



# اسم المادة : معالجة البيانات

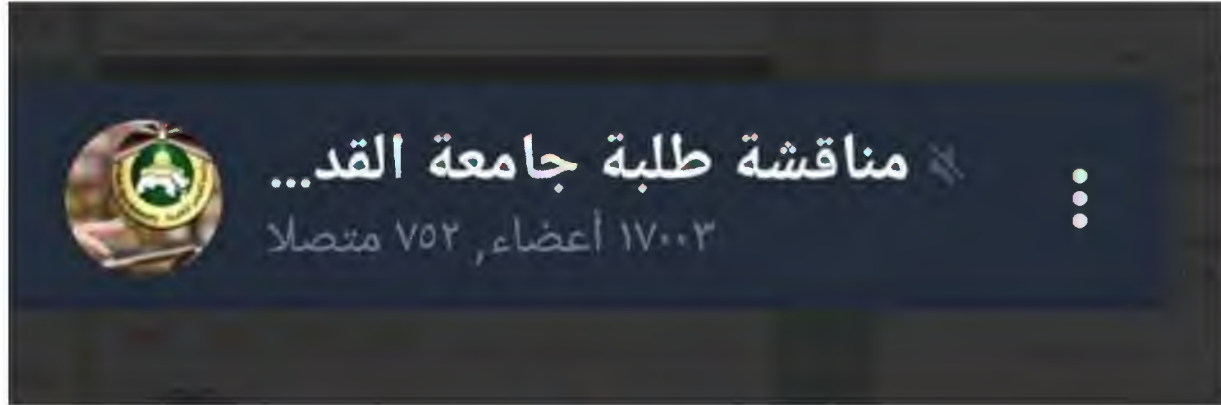
تجمع طلبة كلية التكنولوجيا والعلوم التطبيقية - جامعة القدس المفتوحة

[acadeclub.com](http://acadeclub.com)

وُجد هذا الموقع لتسهيل تعلمنا نحن طلبة كلية التكنولوجيا والعلوم التطبيقية وغيرها من خلال توفير وتجميع **كتب وملخصات وأسئلة سنوات سابقة** للمواد الخاصة بالكلية, بالإضافة لمجموعات خاصة بتواصل الطلاب لكافة المواد:

للوصول للموقع مباشرة اضغط **هنا**

وقفكم الله في دراستكم وأعانكم عليها ولا تنسوا فلسطين من الدعاء 



سنوات نصفي

وسائط متعددة

تلغرام مناقشة طلبة جامعة القدس المفتوحة

<https://t.me/talbaalqds>

للوصول الى القناة اكتب في خانة البحث في التلجرام

مناقشة طلبة جامعة القدس المفتوحة



## مقرر معالجة البيانات "نصفي"

اعداد الطالب : مصطفى الغول

### الوحدة الأولى

#### 2. الخوارزميات و مخططات سير العمليات

عزيزي الدارس، في حياتنا اليومية قد تواجه أنماطا وأشكالا مختلفة من المسائل تحتاج إلى حل. هناك العديد من الحالات المسائل التي تتعامل معها عن طريق المعرفة أو الخبرة السابقة، لكن هذا الأسلوب يكون غير مجد أحيانا، لذا نحتاج إلى إجراءات أو خطوات مرتبة لحل المسألة. هناك العديد من الأساليب أو الإجراءات المتوافرة في بيئة علوم الحاسوب من الممكن استخدامها في حل المسائل. ومما يجدر ذكره أن هذه الخطوات أو الإجراءات التي يتم اتباعها لحل مسألة معينة تسمى الخوارزمية .

عزيزي الدارس، ترجع تسمية الخوارزمية Algorithm إلى عالم الرياضيات العربي (الخوارزمي) من القرن التاسع الميلادي. ويمكن تعريف الخوارزمية بدقة أكبر على أنها مجموعة (سلسلة من الخطوات المرتبطة منطقية، تتميز بالوضوح لوصف طريقة حل يفهمها الإنسان، ويمكن تمثيل الخوارزمية باستخدام لغة تشبه إلى حد بعيد اللغة المستخدمة في لغات البرمجة وهو ما يطلق عليها Pseudocode ، أو بطريقة الرموز الاصطلاحية التي تسمى مخططات سير العمليات Flowcharts . إن حل أي مسألة يتطلب تجزئتها إلى خطوات مبسطة متسلسلة ليتمكن الحاسوب من حلها. وهذا يعني أنه يمكن وضع أكثر خوارزمية لحل المسألة الواحدة. وتتراوح الخوارزميات بين البسيطة والمعقدة، ويعود ذلك إلى طبيعة المسألة التي أعدت لها الخوارزمية.

عزيزي الدارس، تتميز الخوارزمية بخصائص متعددة يجب أن تتوافر فيها، منها:

1- الوضوح والدقة في كل خطوة من خطوات الحل والبعد عن التعقيد.

2- الترابط والوضوح في التسلسل المنطقي للخطوات

3- محدودية الخطوات.

4- تكامل الخطوات وشموليتهما .

30/10/2022 14:15

صفحة 3

3. مراحل حل المسألة باستخدام الحاسوب

1.3 نظرة شاملة لمراحل حل المسائل باستخدام الحاسوب

المسائل التي تواجهك، عزيزي الدارس، في حياتك اليومية كثيرة، وهي تتزايد يوماً بعد يوم، وقد تستطيع معالجة بعض هذه المسائل من خلال خبرتك ومعارفك السابقة، فمثلاً أنت لا ترمي بنفسك إلى الأرض من علو لأنك تعرف نتيجة ذلك مسبقاً، وفي قرارك هذا تأخذ عوامل كثيرة بالاهتمام منها: المسافة التي ستقطعها في أثناء السقوط، والزمن الذي ستستغرقه عملية السقوط، والنتائج المترتبة عليها. في حين تحتاج بعض المسائل إلى التفكير وإيجاد الحلول واتخاذ القرارات المناسبة، الأمر الذي يتطلب مهارة في حل هذه المسائل التي قد تواجهها في حياتنا اليومية. إن استخدام الحاسوب في حل هذه المسائل أمر مهم بل لا يستطيع الاستغناء عنه في معظم الأحيان. إن استخدام الحاسوب في حل المسائل يعتمد على منهجيات معينة يمكن توظيفها في حل أنواع مختلفة من المسائل، وهذا يتطلب درجة عالية من الإتقان.

هنالك، عزيزي الدارس، إجراءات وطرق متعددة متوافرة في حقول علم الحاسوب لتساعدنا على ترتيب طريقة تفكيرنا وتحليلنا في حلنا للمسائل المطروحة أمامنا، أشهرها الطرق الآتية التي سيتم مناقشتها في الوحدة الثالثة

- طريقة التحليل Analytical Approach

- طريقة الخوارزمية Algorithmic Approach.

3- طريقة هندسية البرامج Software Engineering Approach

قبل أن نشرع، عزيزي الدارس، في حل المسألة باستخدام الحاسوب حاول أن تجيب عن الأسئلة الآتية

1- هل من ضرورة الاستخدام الحاسوب؟

2- هل تتطلب عملية المعالجة المحوسبة وقتاً طويلاً؟

3- أيهما أقل تكلفة: حل المسألة باستخدام الحاسوب أم حلها بالوسائل الأخرى؟

4- هل تنطوي عملية المعالجة على تكرار لمجموعة من الخطوات ؟

5- هل هناك حاجة لحل المسألة أكثر من مرة باستخدام بيانات مختلفة ؟

6. هل تعرف كيفية حل المسألة باستخدام الحاسوب؟

صفحة 4

7. هل يمكنك تحديد المعطيات (المدخلات) والنتائج (المخرجات) ؟

8- ما لغة البرمجة التي سنستخدمها ؟

إذا تمكنت من الإجابة عن الأسئلة السابقة فإنك تستطيع تلخيص مراحل حل المسائل باستخدام الحاسوب في الخطوات الآتية

- 1- تعريف المسألة بدقة ووضوح وتحديد عناصرها .
- 2- وضع خوارزمية المسألة بشكل صحيح
- 3- تمثيل الخوارزمية بمخطط سير العمليات (Flowchart) .
- 4- كتابة البرنامج بإحدى لغات برمجة الحاسوب
- 5- إدخال البرنامج والبيانات إلى ذاكرة الحاسوب
- 6- تنفيذ البرنامج (بعد أن تتم ترجمته في الحاسوب إلى لغة الآلة) والحصول على النتائج
- 7- تجربة البرنامج

