من إدخال، إخراج ومعالجة الممثلة في البرنامج.

هذا و يمكن تصنيف المخطط الانسيابي بما يلي:

١- المخطط الانسيابي ذو التتابع البسيط (Simple sequential Flowchart).
٢- المخطط الانسيابي المتفرع (Branched Flowchart).
٣- المخطط الانسيابي ذو التكرار الواحد (simple-Loop Flowchart).
٤- المخطط الانسيابي ذو التكرارات المتعددة (Multi-Loop Flowchart).

و يمكن للبرنامج الواحد أن يشتمل على أكثر من نوع واحد من هذه الأنواع. و سنتناول فيما يأتي شرح هذه الأنواع بشيء من التفصيل.

١- **المخطط الانسيابي ذو التتابع البسيط:** ويتم ترتيب خطوات الحل لهذا النوع بشكل سلسلة مستقيمة من بداية البرنامج حتى نهايته، بحيث تنعدم فيها أية تفرعات Branches و التكرارات loops و يكون الشكل العام لهذا النوع كما هو مبين في الشكل رقم(١).



الشكل رقم(١) :المخطط الانسيابي ذو التتابع البسيط.

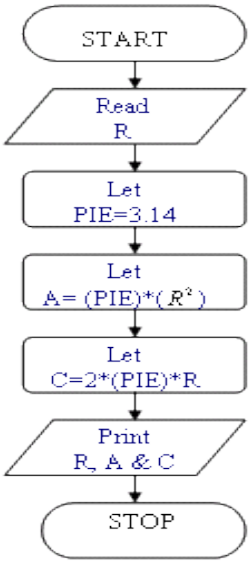
و كلمة **Event** الواردة في الشكل رقم(١) تعني الحدث أو العملية المطلوب تنفيذها.

مثال : أرسم المخطط الانسيابي لإيجاد مساحة و محيط دائرة نصف قطرها معلوم R.

Write an Algorithm and Draw a Flowchart to calculate circumference and area of circle and display result?

مساحة الدائرة = πR^2
محيط الدائرة = $2\pi R$
حيث π = النسبة التقريبية

وقيمتها العددية ثابتة تساوي ٣.١٤ بينما R متغير. تكون خطوات الحل كما مبينة في الشكل رقم(٢):



الشكل رقم(٢).

١.ابدأ.

٢.اقرأ قيمةR .

٣. ضع قيمة PIE=٣.١٤

٤. احسب المساحة(A) من المعادلة A =(PIE)*R*R.

٥. احسب المحيط (C) من المعادلة C =٢*(PIE)*R.

٦. اطبع قيم كل من C, A, R.

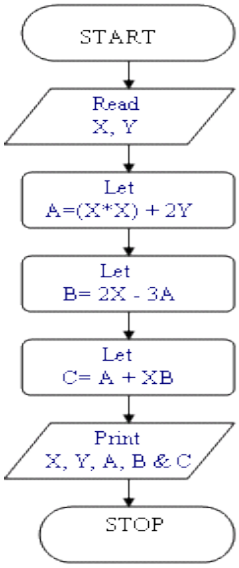
٧. توقف.

مثال: ارسم مخططا انسيابيا لحساب قيمة كل من المتغيرات C, B, A في المعادلة الآتية:

$$A = X^2 + 2Y \ldots (1)$$
$$B = 2X - 3A \ldots (2)$$
$$C = A^2 + XB \ldots (3)$$

إذا علمت أن قيم كل من Y, X معطاة (معلومة)، ثم اطبع قيم كل من C, B, A, Y, X.

الحل: من الواضح أنه يمكننا من حساب قيمة المتغير A في المعادلة(١) لمعرفتنا بقيم المعطيات الأولية Y, X، ويمكننا من حساب قيمة المتغير B في المعادلة (٢) بالاعتماد على قيمة X المعلومة لدينا وقيمة المتغير Aالمحسوبة في الخطوة السابقة، أما قيمة المتغير C في المعادلة (٣) بالاعتماد على قيم كل من المتغيرات B,A,X وكلها معلومة. وتكون خطوات حل المسألة كما هو مبين في الشكل رقم(٣) كما يلي:

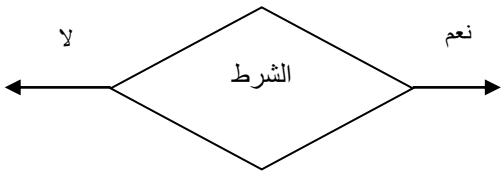


الشكل رقم(٣).

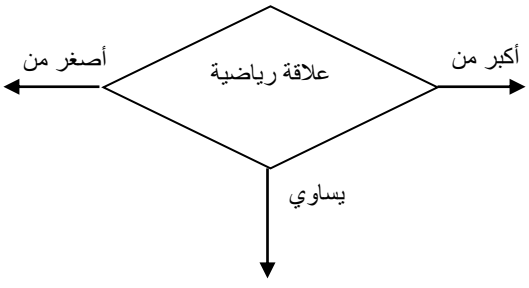
١.ابدأ.
٢.اقرأ قيمة كل من X ,Y.
٣.احسب قيمة A من المعادلة (١).
٤.احسب قيمة B من المعادلة (٢).
٥.احسب قيمة C من المعادلة (٣).
٦.اطبع قيمة كل من C, B, A, Y, X.
٧.توقف.

٢- **المخطط الانسيابي المتفرع**: ويحدث التفرع داخل البرنامج عند الحاجة لاتخاذ قرار أو مفاضلة بين اختيارين أو أكثر فيسير كل اختيار في طريق مستقل (تفرع) عن الآخر. وهناك نوعان من القرارات يمكن للمبرمج استعمال أحدها حسب المسألة المراد حلها.

أ) قرار ذو تفرعين

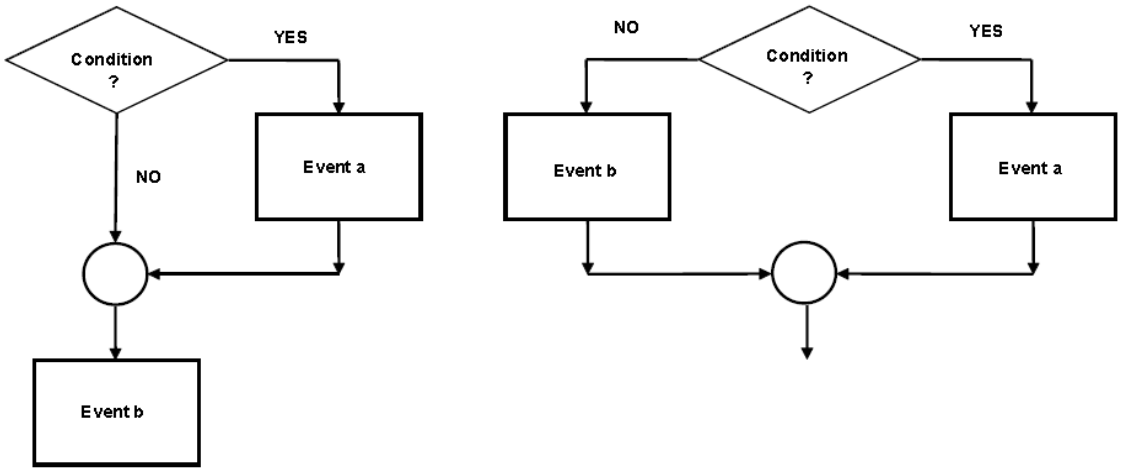


ب) قرار ذو ثلاثة تفرعات



وبشكل عام فإن المخطط الانسيابي المتفرع يمكن أن يأخذ إحدى الصورتين التاليتين:

شكل (أ) شكل (ب)



- في شكل(أ) يبين أنه إذا كان جواب الشرط "نعم" فإن الحدث التالي في التنفيذ هو الحدث (a) ، أما إذا كان جواب الشرط "لا" فإن الحدث التالي في التنفيذ هو الحدث (b).
- في شكل (ب) يبين أنه إذا كان جواب الشرط "نعم" فإن الحدث التالي في التنفيذ هو الحدث (a) يليه الحدث (b) ،أما إذا كان جواب الشرط "لا" فإن الحدث التالي في التنفيذ هو الحدث (b) مباشرة.

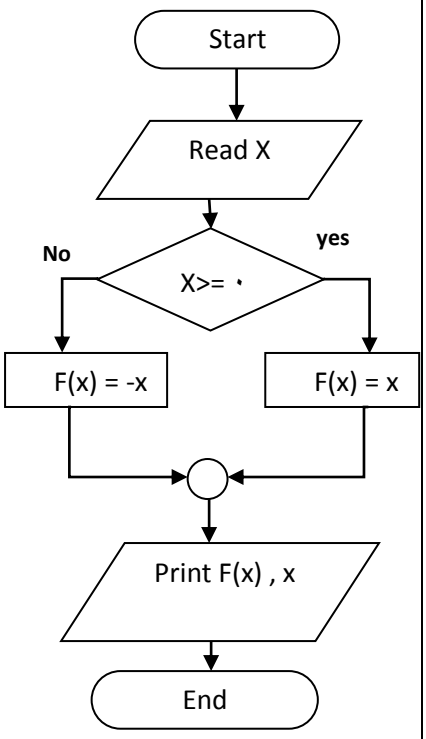
ملاحظة : صيغة الشرط يجب ان تكون محددة بعلاقة بحيث يمكن الاجابة عليها بنعم او لا.

مثال: يوضح القرار ذو التفرعين.