8- توثيق البرنامج (شرح أجزاء البرنامج بالتفصيل، للرجوع إليه وقت الحاجة لحل المسألة ذاتها أو لاستخدامه في حل مسائل أخرى مشابهة)

عزيزي الدارس، إن حل المسألة يتطلب وضع خوارزمية خاصة بها في ضوء تعريفها وفهمها وتحليلها، ومن ثم تمثيل الخوارزمية بمخطط سير العمليات المتسلسل والترابط منطقية، ووضع برنامج لحل المسألة باستخدام الحاسوب، وللحصول على الحل الصحيح للمسألة يجب عليك اتباع الخطوات التي تم ذكرها سابقا وتنفيذها بصورة صحيحة، لأن أي خلل من شأنه أن ينعكس سلبا على الحل بصورة عامة. من الواضح لك، عزيزي الدارس، أنه يجب دراسة المسألة بعناية فائقة من جميع الجوانب وتحليلها قبل الشروع بصياغة أي برنامج وكتابته، وذلك لأن الفهم والتحليل هما الأساس للحصول على النتائج المرجوة

### 2.3 تحليل المسألة

بعد تحليل المسألة من أهم المراحل اللازمة لإيجاد الحل المناسب لها، فهو يعني نهم المسألة وتعريفها ، وتحديد جميع عناصرها والمعلومات اللازمة لحلها. إن طريقة الحل مهما كانت جيدة تبقى عديمة الفائدة إذا لم توصلنا إلى المحل الصحيح. وقد يكون من الممكن حل

صفحة 5

مسألة ما بطرق مختلفة، ولن تستطيع اختبار الحل الأفضل ما لم تحلل المسألة وتدرسها بدقة وعناية

عزيزي الدارس، عند تحليل المسألة عليك أن تحدد ، ما يأتي

- معطيات المسألة أو مدخلاتها ، وهي البيانات التي ستتم معالجتها .

- متغيرات المسألة

- مخرجات المسألة

ومن المفيد أن تعرف أن هناك تبايناً بين المسائل من ناحية تحليلها، فتحليل مسألة رياضية قد يكون أسهل من تحليل مسألة إعداد قوائم خاصة برواتب الموظفين في مؤسسة ما. وتقع مهمة تعريف المسألة على عاتق المتخصصين في المجال الذي تنتمي إليه تلك المسألة، ولا تدخل في إطار المهمات الملقاة على عاتق المتخصصين في علم الحاسوب. ويمكن القول إن تحليل المسائل هر فن أكثر من كونه علماً

عزيزي الدارس، من الواضح أن عملية تحليل المسألة تعتمد على تجميع كل المعلومات التي تتعلق بالمسألة المتوي حلها باستخدام الحاسوب بالإضافة إلى محاولة مناقشتها مع الآخرين. ويمكن تلخيص مفهوم تحليل المسألة في النقاط الخمس الآتية

1. تعريف المسألة بدقة، حيث يتم فهمها وتحديد البيانات المتوافرة، والشروط المرافقة

2- تجزئة المسألة إلى أجزاء (Sub problems) لتسهيل حلها .

3. عمل قائمة بالمدخلات (المعطيات المتوافرة، ووصف كل عنصر من عناصرها .

4- وضع قائمة تحتوي على مخرجات المسألة (نتائج المسألة).

5- تحديد العلاقة المختلفة بين المدخلات إلى جانب تحديد العمليات الضرورية التي تقود إلى وضع طريقة الحل الملائمة للوصول إلى النتائج المرجوة

نلاحظ، عزيزي الدارس، أن طريقة حل المسائل بشكل عام تشبه إلى حد بعيد طريقة حل المسائل الرياضية والفيزيائية التي كنت تعلمتها في مراحل دراسية سابقة، مثل إيجاد مساحة دائرة أو مستطيل، أو حساب الراتب الصافي لموظف، أو حساب معدل علاماتك في الفصل الدراسي

صفحة 6

عزيزي الدارس، إن حل المسألة سواء أكانت بسيطة أم معقدة يجب أن تتم بعناية وفق منهجية متكاملة فلا تتسرع في وضع الحلول السريعة أو النتيجة مباشرة بل يجب التأنى والتفكير بالخطوات قبل اتخاذها .

مثال (1)

أحسب المساحة (4) الدائرة نصف قطرها (R)

هذا السؤال يندرج ضمن الأسئلة السهلة، وهو من النمط المألوف وتري أن معطيات السؤال تقتصر على نصف قطر الدائرة (R)، وأن المطلوب منك هو إيجاد مساحة الدائرة (A). من خلال دراستك تعرف أن العلاقة الرياضية التي تربط بين مساحة الدائرة ونصف قطرها هي

$$A = 3.14 R^2$$

والسؤال الذي يطرح نفسه عليك في هذه الأثناء هو: كيف يمكنك وضع طريقة لحل هذا السؤال نكر قليلا ثم انتقل إلى دراسة الأجزاء الآتية من هذه الوحدة .

عزيزي الدارس، إن حل المسألة السابقة باستخدام الحاسوب يحتاج إلى عدد من الخطوات الواجب اتباعها، ويمكن إيجازها كما يأتي

1- تحديد المعطيات والنتائج المطلوبة

2- وضع خطوات الحل التي تقودك إلى معرفة مساحة الدائرة وذلك بالاستعانة بالعلاقة

A-3.14 R

3. تخزين حاصل ضرب النسبة التقريبية (314) في مربع تصف القطر في الموقع التخزيني المحتش في ذاكرة الحاسوب المساحة الدائرة A

صفحة 7

3.3 : وضع الخوارزمية المناسبة

درست، عزيزي الدارس، في الأجزاء السابقة من هذه الوحدة كيفية تحليل المسألة وكيفية تحديد عناصرها وفهمها وتحليلها، كما ذكرنا سابقا أن مسمى الخوارزمية Algorithm نسبة إلى الخوارزمي عالم الرياضيات العربي في القرن التاسع الميلادي، والخوارزمية عبارة عن سلسلة خطوات واضحة دقيقة مرتبطة منطقية، وظيفتها وصف طريقة الحل لمسألة معينة بأسلوب يفهمه الإنسان. ويمكن تمثيل الخوارزمية بطريقة الرسومات والأشكال الاصطلاحية كما هو الحال في مخططات سير العمليات التي سنتعرف عليها من خلال دراستنا للأجزاء المتبقية من هذه الوحدة، أو بطريقة تشبه لغات الحاسوب الراقية التي تعرف ب Pseudocode

إن حل أي مسألة يتطلب تجزئتها إلى خطوات مبسطة متسلسلة ليتمكن الحاسوب من حلها ، ويعني ذلك أنه يمكن وضع أكثر من خوارزمية لحل المسألة الواحدة. ويمكن تصنيف الخوارزميات إلى نوعين :

1- خوارزميات بسيطة.

2- خوارزميات معقدة .

يرجع هذا التصنيف، عزيزي الدارس، إلى طبيعة المسألة التي تم إعداد الخوارزمية لحلها. من ناحية أخرى هناك عدد من الصفات والخصائص التي يجب أن تتوفر في أي خوارزمية وهي

1- الوضوح والدقة في كل خطوة من خطوات الحل البعد عن التعقيد).

2- الترابط والوضوح في التسلسل المنطقي للخطوات .

3- محدودية الخطرات.

#### 4- التكامل والشمولية

##### تدريب (1)

ضع خوارزمية تصف عملية انتقالك من سكنك إلى جامعة القدس المفتوحة

صفحة 8

#### 4.3 تمثيل الخوارزميات باستخدام مخطط سير العمليات

تبين، عزيزي الدارس، أن الخوارزمية يمكن تمثيلها باستخدام لغة تشبه إلى حد بعيد اللغة المستخدمة في لغات البرمجة أو بطريقة الرموز الاصطلاحية التي تسمى مخططات العمليات Flowcharts ، وليتسنى لنا تمثيل الخوارزمية باستخدام مخطط سير العمليات يجب معرفة مكونات مخطط سير العمليات التي سيتم دراستها في الأجزاء هذه الوحدة، إن مخطط سير العمليات هو عبارة عن مجموعة من الرموز الاصطلاحية التي يمكن من خلالها تمثيل خطوات الخوارزمية لتسهيل فهمها ، عزيزي الدارس، إن حل أي مسألة باستخدام الحاسوب يتطلب معرفة طبيعة المسألة وتحليلها، ومن ثم كتابة الخوارزمية، بالإضافة إلى مخطط سير العمليات، ومن ثم كتابة البرنامج باستخدام إحدى لغات البرمجة كما سيتم توضيحه في الأجزاء اللاحقة من هذه الوحدة .