ارسم المخطط الانسيابي لإيجاد قيمة الدالة $F(x)$ المعرفة حسب القاعدة التالية:

$$F(x)=\begin{cases} x\,, x\geq 0 \\ -x\,, x<0 \end{cases}$$

الحل:

Flowchart	Algorithm	
	١- Start ٢- Read x ٣- If x greater than or equal to Zero then go to step (٤), else go to step (٥). ٤- calculate F(x) = x ,and then go to step (٦) ٥-calculate F(x) = -x ٦- print F(x) , x ٧- End.	١- ابدأ ٢- اقرأ قيمة X ٣- إذا كانت X أكبر من أو تساوي صفر اذهب إلى الخطوة(٤). وإلا فاذهب إلى الخطو (٥). ٤- احسب قيمة الدالة F(x) من F(x) = x ثم اذهب إلى الخطوة (٦). ٥-احسب قيمة الدالة F(x) من F(x) = -x . ٦- اطبع قيمة x, F(x) ٧- توقف.

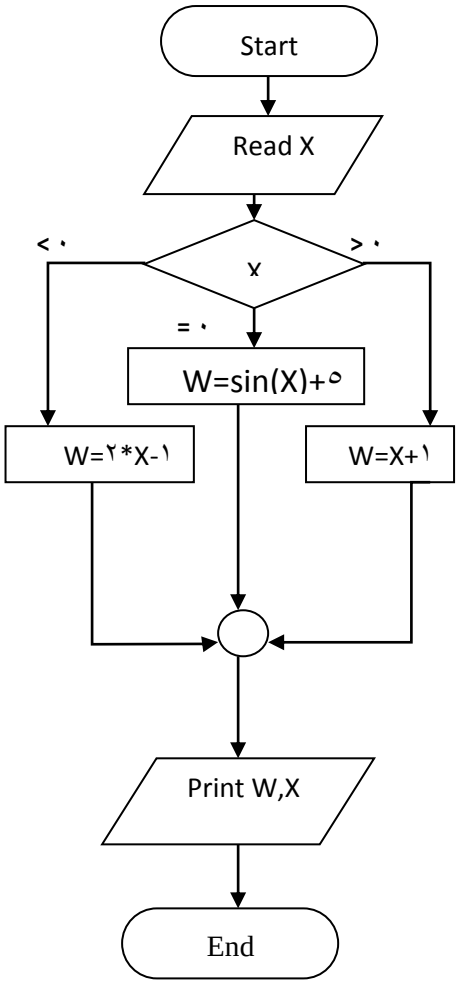
ملاحظة: عند تنفيذ الخطوة (٥) فإن الخطوة المنطقية التالية لها هي خطوة الطباعة (الخطوة (٦)) ، ولذلك لم يتم توجيه البرنامج بالعبارة اذهب إلى خطوة (٦) ، كما أشرنا في الخطوة (٤) حيث تم توجيه البرنامج إلى الخطوة (٦) وتجاوز الخطوة (٥)حيث أن الخطوتان (٤) ، (٥) لا تحققان معا.

مثال: يوضح القرار ذو ثلاثة تفرعات.

ارسم المخطط الانسيابي لحساب قيمة W المعطاة من العلاقة التالية:

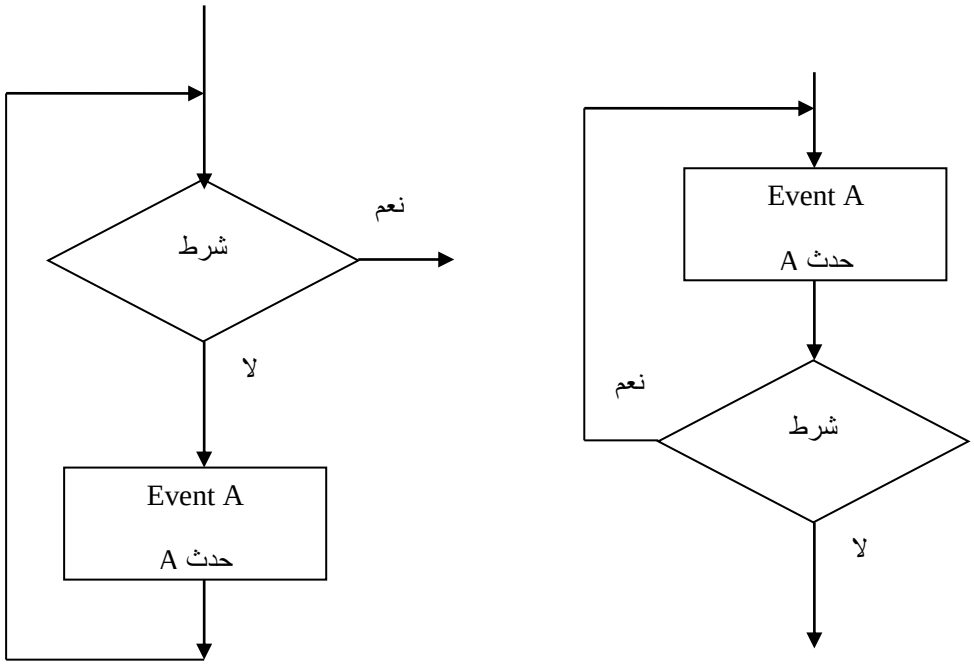
$$W = \begin{cases} X + 1 \,, X > 0 \\ \sin(X) + 5, X = 0 \\ 2X - 1 \,, X < 0 \end{cases}$$

الحل:

Flowchart	Algorithm	
	١- Start ٢- Read x ٣- If x greater than Zero then go to step (٤), If x equal to Zero then go to step (٥), If x less than Zero then go to step (٦). ٤- calculate W =X+١,and then go to step (٧) ٥- calculate W =sin(X)+٥,and then go to step (٧) ٦- calculate W =٢*X-١,and then go to step (٧) ٧- print W , x ٨- End.	١- ابدأ ٢- اقرأ قيمة X ٣- إذا كانت X أكبر من صفر اذهب إلى الخطوة(٤). وإذاكانت X تساوي صفر فاذهب إلى الخطو (٥) أما إذا كانت X أقل من صفر إذهب إلى خطوة (٦). ٤- احسب قيمة W من المعادلة: W =X+١ ثم اذهب إلى خطوة(٧). ٥- احسب قيمة W من المعادلة: W =sin(X)+٥ ، ثم اذهب إلى خطوة(٧). ٦- احسب قيمة W من المعادلة: W =٢*X-١، ثم اذهب إلى خطوة(٧). ٧- اطبع قيم كل من X,W ٨- توقف.

٣- المخطط الانسيابي ذو التكرار الواحد (Simple-Loop Flowchart):

نحتاج هذا النوع من الخرائط لإعادة عملية أو مجموعة من العمليات في البرنامج عددا محدودا أو غير محدود من المرات ويكون الشكل العام لمثل هذه الخرائط كما يلي:



وقد سميت هذه المخططات بمخططات التكرار الواحد لأنها تستعمل حلقة واحدة للتكرار، وتسمى أحيانا بالمخططات الانسيابية ذات التكرار البسيط.

مثال: ارسم مخططا انسيابيا لإيجاد مجموع m من الأعداد الحقيقية $(X_1, X_2, \ldots\ldots, X_m)$

الحل:

النتيجة المطلوبة هي مجموع الأعداد T حيث أن :

$$T= \sum_{i=1}^{m} (X_i)$$

خطوات الحل يمكن أن تسير حسابيا على النحو:

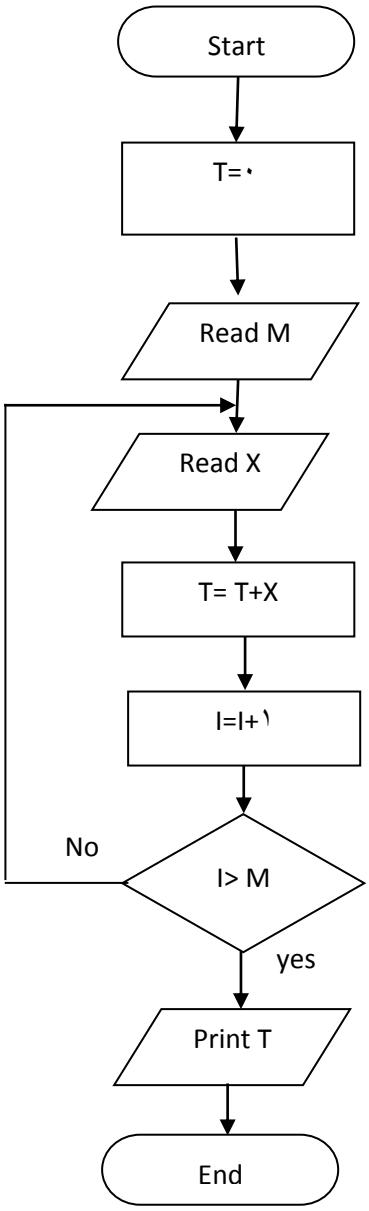
$T_{٠}=٠$, $T_{١}=T_{٠}+X_{١}$, $T_{٢}=T_{١}+X_{٢}$

$T_m=T_{m-١}+X_m=X_{١}+X_{٢}+\ldots\ldots+X_{m-١}+X_m$

ونموذج الحل هذا يمكن أن يختصر بنموذج مكافئ هو

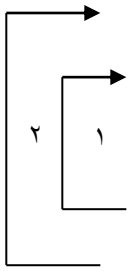
$T = T+X_i$, $i= ١.٢,\ldots..,m$

على أن تكون القيمة الأولى للمجموع T هي $T=٠$



٤- المخطط الانسيابي ذو التكرارات المتعددة (Multi-Loop Flowchart):

في هذه الحالة تكون التكرارات داخل بعضها البعض بحيث لا تتقاطع، فإذا كان لدينا تكراران من هذا النوع كما في الشكل التالي فيسمى التكرار رقم (١) تكراراً داخليا Inner Loop بينما التكرار رقم (٢) تكراراً خارجيا Outer Loop ، ويتم التنسيق بين مثل هذين التكرارين، بحيث تكون أولوية التنفيذ للتكرار الداخلي.