#### 5.3 كتابة البرنامج باستخدام إحدى لغات البرمجة

تعتمد طريقة كتابة البرنامج على اختبار لغة البرمجة التي ستستخدم الكتابة البرنامج (البرنامج Program : هو مجموعة من التعليمات التي يتم كتابتها باستخدام إحدى لغات البرمجة لتنفيذها على الحاسوب بفرض حل مسألة معينة). إن اختبار لغة البرمجة، عزيزي الدارس، يجب أن تتم بعناية فائقة وشديدة لما لها من أثر مباشر في كتابة البرنامج، فيجب معرفة نقاط القوة والضعف اللغة البرمجة المستخدمة بشكل عام إن عملية التحضير الكتابة البرنامج متشابهة في معظم لغات البرمجة المعروفة مع وجود بعض الاختلافات في التعليمات والاقتراحات الخاصة بكل لغة، عزيزي الدارس، قد يتبادر إلى وهناك العديد من الأسئلة مثل: ما الخطوات الواجب اتباعها لكتابة البرنامج ما المراحل التي يمر بها البرنامج حتى يصل إلى مرحلة التنفيذ فيما يأتي الإجابة عن السؤالين السابقين بالإضافة إلى العديد من الأسئلة التي تدور في ذهنك.

عزيزي الدارس، عند التفكير في استعمال الحاسوب لحل مسألة معينة، هنالك العديد من الخطوات التي يجب المرور بها وهي كما يأتي

1. تعريف المسألة وفهمها، وتحديد معالمها من ناحية المدخلات والنتائج، والعمليات

المطلوب إجراؤها على البيانات المدخلة لاستخراج النتائج المطلوبة

في الوحدة الأولى

2. تحديد الخوارزمية اللازمة لحل المسألة



- 3- تتبع الخطوات المحددة في الخوارزمية التي حددت وذلك لمعرفة ما إذا كان هذا الحل صحيح أم لا للمسألة التي يتم التعامل معها. (وهذا يمكن أن يتم من خلال الفحص اليدوي) .
- 4- ترجمة خطوات الخوارزمية إلى اللغة التي حددت ومن الأمثلة على لغات البرمجة الممكن استخدامها (C , Pascal, Basic, ...)
- 5- إدخال البرنامج إلى الحاسوب.
- 6- ترجمة البرنامج إلى لغة الآلة، وذلك باستعمال المترجم الخاص بلغة البرمجة ويمكن توضيح ذلك من خلال الشكل (1):

البرنامج المصدر

المترجم

البرنامج الهدف

Source Program

Compiler

Object Program

الشكل (1): مراحل تحويل البرنامج من البرنامج المصدر إلى البرنامج الهدف

صفحة 10

- 7- تنفيذ البرنامج (أي البرنامج الهدف الذي تم تحويله باستخدام المترجم الخاص بلغة البرمجة المستخدمة

- 8- تعريف البرنامج إلى عينة من البيانات الاختيارية للتأكد من صحته منطقية

- 9- البدء باستخدام البرنامج وتنفيذه على البيانات الحقيقية بعد التأكد من أنه يؤدي القرض المطلوب منه كاملاً للحصول على النتائج المطلوبة في حل المسألة

6.3 تنفيذ البرنامج وتحديثه

عزيزي الدارس، بعد إعداد برنامجك وإدخاله إلى الحاسوب والتأكد من خلوه من الأخطاء تأتي مرحلة ترجمة البرنامج إلى لغة الآلة الحاسوب لا يفهم إلا لغته (لغة الآلة)، أي أن الحاسوب لا يفهم تلك البرامج المكتوبة باللغات عالية المستوى، ويمكن ترجمة البرنامج إلى لغة الآلة من خلال المترجم ( Compiler ) الخاص بلغة البرمجة المستخدمة كما تم ترشيحه سابقاً حيث يحول البرنامج المصدر ( Source Program ) إلى برنامج بلغة الآلة، يطلق عليه عادة البرنامج الهدف ( Object Program ) .

## 10 الوحدة الأولى

عزيزي الدارس، من الممكن جدا في أثناء مرحلة التنفيذ للبرنامج اكتشاف بعض الأخطاء التي لم تكن واضحة من قبل (في مرحلة كتابة البرنامج وتصميمه الأمر الذي يتطلب إجراء تعديل على البرنامج وتحديثه وذلك بإعادة ترجمته إلى لغة الآلة، من ناحية أخرى يمكن تعديل (تحديث البرنامج لسبب آخر وهو ظهور بعض المتطلبات الجديدة للمسألة

### 4 مخططات العمليات

تحدثنا في بعض البنود السابقة من هذه الوحدة عن الخوارزميات، وأبرزنا أهميتها وضرورتها لحل المسائل باستخدام الحاسوب، وأشرنا إلى أن الخوارزميات يمكن تمثيلها بعدة طرق أهمها اللغات الإنسانية، ولغات الحاسوب، بالإضافة إلى مخططات سير العمليات Flowcharts . وسنركز في الجزء التالي من هذه الوحدة على دراسة مكونات مخططات سير العمليات بالإضافة إلى أهميتها .

### 1.4 الرموز الاصطلاحية المستخدمة في مخططات سير العمليات

عزيزي الدارس، إن مخططات سير العمليات تستخدم العديد من الرموز والأشكال الاصطلاحية، وهي تشبه إلى حد ما المخططات الهندسية المعمارية. فأنت بحاجة إلى رسومات توضيحية تسهل عليك وعلى غيرك فهم خطوات حل المسألة المعينة، لا سيما إذا اتسمت هذه المسألة بالصعوبة والتعقيد. وفيما بعد يمكنك ترجمة المخطط إلى برنامج تكتبه بإحدى لغات البرمجة التي أشرنا إليها في فقرات سابقة من هذه الوحدة .

ممکن، عزيزي الدارس، تصنيف الرموز الاصطلاحية التي تتكون منها مخططات سير العمليات على النحو الآتي:

1- مجموعة رموز اصطلاحية قياسية (Standard Symbols) تكون على هيئة أشكال بدل كل منها على عملية معينة، بحيث يمكن الاستدلال على أي خطوة من خطوات الخوارزمية من خلال الرمز الاصطلاحي المستخدم فيها.

2- مجموعة الحروف الأبجدية العربية أو الإنجليزية أو غيرها .

3- مجموعة الأرقام إضافة إلى الرموز الرياضية

4- مجموعة الرموز الخاصة مثل علامات التنقيط وغيرها .

5- قواعد وقوانين تربط المجموعات آنفة الذكر ببعضها ببعض

## صفحة 11

ستستعرض، عزيزي الدارس، في البنود التالية أهم الأشكال والرموز الاصطلاحية المستخدمة في مخططات سير العمليات.

## 2.4 الرمز الطرفي Terminal Symbol

بدل الرمز الطرفي على بداية مخطط سير العمليات أو نهايته كما هو موضح في

الشكل (2):

START

STOP

يكون هذا الرمز على شكل بيضاوي

الشكل (2) الرمز الطرفي

صفحة 12

رمز القراءة/ الطباعة Output Symbol / Input

بدل رمز القراءة الطباعة (الشكل (3)) على

- قراءة بيانات أو إدخالها، مثال قراءة قيمة X

- طباعة معلومات أو إخراجها، مثال طباعة قيمة Y

READ X

PRINT Y

الشكل (3): رمز القراءة والطباعة

صفحة 12

كما يمكن قراءة أو إدخال القيم A , B أو طباعة المعلومات M , L كما يأتي

PRINT ML

READ AB

## 3.4 رمز المعالجة Processing Symbol

وهو عبارة عن مستطيل بداخله كلمات عربية أو إنجليزية، أو أي تعبير رياضي، أو أي نوع من الاختصارات ذات دلالة. ويعني هذا الرمز إجراء عمليات حسابية أو تخزينية أو غيرها

(الشكل (4)) -

صفحة 13

مثال (2)

المعادلة الحسابية  $C = C + 1$

C يمكن تمثيلها كما يأتي:

$C = C + 1$

Increase C by 1

Calculate

$C = C + 1$

الشكل (4): رمز المعالجة

صفحة 13

نلاحظ أن الأشكال الثلاثة السابقة متكافئة من ناحية التفسير والمضمون، وهذا يعكس مدى المرونة التي تتمتع بها مخططات سير العمليات في تمثيل الخوارزميات. من ناحية أخرى يمكن تمثيل تحريك بيانات معينة من (2) إلى (X) باستخدام رست المعالجة على النحو الآتي

Move Z to X

$Z \rightarrow X$

4.4 رمز القرار Decision Symbol

بدل رمز القرار على شرط أو مجموعة شروط معينة والذي بدوره يؤثر في تغيير مسار الإجراءات التي ستنفذ حل المسألة، أو من خلال هذه الشروط يمكن تنفيذ بعض الخطوات أكثر من مرة، أو عدم التنفيذ على الإطلاق حتي يتم اتخاذ قرار ما (الشكل

شكل معين

YES

NO

الشكل (5): رمز الفرار

صفحة 13

عزيزي الدارس، اعتمادا على الشرط قد يتفرع رمز القرار إلى اتجاهين أو أكثر فمثلا إذا كانت  $X = 5$  فإن سير العمليات يتفرع إلى المسلك الأيمن (Yes) أو إلى المسلك الأيسر (No) في حال غير ذلك على النحو الآتي

NO

$$X = 57$$

أما إذا كانت طبيعة المسألة تعتمد على قيمة  $X$  فيما إذا كانت موجبة أو سالبة أو تساوي 0 فإن رمز اتخاذ القرار يكون له ثلاثة مسارات (مخارج) اعتماداً على قيمة  $X$  حيث إن سير العمليات يتفرع إلى المسلك الأيمن إذا كانت قيمة  $X$  موجبة، في حين يتجه سير العمليات إلى المسلك الأيسر إذا كانت قيمة  $X$  سالبة، وسوف يتجه سير العمليات إلى المسار الأوسط إذا كانت قيمة على النحو الآتي:

$$X=0$$

YES

NO

$$X>0$$

في حال لن \*\*

في حال لن \*

في حال أن  $=0$ 

من ناحية أخرى، عزيزي الدارس، هنالك العديد من الأمثلة على استخدام رمز القرار، وفيما يأتي مجموعة من الأمثلة التي من شأنها أن تعزز فهمك لاستخدام هذا النوع من الرموز المستخدمة في مخططات سير العمليات.

مثال (3)

إذا كان مخطط سير العمليات يعتمد على إجراء المقارنة بين  $A$  و  $B$  نهل  $A B$

إن رمز اتخاذ القرار يكون على النحو الآتي:

YES

110

$$A=B$$

من الواضح لنا أنه إذا كان جواب المقارنة السابقة نعم فإن سير العمليات سيتجه إلى المسلك الأيمن في حين سيتجه سير العمليات إلى المسلك الأيسر إذا كانت الإجابة لا.

مثال (4):

رمز اتخاذ القرار التالي يحدد سير العمليات في مخطط سير العمليات إذا كانت قيمة  $A > 5$  حيث إن سير العمليات سيتجه إلى المسلك الأيمن إذا كانت الإجابة نعم، أما إذا كانت الإجابة لا فإن سير العمليات سيتجه إلى المسلك الأيسر على النحو الآتي:

VES

NO

? $A > 5$

مثال (5):

رمز اتخاذ القرار التالي يحدد سير العمليات في مخطط سير العمليات إذا كانت قيمة  $A < 0$  حيث إن سير العمليات سيتجه إلى المسلك الأيمن إذا كانت الإجابة نعم، أما إذا كانت الإجابة لا فإن سير العمليات سيتجه إلى المسلك الأيسر على النحو الآتي:

YES

NO

$A < 0$

مثال (6):

يعتمد سير العمليات على المقارنة بين  $A$  و  $B$ . فإذا كانت  $A$  أصغر من  $B$  أو مساوية لها فإن المسار الأيمن YES هو الذي يسلكه سير العمليات. إلا أن المسار NO هو الذي سيتخذ في حالة أن  $A$  أكبر من  $B$  على النحو الآتي:

A

Yes

NO

? $A \leq B$

صفحة 15

5.4 رمز الاتصال Connections

يستخدم رمز الاتصال عادة عندما يكون مخطط سير العمليات موجودة على أكثر من صفحة، ويدل هذا الرمز على استمرارية سير العمليات، ويستخدم اما كسخرج (Out Connector) في نهاية الصفحة أو كمدخل (In Connector) في بداية الصفحة كما هو موضح في الشكل (6)

يكون الشكل دائرة

صفحة 16

Out Connector-1

in Connector

الشكل (6) رمز الاتصال

صفحة 16

6.4 رمز التكرار Iteration Symbol

بدل رمز التكرار على تكرار خطوة أو عدد من الخطوات مرات محددة. وسيوضح مدى أهمية هذا الرمز في مخططات سير العمليات من خلال دراستك عزيزي الدارس للعديد من الأسئلة في الأجزاء التالية من هذه الوحدة (الشكل (7))

الشكل (7): رمز التكرار

صفحة 16

7.4 رمز البرنامج الفرعي Subroutine Symbol

يستخدم رمز البرنامج الفرعي لاستدعاء برنامج فرعي سلسلة من التعليقات والعمليات المعرفة في مكان آخر)، وغالبا ما يطلق على هذه البرامج الفرعية مطلع الإجراءات Procedures أو الدوال Functions , وذلك في العديد من لغات البرمجة عالية المستوى مثال: لغة C التي سيتم دراستها في الوحدة الرابعة (الشكل (8)).

صفحة 16

الشكل (8): رمز البرنامج الفرعي

4.8 رمز التعليقات التوضيحية Annotation Symbol

يحتوي هذا الرمز على تعليقات و توضيحات لا تنطوي على أي إجراء تنفيذي، كأن تشير إلى نوع عملية ما أو الهدف من خلال سير العمليات، ويكون هذا الرمز متصلا مع المخطط من خلال خط مستقيم متقطع (الشكل (9)).

الشكل (9): رمز التعليقات التوضيحية

## 9.4 رموز تسلسل العمليات Flow Lines

تستخدم هذه الرموز للدلالة على تسلسل خطوات الخوارزمية. ويضاف السهم إلى الخط المستقيم إذا كان التسلسل أو التدفق باتجاه اليسار أو إلى أعلى، في حين لا يعد تحديد الاتجاه (إضافة السهم) مسألة إجبارية إذا كان التسلسل باتجاه اليمين أو إلى أسفل الشكل (10)).

الشكل (10): رمز تسلسل العمليات

مما سبق يتبين لنا ، عزيزي الدارس، أن مخططات سير العمليات توضح ما سيجري عند كتابة البرنامج. وتقع مهمة إعداد هذه المخططات على عاتق مبرمجي الحاسوب، وهي مهمة بالنسبة لك أيضا بوصفك دارسة المقرر « معالجة البيانات، حيث إن مخططات سير العمليات تهتم بتوضيح تسلسل العمليات في الخوارزمية التي بدورها تعد المحور الأساسي في كتابة تحديد سير عمل البرنامج الذي سيتم كتابته لمعالجة البيانات باستخدام الحاسوب. كما تلاحظ، عزيزي الدارس، فإن استخدام الرموز الاصطلاحية يسهل عليك وعلى الآخرين فهم خطوات حل المسألة، وبالتالي كتابة البرنامج ومخططات سير العمليات، وهي ضرورة لا غنى عنها للمبرمجين على اختلاف مستوياتهم، وهي كذلك ضرورية لك وأنت تتعلم أسس البرمجة التي ستطرق إليها كما أشرنا سابقا في الوحدة الرابعة من هذا الكتاب. ويمكنك الاستعانة بمسطرة خاصة لرسم جميع الرموز التي قرأت عنها في الأجزاء السابقة، كما يمكن رسم الرموز على ورق رسم خاص بهذه الغاية (Worksheets)

عزيزي الدارس، لا ترسم مخطط سير العمليات دفعة واحدة، بل أبدأ بإعداد مسودة للمخطط تحتوي على العمليات الأساسية، ثم وسع مخططك ليشمل جوانب المسألة كافة وإعدادك لمخطط سير العمليات تكون قد هيأت نفسك لكتابة برنامجك بلغة البرمجة التي تريد .