وقد سميت هذه الخرائط بخرائط التكرارات المتعددة لأنها تستعمل أكثر من حلقة تكرار واحدة، وقد تسمى أحيانا بخرائط التكرارات المتداخلة أو المتراكبة أو الضمنية .

مثال: ارسم المخطط الانسيابي لإيجاد معدل درجات كل طالب، إذا كان عدد المواد لكل طالب N، وعدد طلاب الصف M ؟

ملاحظات:

سوف نستخدم المتغير S كمخزن لتجميع درجات الطالب،

والمتغير G يعبر عن درجة الطالب في المادة ،

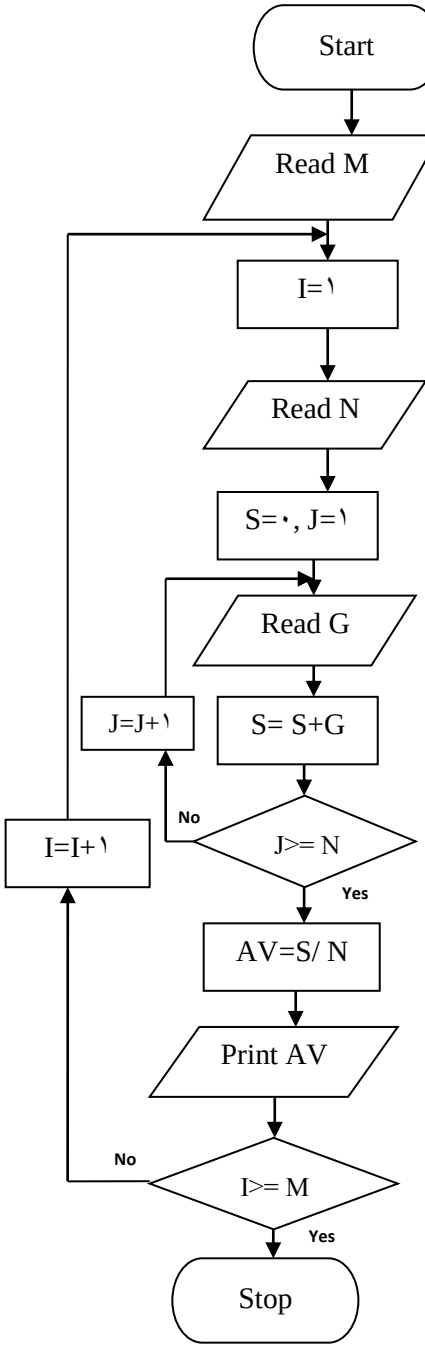
AV يمثل متوسط الدرجات،

N عدد المواد المقررة لكل طالب ،

M عدد الطلاب في كل صف،

J عداد لحصر عدد المواد N ،

I عداد لحصر عدد الطلاب M .

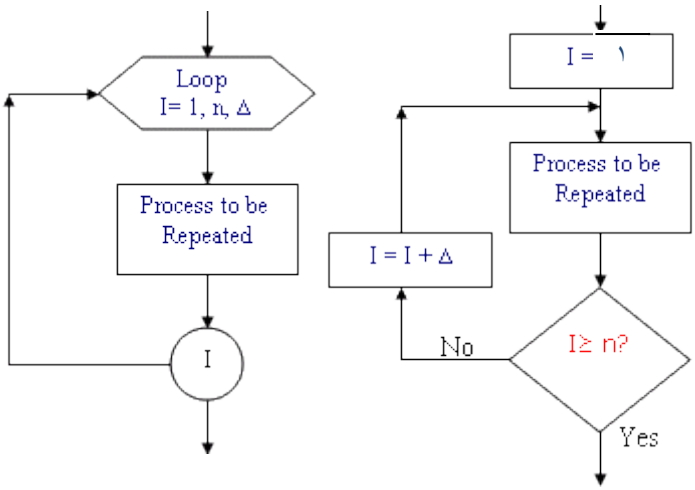


يلاحظ من الشكل أن : التكرار الداخلي يقوم بقراءة وجمع درجات المواد لكل طالب على حده، بينما التكرار الخارجي يقوم بحساب متوسط درجات الطالب الكلية ثم طباعته قبل البدء في إدخال درجات طالب آخر، العداد J للخروج من التكرار الداخلي، العداد I للخروج من التكرار الخارجي.

صيغة التكرار باستعمال الشكل الاصطلاحي (الدوران التكراري Loop):

في الفقرتين السابقتين تعلمنا مفهوم التكرار البسيط والتكرارات المتعددة(المتداخلة) ويمكننا الآن استخدم الشكل الاصطلاحي (الدوران التكراري Loop) . نلاحظ من المثال السابق أننا نحتاج إلى العناصر الآتية:

- القيمة الأولية للعداد I (هنا ١=I).
- القيمة النهائية للعداد I (هنا I=N).
- قيمة الزيادة عند نهاية كل دورة Δ.



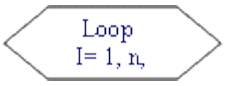
(ب) (أ)

نلاحظ من الشكل السابق في الجزء (أ) أن إجراءات الدوران كانت تتم طبقاً للخطوات الآتية والمفصلة من قبل المبرمج:

١. أعط العداد I قيمة أولية.
٢. أتم الإجراءات المطلوب إعادتها.
٣. اتخاذ قرار: إذا كانت قيمة العداد I وصلت إلى القيمة النهائية N فاخرج إلى الخطوة (٦) في البرنامج وإلا فاذهب إلى الخطوة (٤).
٤. زد I بمقدار Δ.
٥. عد إلى الخطوة (٢).
٦. أكمل ما تبقى من البرنامج.

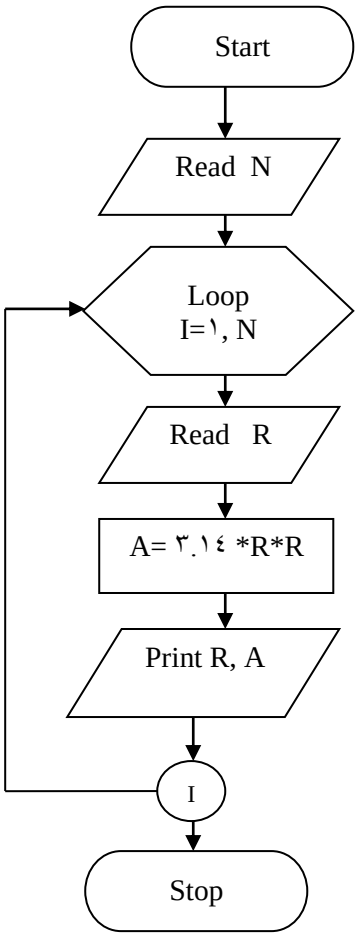
يمكننا استبدال الخطوات المفصلة (١، ٣، ٤، ٥) في الشكل (أ) بخطوة مجملة واحدة مبينة في الشكل الاصطلاحي للدوران (ب) حيث تنفذ هذه العملية بصورة أوتوماتيكية من قبل الحاسوب، وهذا من شأنه تسهيل عملية البرمجة واختصار عدد التعليمات في البرنامج وتجنب بعض الأخطاء.

ملحوظة: تعتبر قيمة Δ تساوي ١ دائمًا إذا لم تعط قيمة أخرى بخلاف ذلك، وفي حالة عدم ذكر قيمة Δ يصبح الشكل الاصطلاحي (الدوران المتكرر) الوارد في الشكل (ب) كما يلي حيث تكون قيمة Δ تساوي ١وبصورة أوتوماتيكية.



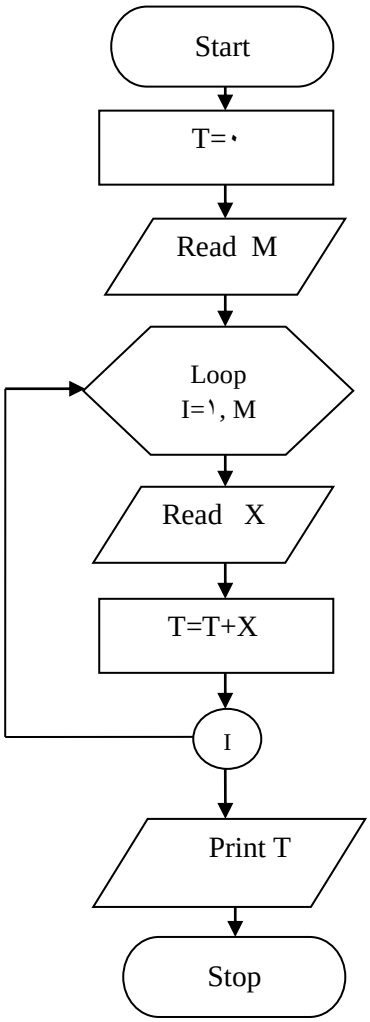
مثال: ارسم خريطة المخطط الانسيابي لإيجاد مساحة N من الدوائر باستخدام الشكل الاصطلاحي للدوران.

الحل:

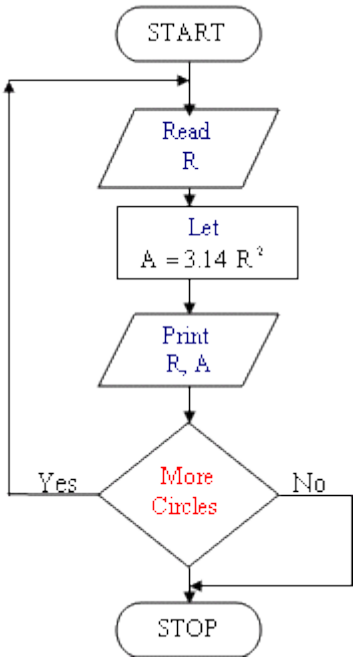


مثال: ارسم مخططاً انسيابياً لإيجاد مجموع M من الأعداد الحقيقية X_m ,.....,X_2,X_1 .