قبل أن تنتقل إلى أنواع مخططات سير العمليات سوف نذكر بإيجاز فوائد مخططات سير العمليات، وهي كما يأتي:

- 1- إعطاء صورة واضحة ومتكاملة عن الخطوات اللازمة لحل المسألة.
- 2- تسهيل متابعة خطوات الحل بشكل واضح، وبخاصة في حالة وجود التفرعات والاحتمالات.
- 3- تسهيل كتابة البرنامج اللازم لحل المسألة.
- 4- المساعدة على تشخيص الأخطاء التي قد تقع في البرامج وتحديد مواطن الخلل فيها.



5- تعد من وثائق المسألة المعنية ومراجعنا المهمة.

6- يمكن الاستعانة بها في حل مسائل مشابهة .

أسئلة التقويم الذاتي (1)

1- لخص مكونات مخطط سير العمليات مستعرضة أهم الأشكال الاصطلاحية.

2- بين أهمية مخططات سير العمليات وماهية حل المسائل باستخدام الحاسوب؟

صفحة 18

5. أنواع مخططات سير العمليات

عزيزي الدارس، في الجزء السابق من هذه الوحدة درسنا المكونات الأساسية (الرموز الاصطلاحية وأهميتها في مخطط سير العمليات، وفي هذا الجزء سندرس أنواع مخططات سير العمليات التي تنقسم إلى ثلاثة أنواع رئيسية هي

1- المخططات المتسلسلة (التتابعية) البسيطة (Simple Sequential Flowcharts) .

2- المخططات المتفرعة (Branchal Flowcharts) .

3- المخططات التكرارية (Loop Flowcharts)

1.5 المخططات المتسلسلة (التتابعية) البسيطة

Simple Sequential Flowcharts

تتكون المخططات التتابعية البسيطة من خطوات حل المسألة مرتبة بصورة تسلسلية. بمعنى أنها لا تنطوي على تفرعات أو تكرار للعمليات، ويبين الشكل (11) الشكل العام للمخطط سير عمليات متسلسل بسيط:

الشكل (11): الشكل العام للمخططات سير العمليات المتسلسلة البسيطة

صفحة 19

والسؤال الذي يطرح نفسه في هذه الأثناء، لندرته من خلال التدريب الآتي:

تدريب (2)

ما أصغر مخطط سير عمليات يمكنك رسمه؟

عزيزي الدارس، تستخدم المخططات المتسلسلة (المتابعة) في :

أ- حل المسائل والمهام العامة اليومية التي تعيننا على فهم المسائل الرياضية

ب- حل المسائل الرياضية والعلمية المهمة عند كتابة البرامج واستخدام الحاسوب

هل تستطيع أن ترسم مخطط سير عمليات متسلسلة بسيطة من واقع حياتك اليومية؟ حاول ذلك، ثم قارن المخطط الذي رسمته بالمخطط المبين في المثال التالي

مثال (7)

ارسم مخطط سير عمليات متسلسلة بسيطاً يبدأ بلحظة نهوضك من النوم ويصف خروجك من البيت إلى مركز الدراسي ثم عودتك إلى البيت.

يمكنك ، عزيزي الدارس، أن تجد حل هذا المثال في الشكل (12) .

20 الوحدة الأولى

الشكل (12): مخطط سير عمليات متسلسلة بسيط من واقع حياتك اليومية

صفحة 21

والآن ننتقل إلى نوع آخر من المسائل التي يمكنك وضع مخطط سير عمليات متسلسلة بسيط ليساعدك على حلها، والسؤال الذي يطرح نفسه عليك هنا هو كيف يمكنك بناء مخطط سير عمليات لحل المسألة الرياضية الآتية؟

$$Y = x^2 + 2x - 10 \quad \text{صفحة 21}$$

من الواضح، عزيزي الدارس، أنه ليس بمقدورك حل المسألة الرياضية المذكورة إلا إذا كانت قيمة  $x$  معلومة، وإذا رغبت في حلها باستخدام الحاسوب، فإنك تقوم بقراءة قيمة  $x$

صفحة 21

من خلال رمز الإدخال ، ثم تقوم بإسناد قيمة التعبير الحسابي إلى  $Y$  في ضوء المعادلة الرياضية، وبعد ذلك نطلب من الحاسوب طباعة قيمة  $Y$  والتوقف عند ذلك الحد .

نشاط (1)

ارسم مخطط سير العمليات الحسابية لقيمة  $Y$ ، وقارنه بالمخطط المبين في الشكل (13)

الشكل (13): مخطط سير العمليات الحسابية لقيمة  $Y$  للاقتراح  $x^2 + x^2 = Y - 10$

صفحة 22

## نشاط (2)

ارسم مخطط سير العمليات لإيجاد مجموع عددين مختلفين. افترض أن العدد الأول هو (A)، وأن العدد الثاني هو (B)، وأن مجموعهما يساوي (C)

ابدأ خطوات الحل بوضع خوارزمية هذه المسألة، ثم اكتب خطوات الحل وصمم مخطط سير العمليات، ويمكن تلخيص خطوات الحل على النحو الآتي

صفحة 22

و ابدأ المخطط

\* اقرأ قيمة كل من (A) و (B) .

\* اجمع العددين (4) و (B) وأسندهما إلى (C) حسب العلاقة الرياضية  $C = A + B$

\* و اطبع قيمة (C).

\* أنه المخطط

## تدريب (3)

ارسم مخطط سير العمليات لإيجاد مساحة أي مربع

## مثال (8):

أرسم مخطط سير العمليات لحساب معدلك في مقررات الحاسوب والرياضيات واللغة العربية واللغة الإنجليزية .

قبل وضع المخطط لا بد لك، عزيزي الدارس، من أهم المسألة وتحليلها. وينطوي ذلك على تحديد المدخلات (البيانات) والمخرجات (النتائج) من خلال وضع خوارزمية حل هذه المسألة وتتكون المدخلات من علامتك في كل من مقررات: الحاسوب (C)، والرياضيات (M)، واللغة العربية (A)، واللغة الإنجليزية (E)، ولنمرر إلى معدلك في هذه المقررات بالتغير V

أما خطوات حل هذه المسألة (الخوارزمية) فهي

\* ابدأ المخطط.

\* أدخل قيم المتغيرات C , M , A , E .

\* اجمع المتغيرات الواردة في الخطوة السابقة وأسند الجميع إلى D.

\* احسب المعدل بقية المجموع D على عدد المقررات، وهو هنا = 4.

\* اطبع النتيجة V.

\* أنه المخطط

صفحة 23

نشاط (3)

ارسم مخطط سير العمليات السابق وقارنه بمخطط سير العمليات المبين في الشكل (14)

الشكل (14): مخطط سير العمليات الحساب معدلك في مقررات: الحاسوب، والرياضيات  
واللغة العربية واللغة الإنجليزية

صفحة 24

أسئلة التقييم الذاتي (2)

1- اكتب الخوارزمية، وصمم مخطط سير العمليات، لإيجاد المتوسط الحسابي لثلاثة أعداد ، إذا علمت أن المتوسط الحسابي يساوي مجموع الأعداد الثلاثة مقسوم على 3.

2- اكتب الخوارزمية، وصمم مخطط سير العمليات التحويل وحدة قياس الأوزان من الكيلوجرام إلى الباوند (إذا علمت أن 1 كيلو جرام يساوي 2.2 باوند) مع العلم بأننا نريد تحويل 6 كيلوجرام إلى وحدة الباوند

صفحة 24

## 2.5 المخططات المتفرعة Bra11ched Flowcharts

إن جميع المخططات والخوارزميات التي ناقشناها في الأمثلة والتدريبات في البند السابق، كانت تعمل وفقا لتسلسل محدد يعرف بالتسلسل التتابعي؛ أي أن الخطوات تنفذ خطوة خطوة وفقا لترتيب معين، ولكن ، عزيزي الدارس، في كثير من الأحيان قد يتطلب الأمر منك أن تنفذ خطوة أو أكثر في الخوارزمية التي تم إعدادها لحل المسألة اعتمادا على شرط معين ، فلا بد لمخطط سير العمليات أن يحتوي على الرمز الاصطلاحي المسمى رم القرار ( Decision Symbol)، وذلك للمقارنة بين اختيارين أو أكثر حسب الشرط الوارد في الجدير ذكره أنه في المخططات المتفرعة سيتم تنفيذ خطوة أو مجموعة الخطوات اعتمادا على جواب الشرط الموجود في رمز اتخاذ القرار حيث نشبع بناء على جواب الشرط في حين يتم إهمال الآخر، ويمكن ، عزيزي الدارس، تقسيم المخططات المتفرعة إلى نوعين هما :

1- مخططات ينبثق عن رمز القرار فيها مساران اعتمادا على جواب الشرط كما هو موضح في الشكل (15/أ)

2- مخططات ينبثق عن رمز القرار فيها ثلاثة مسارات كما هو موضح في الشكل (15 ب) .

الشكل (15): (أ) الصيغة العامة لجزء من مخطط سير عمليات يحتوي على رمز

قرار ذي مسارين، في حين يرضع (ب) الصيغة العامة لجزء من مخطط سير عمليات

يحتوي على رامز قرار ذي ثلاثة مسارات

صفحة 25

وفيما يأتي، عزيزي الدارس، تعرض مثلاً (الشكل 16) يوضح الصيغة الواردة في الشكل

(15) السابق

مثال (2):

صفحة 26

الشكل (16)، (أ) مثال على الصيغة الواردة في الشكل (15-1) السابق حيث إنه إذا كانت تلبية

4 أكبر من قيمة B فيتم تنفيذ الخطوة  $A = C$ ، وإلا فإن الخطوة  $B = C$  هي التي تنفذ ، في حين

أن (ب) مثال على الصيغة الواردة في الشكل (15-ب) السابق. إذا كانت قيمة  $A=0$  فإن 10

$X=$  . وإذا كانت  $A>0$  فإن  $X=5A+ 10$ ، أما إذا كانت  $A<0$  فإن  $X=2A-2$

26 الوحدة الأولى

ناقشنا ، عزيزي الدارس، العديد من الأمثلة التي تعمل على حل المسائل الرياضية في البند

السابق والتي كانت تعمل وفقاً لتسلسل محدد يعرف بالتسلسل التتابعي. وفي الأمثلة الآتية

سنتناول نمطاً آخر من المسائل الرياضية تعتمد على رمز اتخاذ القرار في مخططات سير عملياتها .

مثال (10):

ارسم مخطط سير العمليات لإيجاد قيمة الدالة

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

صفحة 27

عزيزي الدارس، فكر ملياً، ثم ارسم مخطط سير العمليات الذي تراه مناسباً لحل هذه المسألة.

قارن المخطط الذي رسمته بالمخطط المبين في الشكل (17).

الشكل (17) مخطط سير عمليات حساب قيمة الدالة  $F(x)$

صفحة 27

تلاحظ في الشكل (17) أن الشرط الموجود في رمز القرار هو مئارنة قيمة  $X$  مع الصفر. فإذا كانت  $X$  أكبر من الصفر أو مساوية له فإن سير العمليات يتخذ المسلك الأيمن (أي أن  $X = F$ )، وإلا فإن سير العمليات يتخذ المسلك الأيسر (أي أن  $F = -X$ ).

تدريب 4

ارسم مخطط سير العمليات لحل الدالة الرياضية الآتية:

$$X+5, X<0$$

$$F = x1 = x2-1, X = 0$$

$$X? -5X, X>0$$

مثال 4

ارسم مخطط سير العمليات لإيجاد العدد الأكبر بين العددين  $A$  و  $B$  وطباعته الشكل (18)).

صفحة 28

الشكل (18): مخطط سير العمليات لإيجاد العدد الأكبر بين عددين وطباعته

صفحة 29

تلاحظ، عزيزي الدارس، من المثال السابق أن بيانات هذه المسألة هي  $A$  و  $B$ ، كما تلاحظ في رمز القرار (Decision Symbol) المقارنة بين  $A$  و  $B$  على شكل سؤال هو هل  $A$  أكبر من  $B$ ؟ ويكون تتابع الخطوات مرهونة بالإجابة عن هذا السؤال. وترى أننا أسندنا العدد  $A$  إلى المتغير  $X$  في الاحتمال الأول Yes (المسار الأيمن)، بينما أسندنا العدد ( $B$ ) إلى المتغير  $X$  في الاحتمال الثاني No (المسار الأيسر).

صفحة 29

### 3.5 المخططات التكرارية Iterative / loop Flowcharts

هذا هو النوع الثالث من أنواع مخططات سير العمليات، وتستخدم المخططات التكرارية عند الحاجة إلى تكرار (إعادة) تنفيذ خطوة أو مجموعة خطوات عددا محددة من المرات نختاره بالتحكم بالشرط، ونذكر هنا أن عملية انتهاء الدورة بعد عدد محدد من مرات التنفيذ، تعد عملية مهمة يجب حدوثها، فعلينا أن نكون حذرين من الوقوع في التكرار اللامتناهي، إن عمليات

التكرار من الإضافات المهمة التي تتصف بها الخوارزميات المستخدمة في معالجة البيانات، وقد يأخذ التكرار أحد الشكلين المبينين في الشكل (23):

الشكل (23): (أ) تكرار الخطوات حتى يتحقق الشرط، وهي الصيغة المستخدمة في Do - While، في لغة السي C (تكرار الخطوات حتى يصبح جواب الشرط نعم) ، في حين أن (ب) إذا تحقق الشرط (إذا كان جواب الشرط نعم) فيتم تنفيذ خطوة واحدة أو مجموعة من الخطوات.

صفحة 34

عزيزي الدارس، يمكن أن نلاحظ أن جسم الدورة (مجموعة الجمل المراد تكرارها) في الشكل (23-1) ينفذ مرة واحدة على الأقل، وذلك قبل أن نختبر الشرط، حيث يشم التكرار أو الخروج تبعاً لقيمة الشرط المنطقية (جواب الشرط نعم/لا). أما بالنسبة للشكل (23-ب) فإنه قد يحدث ألا ينفذ جسم الدورة أبداً، حيث يتم اختبار الشرط أولاً قبل الدخول إلى جسم الدورة، وفي هذه الحالة يحتمل الخروج من الدورة دون تكرار، كما يحتمل البدء بتنفيذ أول تكرار ،

صفحة 34

وهناك ما يعرف بمخططات التكرار البسيطة (Simple Loop Flowcharts) فعندما ننظر إلى الشكل السابق (الشكل 23 أ. به) نلاحظ، عزيزي الدارس، أن هناك تكرار بسيطة واحدة

سنناقش الآن مفهومًا جديدًا يتعلق بالمخططات التكرارية، و يسمى مفهوم العداد ( Counter Concept). فما هو هذا المفهوم ؟

كما في بداية هذه الوحدة قد تعرفنا على تحليل مسألة حساب مساحة دائرة نصف قطرها يساوي R، والسؤال الذي يطرح نفسه، ماذا لو أردنا حساب مساحة عدة دوائر في الوقت ذاته ورسم مخطط سير عمليات لذلك؟ هل سنرسم مخططاً بسيطاً لحساب دائرة واحدة ثم تكرار رسم المخطط نفسه في كل مرة لحساب مساحة كل دائرة على حدة ؟

إن هذا ممكن بالطبع، لكنه يغدو أمراً لا منطقية في مراحل متقدمة من دراستك، إذ بإمكانك اللجوء إلى طرق أكثر منطقية وأقل تعقيداً توفر الوقت والجهد. إذاً يمكن القول ، عزيزي الدار، أن المخططات التكرارية تختصر الوقت والجهد، وتمكنك من تحقيق الفرض واته في وقت أقصر وموارد أقل. وتعتمد المخططات التكرارية أساساً على مفهوم العداد

مثال (15):

مخطط سير العمليات التالي (الشكل 24 - أ) يستخدم مبدأ التكرار البسيط ، ويقوم بطباعة الأعداد الفردية من 1 إلى 15

عند التفكير برسم مخطط سير العمليات الطباعة الأعداد الفردية من 1 إلى 15 هنالك مجموعة من النقاط يجب تحديدها هي

1- نقطة البداية (أي القيمة الابتدائية للأعداد ألا وهي 1)

2. مقدار الزيادة في كل مرة، أي القيمة المراد إضافتها على العداد في كل مرة ليتسنى الحصول على القيمة التالية اني حالتنا هذه يجب إضافة العدد 2)

3- شرط التوقف العملية التكرار الذي من خلاله سيتم التوقف عن التكرار .