الحل:



مثال: ارسم مخططا انسيابيا لإيجاد مساحة مجموعة من الدوائر أنصاف أقطارها معلومة:



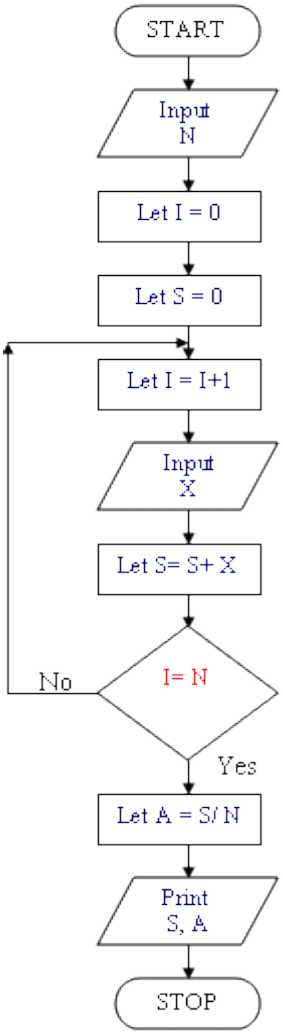
١. ابدأ.
٢. اقرأ نصف قطر الدائرة (R).
٣. أوجد مساحة الدائرة (A).
٤. اطبع قيم كل من A, R.
٥. هل هناك مزيد من الدوائر؟
فإن كان نعم فعد إلى الخطوة(٢) وإن كان لا فعد إلى الخطوة (٦).
٦. توقف.

في كثير من الأحيان نحتاج في برامج الحاسوب الإلكتروني إلى جمع مجموعة كبيرة من الأعداد التي تمثل معطيات ظاهرة معينة، فمثلاً قد نرغب في إيجاد الوسط الحسابي لأعمار طلاب الجامعة، ولتحقيق هذا أولاً يجب أن نحسب مجموع أعمار الطلاب، وطبعًا ليس عمليًا إعطاء رمز أبجدي لكل عمر طالب فقد تحتاج لأكثر من عشرة الآلاف رمز، في مثل هذه الحالات نصمم خوارزمية معينة للتجميع تسمى خوارزمية التجميع Summers Algorithm تتضمن خطوات محددة إذا اتبعها الحاسب استطاع أن يجمع أي كمية من البيانات باستخدام متغيرين اثنين إحداهما هو المتغير الذي نجمعه والآخر هو الجمع الإجمالي (المجمع)، ويمكن تحديد الخطوات التي يجب أن يتبعها الحاسب لتحقيق ذلك في أربع خطوات هي:

١. اجعل المجمع مساويًا الصفر.
٢. ادخل قيمة واحدة للمتغير.
٣. اجعل القيمة الجديدة للمجمع تساوي القيمة القديمة له زائد القيمة المدخلة للمتغير، أي أن: قيمة المجمع الجديدة=قيمة المجمع القديمة + آخر قيمة مدخلة للمتغير.
٤. كرر ابتداءً من الخطوة الثانية.

مثـال: ارسـم المخطـط الانسـيابي لإيجـاد الوسـط الحسـابي لأعمـار طـلاب شـعبتك.

الحل: نفترض أن إجمالي عدد الطلاب =N ونستخدم عددًا لرقم كل طالب ونرمز لـه بـالرمز I ونرمز لعمر الطالب بX ونستخدم مجمعًا لأعمار الطلبة ونرمز له بالرمزS ونستخدم الرمز A ليدل على معدل أعمار الطلبة.



١. ابدأ.
٢. ادخل إجمالي عدد الطلاب (N).
٣. اجعل I=٠.
٤. اجعل S=٠.
٥. اجعل I=I+١.
٦. ادخل X.
٧. اجعل S=S+X.
٨. إذا كانت I=N اذهب إلى الخطوة ٩
وإلا اذهب إلى الخطوة ٥.
٩. اجعل A=S/N.
١٠. توقف.

❖ البرامج (الروتينات) الفرعية Sub-Routines

عبارة عن برامج مكونة من مجموعة من الجمل المتسلسلة والمتكاملة لتنفيذ عمل معين تكتب مرة واحدة وتنفذ مرة واحدة أو اكثر عندما يتم استدعائها في اي موقع من البرنامج الرئيسي.

عند استخدام البرامج الفرعية نستخدم الامريين التاليين:

١- أمر الاستدعاء GOSUB : هو الامر الخاص بتوجيه التحكم في سير تنفيذ البرنامج الرئيسي الى اول جملة تنفيذية في البرنامج الفرعي والصيغة العامة لهذا الامر:

Ln GOSUB n

حيث أن :

- Ln : تعني رقم السطر.
- GOSUB : تعني اذهب الى البرنامج الفرعي.
- n : تعني رقم اول سطر في اول جملة تنفيذية في البرنامج الفرعي.

٢- أمر الرجوع Return : الصيغة العامة :

Ln Return

مثال : اكتب برنامج بلغة بيسك يتضمن برنامج فرعي لطباعة التالي:

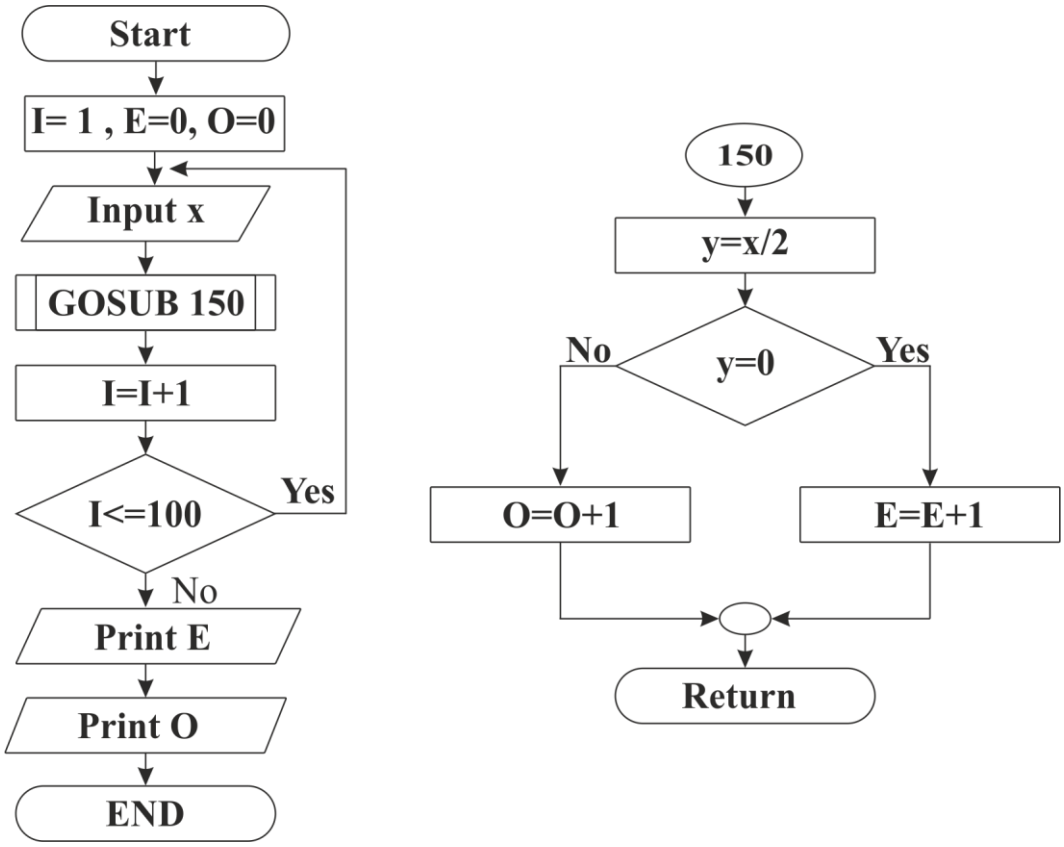
```
١٠- GOSUB ١٠٠
٢٠- Print "---------------------------------"
٣٠- GOSUB ١٠٠
٤٠- END
١٠٠- Print "************************"
١٠١- Print "        COMPUTER"
١٠٢-Print "************************"
١٠٣- Return
```

```
****************************
        COMPUTER
****************************
---------------------------------
****************************
        COMPUTER
****************************
```