صفحة 35

يمكنك، عزيزي الدارس، ملاحظة الاختلاف بين مخطط سير العمليات في الشكل (1-24) الذي يقوم بطباعة الأعداد الفردية المحصورة بين 1 و 15 في حين أن الشكل (24-ب) يطبع الأعداد المحصورة بين 1 و 15 بشكل متسلسل، ويمكن توضيح ذلك في النقاط الآتية

- نقطة البداية هي  $k = 1$

- مقدار الزيادة هو ايد من 2.

الشكل (24) مخطط سير العمليات (1) بوضع طباعة الأعداد الفردية المحصورة بين او 15، ومخطط سير العمليات (ب) بوضع طباعة الأعداد من 1 إلى 15.

صفحة 36

مثال (16)

احسب مساحات و دوائر بأنصاف أقطار مختلفة تتكرر البيانات في هذا المثال على أنصاف أقطار الدوائر، كما أن العلاقة التي تربط المساحة بنصف القطر هي واحدة إذا لا داعي لتكرار الرموز الاصطلاحية (الأشكال التي يتكون منها مخطط سير العمليات خمس مرات. بل يكفي أن تضيف رمز الفرار على مخطط سير العمليات است كه هل نفت مجموعة خطوات حساب مساحة الدائرة خمس مرات أم لا. فإذا كان الجواب نعم، فأوقف مخطط سير العمليات، وإلا كرر حساب مساحة الدائرة التعصف قطر حر، ويبين الشكل (125) مخطط سير العمليات حساب مساحة دائرة واحدة، في حين يوضح الشكل 25 ب) احتساب مساحات خس دوائر مختلفة في سن التتر (أي في كل مرة يتم إدخال شئ نظر مختلف ) -

الشكل (25): (أ) يمثل مخطط سير العمليات لحساب مساحة دائرة واحدة، في حين يمثل اب) مخطط سير العمليات لحساب مساحات خس دوائر

صفحة 37

تحدثنا ، عزيزي الدارس، فيما تقدم عن مخططات التكرار البيط، وحن موعد الحديث عن نوع آخر من المخططات التكرارية ويسمى المخططات التكرارية المتداخلة ( Nested Loop Flowcharts). فما هذه المخططات؟

عندما يحتوي مخطط تكراري على آخر من النوع ذاته، أي تكون هناك تكرارات متضمنة بعضها في بعض ، نسمي التكرار الثاني التكرار الداخلي (Inner Loop) ، والآخر التكرار



الخارجي (Outer Loop) . أما المخطط الكلي فيسمى مخطط تكرارية متداخلا وعند استعمالك للمخططات التكرارية المتداخلة، ينبغي عليك مراعاة ألا يكون هناك تقاطع بين التكرارات المختلفة.

مثال (19)؛

ارجع إلى السؤال الخاص بحساب صافي رواتب موظفي جامعة القدس المفتوحة وأضف المعلومات الآتية

افترض أن بعض الموظفين أسسوا صندوق توفير اهمية شيرية قدرها 20 دينار اخضم هذه المساهمة من صافي الراتب للموظفين المشتركين في صندوق التوفير، وارسم مخطط سير العمليات.

عند تحليل هذا المثال تجد أن هناك بيانات جديدة لا بد من مراعاتها، وهي مقدار المساحة الشهرية في صندوق التوفير (D)

إذا ينبغي عليك التحقق من اشتراك أي موظف من موظفي الجامعة في صندوق التوفير. فإذا كان شركا ناختم مساهمته في الصندوق من صافي واتبه، وإلا تابع مخطط سير العمليات كالمعتاد، وبين الشكل (28) مخطط سير العمليات لهذا المثال :

صفحة 41

9. مسرد المصطلحات

- الإجراءات Procedures سلسلة من التعليمات والعمليات المعرفة في مكان آخر لإجراء مهمة أو أكثر

- البرنامج Program هو عبارة عن مجموعة من التعليمات التي يتم كتابتها باستخدام إحدى لغات البرمجة لتنفيذها على الحاسوب بغرض حل مسألة معينة

البرنامج المصدر Source Program مكتوب بلغة غير لغة الآلة مثل لغة

C.

- البرنامج الهدف Object Code البرنامج الناتج عن عملية الترجمة

- الخوارزمية Algorithm هي مجموعة (سلسلة من الخطوات المرتبطة منطقياً لحل مشكلة ما

- الدوال Functions سلسلة من التعليمات والعمليات المعرفة في مكان آخر لإجراء مهمة ما

- رمز الاتصال Connection Symbol: يستخدم رمز الاتصال عادة عندما يكون مخطط العمليات موجودة على أكثر من صفحة ويدل هذا الرمز على استمرارية سير العمليات

- رمز البرنامج الفرعي Subroutine Symbol يستخدم رمز البرنامج الفرعي لإستدعاء برنامج فرعي

- رمز التعليقات التوضيحية Annotation Symbol يحتوي هذا الرمز على تعليقات و توضيحات

- رمز التكرار Iteration Symbol بدل رمز التكرار على تكرار خطرة أو عدد من الخطوات مرات محددة

صفحة 45

الرمز الطرفي Terminal Symbol بدل الرمز الطرفي على بداية مخطط سيس العمليات أو نهايته.

- رمز القراءة/ الطباعة Output Symbol / Input : يدل على رمز القراءة أو الطباعة

- رمز القرار Decision Symbol بدل رمز القرار على شرط مجموعة شروط معينة والذي بدوره يؤثر في تغيير مسار الإجراءات التي سيتم تنفيذها لحل المسألة

- رمز المعالجة Processing Symbol: هو عبارة عن مستطيل بداخله كلمات عربية أو إنجليزية، أو أي تعبير رياضي، أو أي نوع من الاختصارات ذات دلالة ويعني هذا الرمز إجراء عمليات حسابية أو تخزينية أو غيرها .

- رموز اصطلاحية قياسية Standard Symbols: مجموعة رموز اصطلاحية قياسية تكون على هيئة أشكال يدل كل منها على عملية معينة، بحيث يمكن الاستدلال على أي خطوة من خطوات الخوارزمية من خلال الرمز الاصطلاحي المستخدم فيها

- رموز تسلسل العمليات Flow Lines: تستخدم هذه الرموز للدلالة على تسلسل خطوات الخوارزمية. ويضاف السهم إلى الخط المستقيم إذا كان التسلسل أو التدفق باتجاه اليسار أو إلى أعلى، في حين لا يعتبر تحديد الاتجاه إضافة السهم) مسألة إجبارية إذا كان التسلسل باتجاه اليمين أو إلى أسفل.

- طريقة التحليل Analytical Approach: الطريقة المستخدمة في حل المسائل الفيزيائية والرياضية وتستخدم فيها المعادلات الرياضية والعمليات المختلفة لإيجاد الحلول المناسبة.

- طريقة الخوارزمية Algorithmic Approach : الطريقة المتبعة لحل المسائل باستخدام الحاسوب

- طريقة هندسة البرامج Software Engineering Approach: الطريقة التي يتبعها علماء الحاسوب والمبرمجين لتحليل الاستراتيجيات واستخدامها لتصميم الأنظمة والحزم البرمجية الكبيرة وبنائها وصيانتها .

- المترجم Compiler يتم ترجمة البرنامج المصدر إلى لغة الآلة من خلال المترجم الخاص بلغة البرمجة المستخدمة
- مخططات سير العمليات Flowchart تستخدم العديد من الرموز والأشكال الاصطلاحية، وهي تشبه إلى حد ما المخططات الهندسية المعمارية لتوضيح خطوات حل أي مسألة ما .
- المخططات التكرارية Loop Flowcharts تستخدم المخططات التكرارية عند الحاجة إلى تكرار (إعادة) تنفيذ خطوة أو مجموعة خطوات عدد محدد من المرات.
- المخططات التكرارية المتداخل Nested Loop Flowcharts : عندما يحتوي مخطط تكراري على آخر من النوع ذاته، أي تكون هناك تكرارات متضمنة بعضها في البعض الآخر، تسمى التكرار الثاني بالتكرار الداخلي والآخر بالتكرار الخارجي
- المخططات المتسلسلة البسيطة Simple Sequential Flowcharts: تتكون المخططات التتابعية البسيطة من خطوات حل المسألة مرتبة بصورة تسلسلية معنى أنها لا تنطوي على تفرعات أو تكرار للعمليات
- المخططات المتفرعة Branched Flowcharts : مخططات ينبثق عن رمز القرار فيها باران أو ثلاثة مسارات اعتمادا على جواب الشرط .
- المسائل الفرعية Sub Problems: تجزئة المسألة إلى أجزاء لتسهيل حلها.