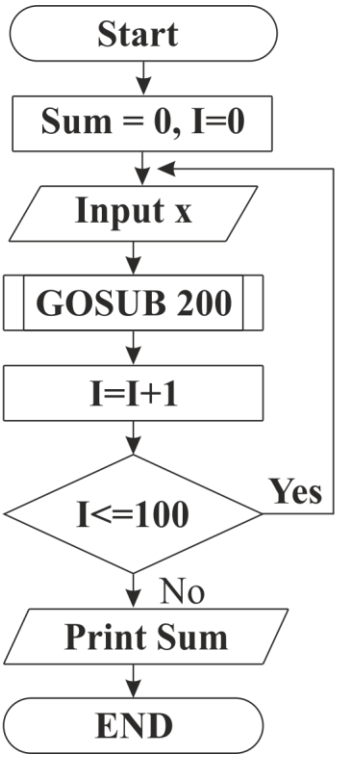
الهدف الاساسي من كتابة البرامج الفرعية

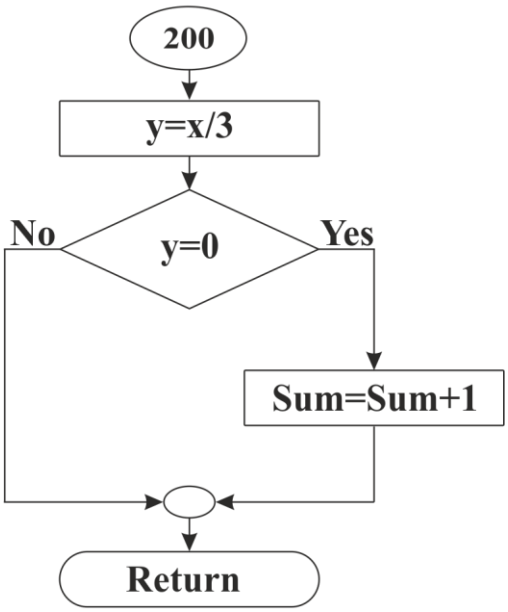
١- يستخدم لتقليل التكرار حيث قد يحتاج المبرمج عند كتابة برامج معينة الى تكرار مجموعة من العبارات والاجراءات اكثر من واحدة من مواقع متعددة من البرنامج ولأجل عدم كتابة وتكرار هذه الاجراءات نستخدم البرنامج الفرعي الذي يكتب مرة واحدة ويمكن استدعاءه عدة مرات من البرنامج الرئيسي.
٢- سهولة اكتشاف الاخطاء والسيطرة عليها خلال البرنامج بالإضافة الى امكانية حذف او اضافة برامج فرعية جديدة.
٣- توفير مواقع خزن في الذاكرة نتيجة لاختزال بعض العمليات المتكررة في البرنامج.

مثال : ارسم مخطط انسيابي لحساب عدد الاعداد الفردية وعدد الاعداد الزوجية لمجموعة مكونة من ١٠٠ عدد باستخدام البرنامج الفرعي او (باستخدام الاستدعاء الذاتي) .

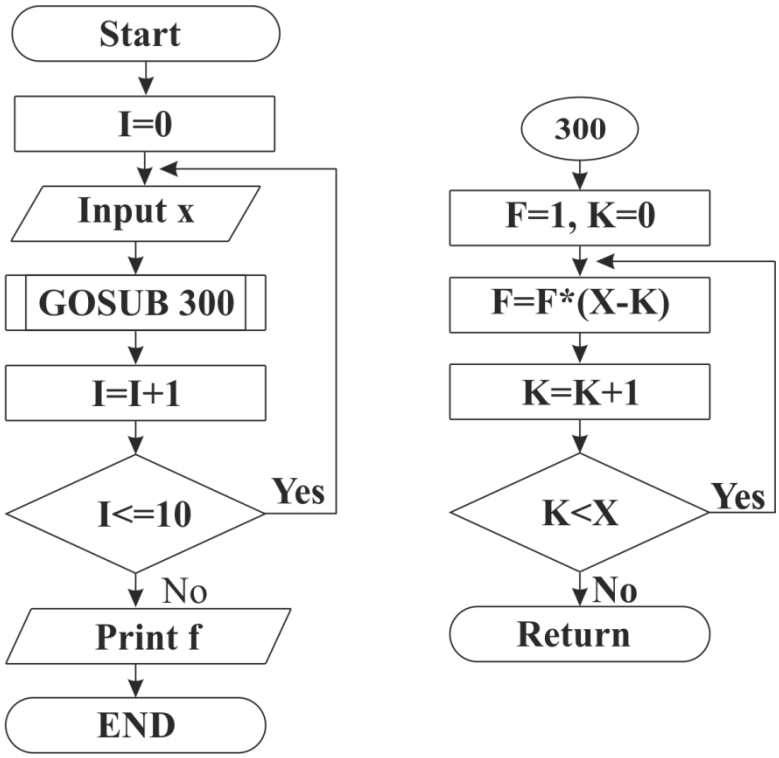


مثال : ارسم المخطط الانسيابي لحساب مجموع الاعداد التي تقبل القسمة على ٣ لمجموعة مكونة من ١٠٠ عدد مستخدما الاستدعاء الذاتي.

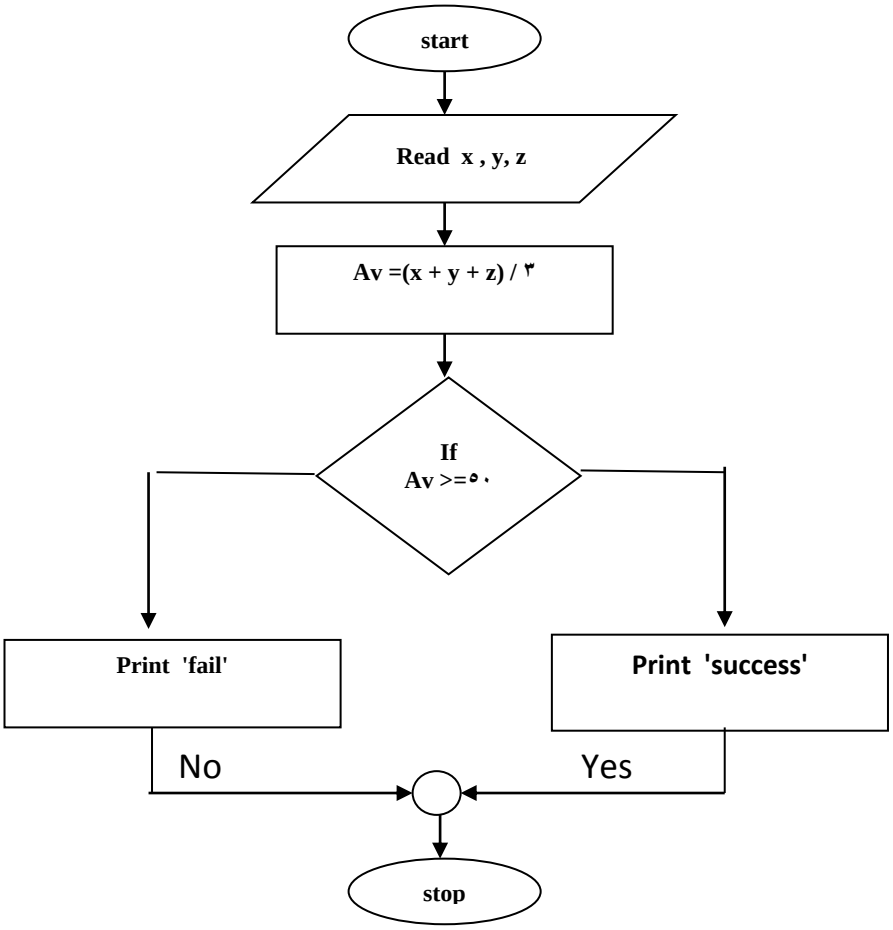




مثال: ارسم مخطط انسيابي لطباعة مفكوك ١٠ اعداد باستخدام البرنامج الفرعي.



مثال: اكتب خوارزمية تجد المعدل لثلاث درجات فإذا كان المعدل ناجح تطبع كلمة 'success' وعكس ذلك تطبع كلمة 'fail' ؟



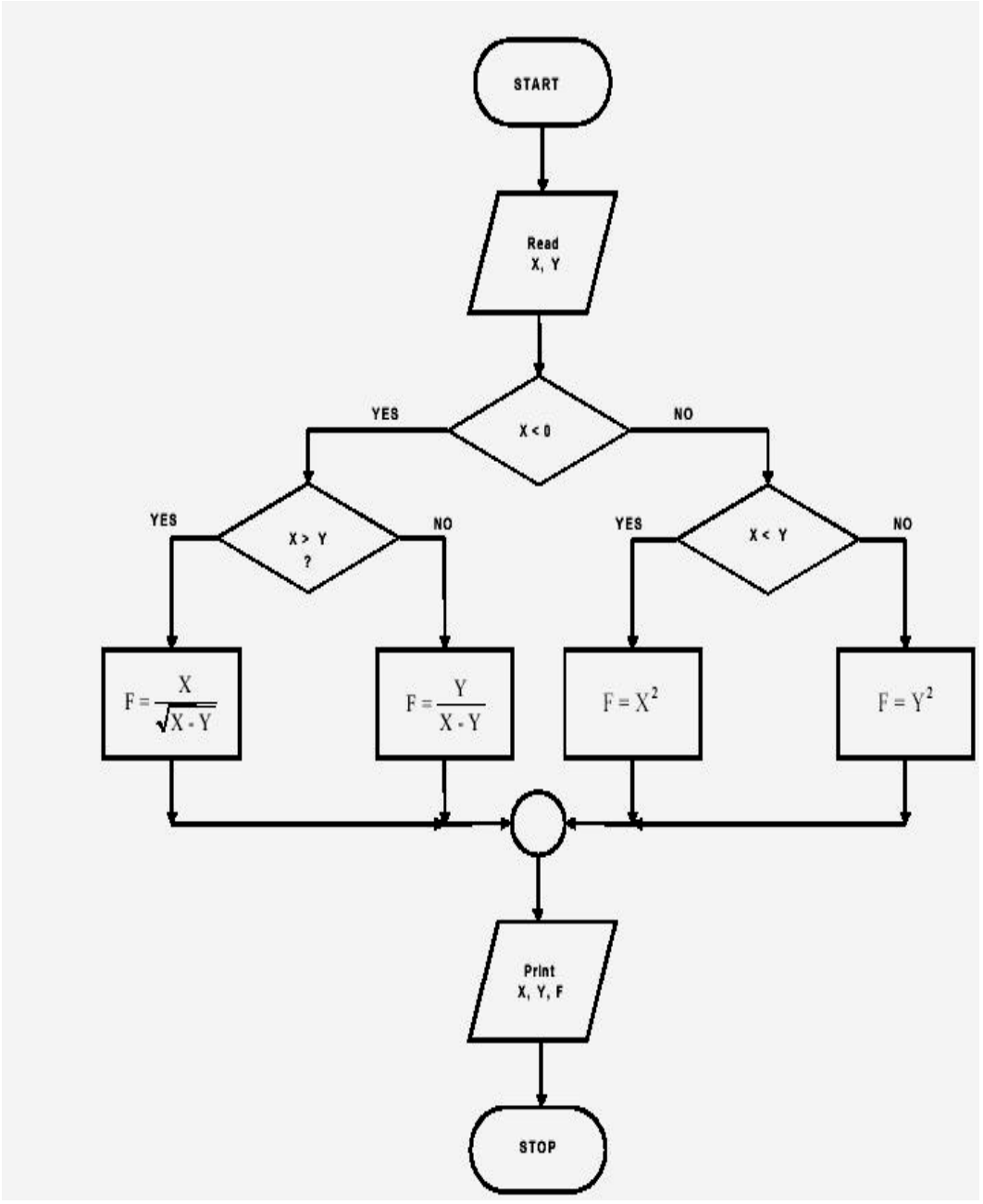
: H .W

١. اكتب خوارزمية تقرأ عدد فإذا كان العدد سالب تطبع كلمة 'negative' أما إذا كان العدد موجب تطبع كلمة 'positive' ؟

٢. اكتب خوارزمية تقرا ثلاث درجات فإذا كان المعدل مابين ٨٠..١٠٠ تطبع كلمة 'very good' وإذا كان مابين ٥٠..٧٩ اطبع كلمة 'Good' أما إذا كان اقل من ٥٠ تطبع 'fail' ؟

٣. اكتب خوارزمية تقرأ ثلاثة أعداد ومن ثم تطبع العدد الأكبر ؟

٤. ما هو الإخراج للخوارزمية التالية عندما x=٣ و y=-٢ ؟



❖ **المصفوفات** **Matrix**

وتكون على ثلاث انواع:

١- مصفوفة ذات البعد الواحد DIM-١.

٢- مصفوفة ذات البعدين DIM-٢.

٣- مصفوفة ثلاثية الابعاد DIM-٣.

المصفوفة ذات البعد الواحد: هي قائمة أو متسلسلة تتألف من مجموعة من العناصر في خلايا الذاكرة وتخزن في كل خلية قيمة واحدة فقط، وهذه الخلايا مجاورة لبعضها البعض.

مثلاً: المجموعة ذات الاسم الرمزي X تتألف من ٥ عناصر هي :

(٤٢، ٢٢,٧٧,٨٥,٥٠) تسمى العناصر.

X(١)،X(٢)،X(٣)،X(٤)،X(٥). تسمى بالمؤشرات.

ويمكن تمثيلها في وحدة الذاكرة في ترتيب متسلسل. فارقام اماكن التخزين مرتبة بشكل تصاعد تبدأ من ادنى قيمة للمؤشر الى اعلى قيمة.

مثلاً : X(١) هي العنصر الأول في المصفوفة X.

X(٢) هي العنصر الثاني في المصفوفة X.

وكما في الشكل التالي:

X(١)	٤٢
X(٢)	٢٢
X(٣)	٧٧
X(٤)	٨٥
X(٥)	٥٠

ولغرض تعريف المصفوفات في البرنامج نستخدم عبارة DIM حيث يتم من خلالها تعريف اسم المصفوفة وعدد عناصرها وخزنها في ذاكرة الحاسبة والصيغة العامة لها:

Ln DIM Array name (Constant).

Ln: تعني رقم السطر في البرنامج.

DIM: عبارة غير تنفيذية مهمتها اعلام مترجم الحاسبة بأن هناك مصفوفة يمكن احتوائها في ذاكرة الحاسبة، ويجب ان تسبق اي جملة تنفيذية في البرنامج لها علاقة بالمصفوفة.

Constant : الثابت .. يمثل عدد القيم او العناصر المراد خزنها في المصفوفة .

مثال: ارسم مخطط انسيابي لقراءة عناصر المصفوفة X(١٠) وطباعتها.

مثال: ارسم مخطط انسيابي لقراءة عناصر المصفوفة X(١٠) وطباعتها بشكل معكوس.

مثال: ارسم مخطط انسيابي لقراءة عناصر المصفوفة X(١٠) وطباعة الاعداد الموجودة في المواقع الفردية منها.

مثال: ارسم مخطط انسيابي لقراءة عناصر المصفوفة X(١٠) وطباعة الاعداد الموجودة في المواقع الزوجية منها.

مثال: ارسم مخطط انسيابي لقراءة عناصر المصفوفة X(١٠) وطباعة الاعداد الفردية منها.

مثال: ارسم مخطط انسيابي لقراءة عناصر المصفوفة X(١٠) وطباعة الاعداد الزوجية منها.

مثال: ارسم مخطط انسيابي لحساب معدل عناصر مصفوفة مكونة من ٥٠ عنصر.

مثال: ارسم مخطط انسيابي لإيجاد اكبر عنصر في مصفوفة مكونة من عشرة عناصر.

مثال: ارسم مخطط انسيابي لإيجاد اصغر عنصر في مصفوفة مكونة من عشرة عناصر.

مثال: اذا توفرت لديك بيانات عن اوزان واطوال واعمار ٥٠ طالب في قائمة.

المطلوب : رسم مخطط انسيابي واحد لحساب وطبع:

١- عدد الطلاب الذين تزيد اعمارهم عن ٦٠ كيلو غرام واطوالهم عن ١٦٠ سنتيمتر.
٢- عدد الطلاب الذين تزيد اعمارهم عن ٢٠ سنة واطوالهم عن ١٥٠ سنتيمتر.

❖ **خوارزميات الترتيب** **Sorting**

الترتيب: هو عملية ترتيب مجموعة العناصر البيانية وفق تسلسل معين. فالبيانات الرقمية تُرتبْ بشكل تصاعدي اول تنازلي، أما البيانات الحرفية فترتب ابجدياً.

٠<١<٢<٣<٤<٥<٦<٧<٨<٩

Blank<A<B<C<D…...<Z

الغرض من الترتيب:

تعددت اغراض عملية الترتيب وأهمها:

١- لزيادة كفاءة خوارزمية البحث.
٢- لتبسيط وتسهيل عملية معالجة الملفات.
٣- لحل مشكلة تشابه وتكرار القيود.

خطوات عملية الترتيب:

تتلخص خطوات خوارزمية الترتيب بكافة انواعها بالمراحل التالية:

١- قراءة الحقول المراد ترتيبها.
٢- الاستدلال (الاستنتاج) لموقع العنصر في الترتيب الجديد.
٣- نقل العنصر البياني الى موقعه الجديد.

انواع خوارزميات الترتيب:

أ- **الترتيب الداخلي**: هو الترتيب الذي يحدث داخل الذاكرة الرئيسية للحاسوب(**Main Memory**) عندما يكون حجم البيانات مناسباً، ومن اهم انواعه:

١- الترتيب بطريقة التبادل.
٢- ترتيب الفقاعة Bubble Sort.
٣- الترتيب بالاختيار Selection Sort.
٤- الترتيب بالحشر Insertion Sort.
٥- الترتيب السريع Quick Sort.
٦- الترتيب بالدمج Merge Sort

ب- الترتيب الخارجي External Sort

هو ترتيب البيانات المخزونة في وسائط الخزن الثانوية على شكل ملفات عندما يكون حجم البيانات كبير جداً بحيث يتعذر استيعابها كلها في الذاكرة في وقت واحد اثناء عملية الترتيب. ومن اهم انواعه:

١- الترتيب بالدمج ذي المسارين Two-Way-Merge Sort.
٢- الترتيب بالدمج المتعدد المسارات K-Way-Merge Sort.
٣- الترتيب بالدمج المتوازن ذي المسارين Balance Two-Way.
٤- الترتيب بالدمج متعدد الاطوار Polyhase Two-Way Merge.

العوامل الرئيسية المحددة لاختيار خوارزمية الترتيب.

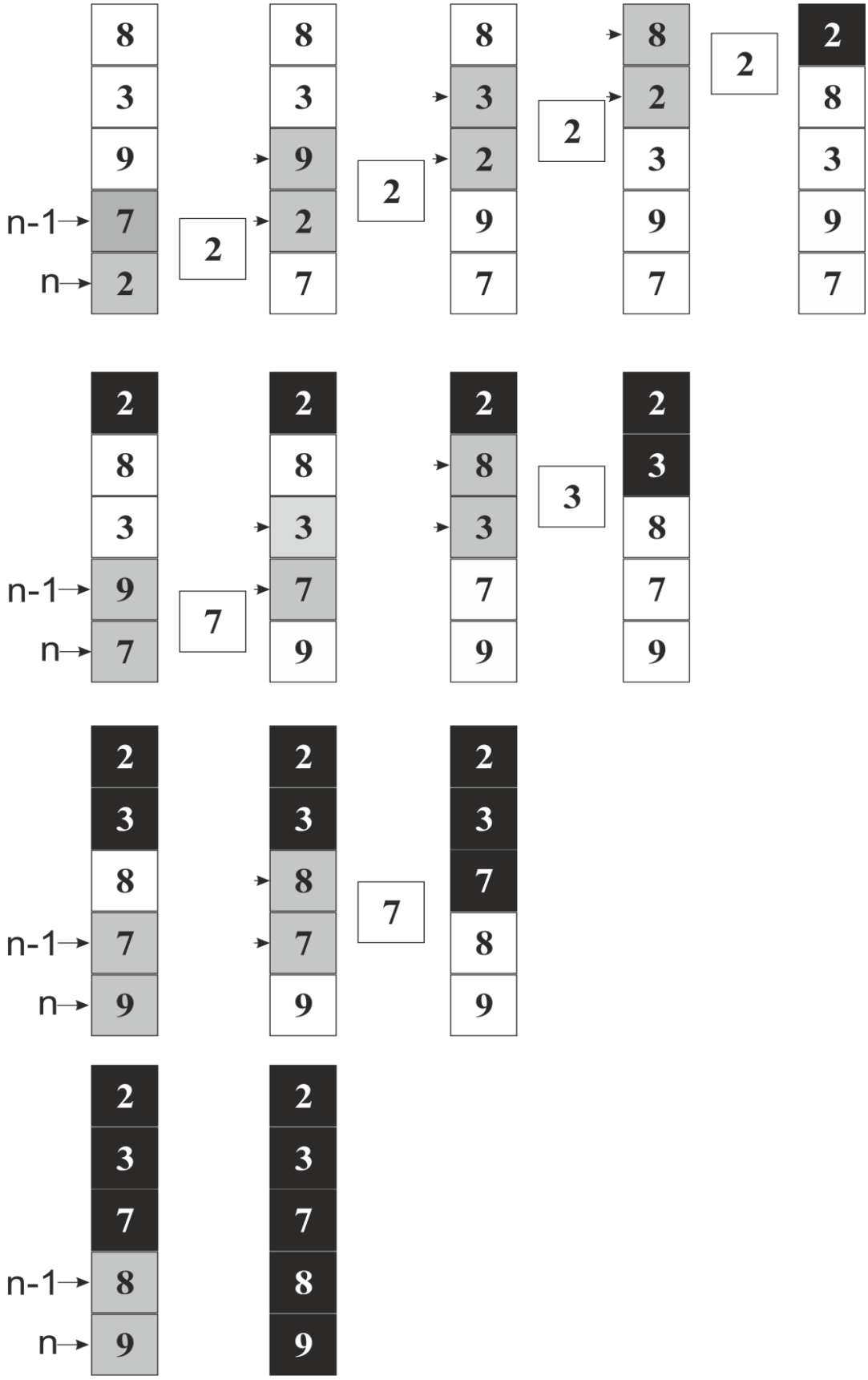
ان اختيار خوارزمية الترتيب يجب ان يكون في ضوء عدد من العوامل من أهمها:

١- حجم البيانات المخزونة.
٢- نوع الخزن (الذاكرة الرئيسية، قرص ..الخ).
٣- درجة ترتيب البيانات(غير مرتبة، شبه مرتبة).

❖ **ترتيب الفقاعة** **Bubble Sort**

تعمل هذه الخوارزمية على اساس نقل اصغر قيمة في المجموعة الى الموقع الاول ثم ثاني اصغر قيمة الى الموقع الثاني وهكذا(في حالة الترتيب التصاعدي) ويتم الترتيب كما يلي:

١- في المرحلة الاولى: نقارن العنصرين في الموقعين [n] و [n-١] ونرى ايهما اكبر ونبادل موقعهما ليكون الاصغر قبل الاخر، ونستمر لأعلى القائمة لحين الحصول على مقارنة العنصر في الموقع الثاني مع العنصر في الموقع الاول.
٢- في المرحلة الثانية: نقارن بنفس الطريقة السابقة ولكن من العنصر في الموقع [n] الى العنصر في الموقع الثاني، الشكل رقم(٤) يوضح خطوات عمل الخوارزمة.


8
3
9
n-1 7
n 2
2
8
3
9
2
7
2
8
3
2
9
7
2
8
2
3
9
7
2
8
3
9
7
2
8
3
7
9
3
2
3
8
7
9
2
3
8
n-1 7
n 9
2
3
8
7
9
7
2
3
7
8
9
2
3
7
n-1 8
n 9
2
3
7
8
9

الشكل رقم (٤)

❖ **الترتيب بالاختيار** **Selection Sort**

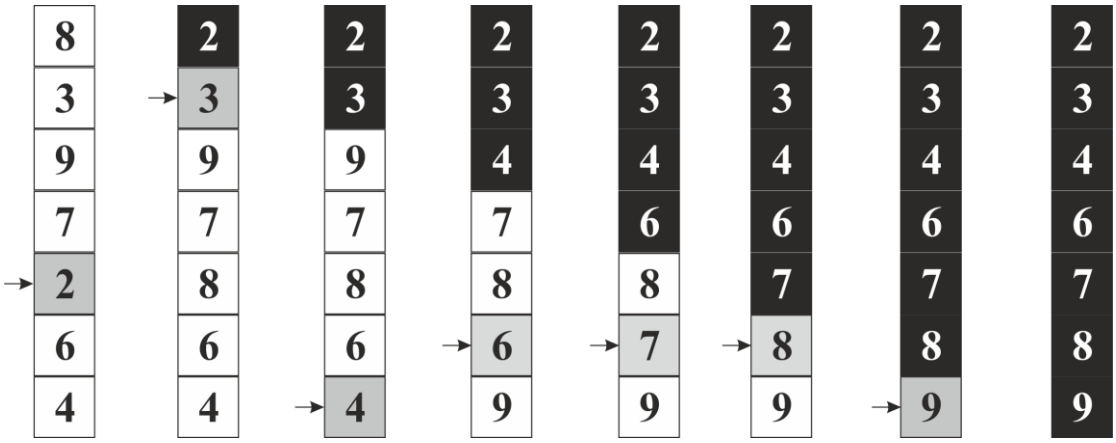
يتم الترتيب في هذه الخوارزمية البحث عن اصغر عنصر ووضعه في الموقع ثم البحث عن ثاني اصغر عنصر ووضعه في الموقع الثاني وهكذا الى ان يتم ترتيب جميع العناصر، وتتلخص هذه الخوارزمية:

١- ايجاد اصغر عنصر في القائمة واستبداله من موقعه مع العنصر في الموقع الاول من القائمة.

٢- ايجاد اصغر عنصر من المتبقي من القائمة واستبداله من موقعه مع العنصر في الموقع الثاني من القائمة.

٣- نستمر في هذه العملية لحين الوصول الى نهاية القائمة.

مثال : رتب القائمة التالية تصاعدياً (٨ ٣ ٩ ٧ ٢ ٦ ٤).



الشكل رقم (٥)

❖ **تنظيم البيانات** **Data Organization**

هناك تنظيم هرمي خاص للبيانات يتكون من ملف البيانات يسمى (Data file) الذي يحوي مجموعة من السجلات (القيود Record) التي تتكون من عدد من الحقول (Fields) وفيما يأتي وصف هذا التنظيم:

١- الملف File : عبارة عن مجموعة من السجلات المنظمة على أساس منطقي يربط بينها خاصية أو أكثر وتعبر عن مضمون محدد مفهوم معين.

٢- السجل Record : هو الوحدة المنطقية المكونة للملف ويتكون السجل من مجموعة من الحقول (Fields) المتعلقة بموضوع معين، مثال ذلك :

رقم الموظف	اسم الموظف	القسم	الشهادة	.....

يمثل سجل موظف واحد في ملف موظفي دائرة معينة.

٣- الحقل (Fields) : وهو أصغر وحدة يتكون منها الملف والحقل يتكون من مجموعة من الرموز (Characters) المستخدمة في اللغة.

تصنف الملفات تبعاً لطريقة تداول البيانات فيها الى:

١- **الملفات التتابعية (التسلسلية)** **Sequential files**

هي تلك الملفات التي تعتمد في تخزين واسترجاع البيانات على الطريقة التتابعية (Sequential Access) تعني وصول تسلسلي تتابعي، فاذا اردنا الوصول لسجل معين في الملف ولنفرض رقم ٢٠ فسوف نمر على جميع السجلات التي تسبقه وصلاً إليه.

٢- **الملفات العشوائية** **Random File**

هي تلك الملفات التي يمكن الوصول الى اي سجل فيها بطريقة مباشرة (Direct Access) دون الحاجة الى المرور على ما يسبقها من بيانات ودون اعتبار الى مواضع تخزينها.

❖ **تطوّر أساليب البرمجة**

هناك ثلاثة أساليب للبرمجة: عشوائية – هيكلية – شيئية.

١. البرمجة العشوائية Random Programming

ويتم التركيز في هذا الأسلوب على حل المسألة برمجيا للوصول إلى الهدف دون النظر إلى عملية تنظيم البرنامج مما يؤدي إلى إنتاج برامج صعبة التطوير أو اكتشاف الأخطاء، ولا يظهر مشاكل هذا الأسلوب إلا إذا كان البرنامج كبيرا أما في حالة البرامج الصغيرة والبسيطة ربما كان هذا الأسلوب مناسب لها.

٢. البرمجة الهيكلية Structured Programming

ويعتمد هذا الأسلوب على تجزئة البرنامج إلى عدة برامج جزئية أو فرعية حيث يتم الربط بين هذه البرامج الفرعية لتشكيل البرنامج العام ويحتاج هذا الأسلوب إلى تخطيط جيد وتظهر فاعليته في حالة المسائل متوسطة الحجم كما يسهل اكتشاف الأخطاء بهذا الأسلوب و إجراء عمليات التطوير بالإضافة إلى عدم تكرار المقاطع البرمجية.

٣. البرمجة الشيئية Object Oriented Programming (OOP)

لغات البرمجة متعددة ومتطورة، وكانت في بدايتها مثل (C, BASIC, FORTRAN,COBOL) تستخدم أسلوب تقليدي في البرمجة وهذا الأسلوب يعتمد على التحديد الدقيق لترتيب تنفيذ الأوامر وإجراء العمليات.

في البرمجة الشيئية (OOP) تنفيذ الأوامر لا يتبع التسلسل المنطقي، وذلك لأن كل فعل من المستخدم (User) يتسبب في حدوث حدث معين (Event) وهذا بدوره يدفع برنامج معين أو جزء آخر من البرنامج إلى العمل علماً بأن هذا الجزء مكتوب مسبقاً ومرتبط بهذا الحدث.

❖ **البرمجة بالكائنات** **The Object Model**

- الكائن (Object)
- الخصائص (Properties)
- الوسائل (Methods)
- الأحداث (Events)

في البرمجة الشيئية لا يتبع التسلسل المنطقي مثل البرمجة القديمة ولكن يقوم المشغل بالضغط على أزرار معينة فيحصل على استجابة لما يريده.

الكائن object: هو شيء له وجود ويتميز بمجموعة من الخصائص والوسائل والاحداث التي تقع عليه.

أمثلة على بعض الكائنات:

- جهاز الكمبيوتر
- السيارة
- قلم الحبر

فالسيارة مثلاً كائن (Object) يتكون من كائنات (Objects) مثل المحرك وكذلك المحرك يتكون من كائنات أخرى.

الخصائص Properties: وهى السمات والخواص التي تميز الكائن (لونه – حجمه – مكانه – شكله .....).

الوسائل Methods: هي الوظائف والأفعال التي تكون مصاحبة للكائن (Object)، هي وصف لسلوك معين مصاحب للكائن، أي هي أفعال محددة مسبقاً لكل كائن.

مثال: سيارة (Object)، BMW١ (ObjectName) ، لونها أحمر (Property) ، وتسير في الشارع (Method).

ويتم تنفيذ الوسيلة عن طريق مناداة الوسيلة، ويتم ذلك برمجياً كالتالي:

ObjectName.Method()

BMW١.Move

الحدث Event: هو عبارة عن الفعل الذي يقع على الكائن ويستجيب له.

كل كائن (Object) له أحداث معينة (Events) تقع عليه ويستجيب لها، حيث يمكن للمبرمج وضع مجموعة معينة من الأوامر التي يتم تنفيذها عند وقوع حدث معين.

مثال: إنسان يمشي الى قاعة الامتحان اسمه أحمد وبدأ وقت الامتحان فأسرع لتفادي التأخر والحرمان من الامتحان.

الكائن (Object): إنسان.

الخاصية (Property): الاسم.

قيمة الخاصية (Value): أحمد.

الحدث (Event): بدأ وقت الامتحان.

الوسائل (Methods): يسرع.

بدأ وقت الامتحان يمثل حدث (Event) واستجابة هذا الكائن (الإنسان) للحدث بشكل معين مثل الاسراع يمثل (Methods).

مثال: عند فتح ملف نصي والتعديل فيه وعند الضغط على علامة الإغلاق (×) فإن صندوق الحوار يظهر للتأكد منك على حفظ الملف أو عدم الحفظ أو إلغاء الأمر.
ماذا حدث؟! عند ضغطك على الإغلاق فإنك بذلك قمت بتشغيل حدث إغلاق الملف فوجد تغيير في الملف ولم يحفظ بعد واستجابة لهذا الحدث تم تنفيذ وسيلة عرض صندوق الحوار.

بعض المصطلحات الهامة في البرمجة الشيئية

- **مفهوم التغليف Encapsulation**
- **مفهوم التوريث Inheritance**

التغليف: يعني إخفاء بيانات داخل الكائن أو التصنيف بحيث لا يتم الوصول إليها إلا بصلاحيات معينة.
أمثلة:
جهاز الهاتف المحمول الذي نستخدمه لا نعرف تفاصيل عمله، ومع ذلك فنحن نستخدمه ونتعامل معه في حدود المسموح لنا، وهو بذلك يمثل مفهوم التغليف من حيث عدم ظهور المعلومات الداخلية للهاتف.... وهكذا.

الغرض من التغليف Encapsulation:
الحفاظ على بيانات التصنيف وحمايتها و التركيز فقط على ما يراد استخدامه فعلاً.

T.V

Mobile

Capsules

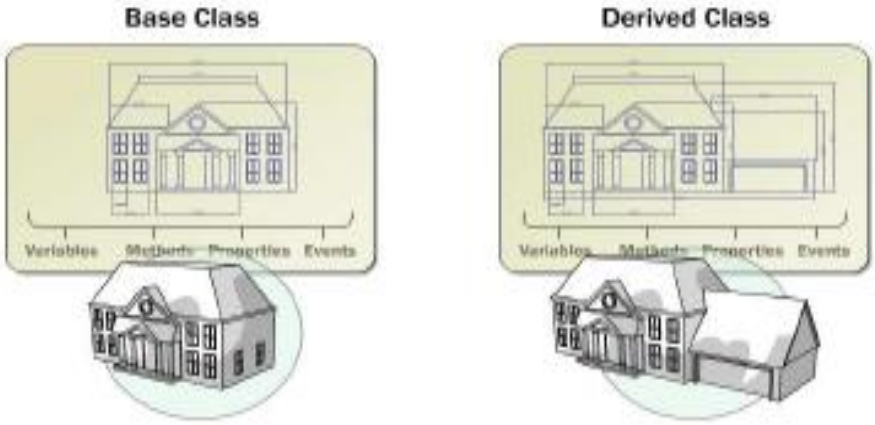
التوريث: يقصد به أن تصنيف (Class) معين (Derived) يرث (Inherit) خصائص ووظائف تصنيف آخر (Base) وهذا التصنيف الجديد يكون لديه جميع خصائص التصنيف الأساسي بالإضافة إلى خصائص الإضافية.

أمثلة:

افترض أنك مهندس معماري وطلب أحد الأشخاص منك رسم هندسي لبيت وقد أعجب التصميم صاحب هذا الشخص وطلب منك تصميم بيتا مثله مع إضافة حمام سباحة في سطح البيت، ماذا كنت ستفعل؟

الذي سوف تفعله هو أنك تأخذ نسخة من نفس الرسم الهندسي للبيت ثم تضيف إلى الرسم حمام السباحة في السطح كمثال.

فيكون الرسم الهندسي الأول للبيت يمثل (Base Class) أو (Parent Class) والرسم الهندسي الثاني يمثل (Derived Class) أو (Child Class).

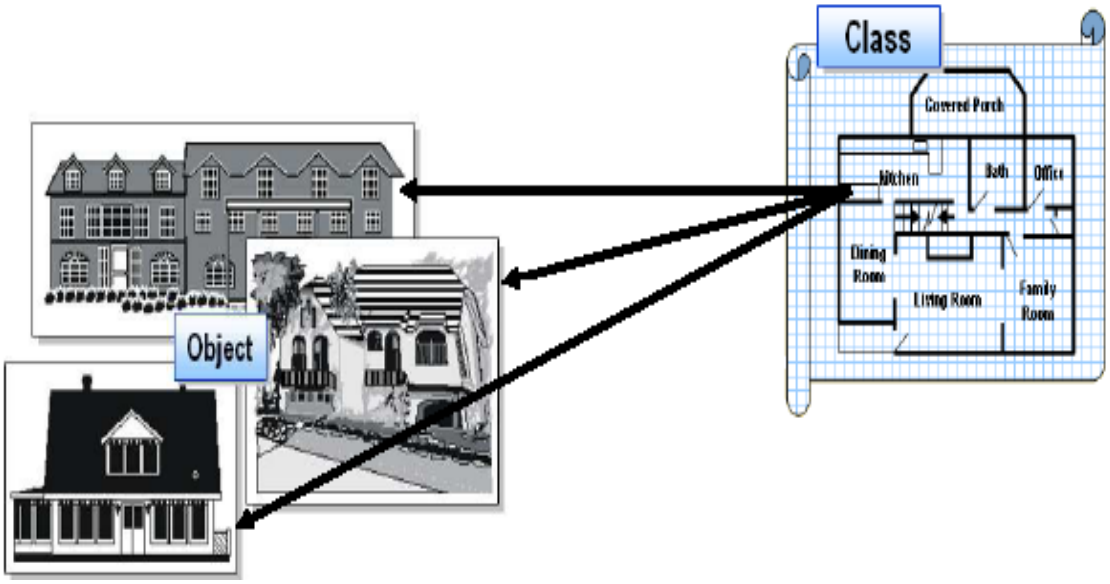


الغرض من التوريث Inheritance:

هو إعادة استعمال ما تم تصميمه من فئات (Classes) والتعديل فيها حسب الحاجة بدلاً من إعادة كتابة الفئة من جديد، وهذا من أهم مميزات البرمجة الشيئية.

الفئة **Class**: هي عبارة عن قالب أو مخطط يتم منه إنشاء كائن معين، وهو يمثل جميع الخصائص والوظائف التي سوف يحتويها الكائن بعد ذلك.

أما الكائن (Object) فهو يمثل وحدة مستقلة تم إنشاؤها من الفئة (Class) وهي التي تستخدم فعلاً لأداء الوظائف المختلفة للفئة (Class).



يحتوي التصنيف (Class) على خصائص (Properties) ووظائف (Methods) وأحداث (Events) وهي التي سوف يتضمنها بعد ذلك أي كائن سوف يتم إنشاءه من هذا التصنيف.

فعند استخدام قالب الرسم الهندسي لبناء منزل ممكن أن نبني منزلاً لونه أبيض وآخر لونه أصفر وآخر لونه بني وهذا ما نسميه في البرمجة خصائص الكائن (Object Properties).

House.Color = Yellow