#### 47 الوحدة الأولى

##### الخلاصة

تتلخص محتويات هذه الوحدة، عزيزي الدارس، بما يلي ترجع تسمية الخوارزمية Algorithm إلى عالم الرياضيات العربي (الخوارزمي) من القرن التاسع الميلادي، الخوارزمية هي مجموعة (سلسلة) الخطوات المرتبطة منطقياً لحل مشكله ما حل المسائل باستخدام الحاسوب يمر عبر المراحل التالية: تحليل المسألة، وضع الخوارزمية المناسبة، تمثيل الخوارزميات باستخدام مخطط سير العمليات، كتابة البرنامج باستخدام إحدى لغات البرمجة، تنفيذ البرنامج وتحديثه. مخططات سير العمليات تستخدم العديد من الرموز والأشكال الاصطلاحية، وهي تشبه إلى حد ما المخططات الهندسية المعمارية، لتوضيح خطوات حل أي مسألة ما . تنقسم مخططات سير العمليات إلى ثلاثة أنواع رئيسية: المخططات التتابعية البسيطة ( Simple Sequential Flowcharts)، المخططات التفرعة سلسلة ( Loop Flowcharts Salobil)، المخططات المتفرعة ((Branched Flowcharts

7. لمحة عن الوحدة الدراسية الثالثة

تتناول الوحدة الدراسية الثالثة، عزيزي الدارس، موضوع الخوارزميات، وحل المسائل والطرق المختلفة لحل المسائل باستخدام الحاسوب، ومراحل حل المسائل باستخدام الحاسوب، وتطوير الخوارزميات وأنواع التراكيب المختلفة مثل التتابعية والتفرعية والتكرارية، مخططات سير العمليات، والرموز المستخدمة في رسم مخططات سير العمليات، وأنواع مخططات سير العمليات، وسوف ندرس نبذة عن الملفات وأنواعها وقواعد البيانات والتحليل الكينوني والرموز المستخدمة في UML في الوحدة الأولى

صفحة 46

## الوحدة الثانية

### 2. الخوارزميات و حل المسائل

ذكرنا ، عزيزي الدارس، في الوحدة السابقة أن الخوارزمية هي مجموعة نهائية من الخطوات المتتابعة لحل مشكلة ما. سميت بهذا الاسم نسبة إلى العالم المسلم الخوارزمي يستعمل هذا المصطلح في علم الرياضيات والحاسوب لوصف طريقة حل مشكلة معينة. قد يكون هناك أكثر من خوارزمية واحدة لحل مشكلة معينة، من هنا تأتي أهمية قياس ومقارنة « جذوري » أو درجة فعالية الخوارزميات المختلفة حسب مقاييس جدوى محددة. تقاس درجة فعالية الخوارزمية حسب الزمن وكمية الذاكرة اللازمين لإنجاز العملية (حل المشكلة المعطاة) كدالة تعتمد على حجم الإدخال.

### 3. طرق حل المسألة باستخدام الحاسوب

عزيزي الدارس، درست في الوحدة الثانية طرق حل المسألة باستخدام الحاسوب مثل: الخوارزميات، ومخططات سير العمليات، وهي تعد من الأساليب الأساسية والمهمة في حل المسائل باستخدام الحاسوب. وفيما يأتي نتعرف على الطرق المختلفة لحل المسائل باستخدام الحاسوب.

### الطرق المختلفة لحل المسائل باستخدام الحاسوب

هنالك، عزيزي الدارس، إجراءات وطرق متعددة، وتستخدم في حقول علوم الحاسوب لتساعدنا على ترتيب طريقة تفكيرنا وتحليلنا للمسائل المطروحة أمامنا ومن أشهرها الطرق الأربع الآتية:

### 1.3 طريقة التحليل للمسألة Analytic Approach

وهي الطريقة المتبعة بل والمستخدم في معظم الأحيان في حل المسائل الفيزيائية والرياضية، وتتميز هذه الطريقة بأنها تستخدم المعادلات الرياضية والعمليات المختلفة لإيجاد الحلول المناسبة.

### 2.3 طريقة تطوير الخوارزمية المناسبة Algorithmic Approach

وهي الطريقة المتبعة والسائدة لدى علماء الحاسوب والمبرمجين لحل المسائل باستخدام الحاسوب، وقد عرضنا ، عزيزي الدارس، هذه الطريقة في الوحدة الثانية من هذا المقرر .

### 3.3 : طريقة هندسة البرمجيات

#### Software Engineering Approach

وهي الطريقة المتبعة لدى معظم علماء الحاسوب والمبرمجين، إن لم يكن عندهم جميعهم، لتحليل الاستراتيجيات واستخدامها لتصميم الأنظمة والحزم البرمجية الكبيرة وبنائها وصيانتها (Solivare Systems inl Tack1ges) ، وأشهر هذه الأساليب طريقتان: الأولى طريقة أسفل - أعلى (Top - Bottom)، التي تعني تطوير أجزاء (Modules) محددة ومطلوبة وتصميمها، ومن ثم ضم هذه الأجزاء وتوحيدها لحل مسألة كبيرة مثل تصميم أجزاء مكتبية (Software Modules Libairies) كما هو متبع في الكثير من الأنظمة الكبيرة. أما الطريقة الثانية فهي طريقة أعلى أسفل (Down - Top)، وهي مشهورة أكثر من الطريقة الأولى، وتعني التركيز على حل المسألة الكبيرة بتقسيمها إلى وحدات أو أجزاء أصغر، ومن ثم وضع الخطة اللازمة والواضحة لتصميم هذه الوحدات أو الأجزاء. ومن الجدير ذكره، عزيزي الدارس، أن هذه الطريقة طريقة هندسة البرمجيات) سيستم دراستها بالتفصيل في مقرر «1494 هندسة البرمجيات».

عزيزي الدارس، هناك العديد من المسائل الممكن معالجتها بشكل روتيني في حياتنا اليومية، غير أن المسائل التي تواجهك بوصفك دارس ذا نمط آخر، فهي تملك صبغة منهجية والسؤال الذي قد يواجهك هو: ما الخطوط العريضة والمفاهيم الأساسية التي يجب اتباعها عند حل المسائل، والتي ستساعدك في التفكير في كيفية إيجاد الطرائق المناسبة لحل المسائل وتطويرها ؟

للإجابة عن هذا السؤال وتطوير الحل المناسب لأية مسألة باتباع طريقة ما من الطرق السابقة، يجب علينا ، عزيزي الدارس، أن نهتم بالعناصر والمكونات والخطوات الأساسية التالية لحل المسألة، حيث تشمل هذه الخطوات

#### - المسألة|The problem

وتتمثل في دراسة المسألة التي تريد حلها وفهمها بصورة جيدة.

#### 2- التحليل analysis

وتعني إعادة صياغة المسألة وتحديد المعطيات (Inputs) والمخرجات (Outputs) وعمليات المعالجة (Troocessing) .

#### 54 الوحدة الثانية

##### - الحل Solution:

صياغة طريقة الحل من خلال تطبيق مجموعة من الإجراءات (الخطوات) على المدخلات الإنتاج المخرجات، ويمكن عمل ذلك بالاستعانة بالحاسوب لتنفيذ البرنامج الذي أعد لحل المسألة، ويتطلب إعداد البرنامج تطبيق طريقة الحل ليقوم الحاسوب بتنفيذها للحصول على النتائج المطلوبة، يكتب البرنامج بإحدى لغات البرمجة، ثم يترجم إلى لغة الآلة التي يفهمها الحاسوب، ثم يقوم الحاسوب بإخراج النتائج للمسألة المطلوب حلها عند تنفيذ البرنامج.

##### - الاختبار Test:

التأكد من أن طريقة الحل المتبعة في الخطوة الثالثة للمسألة المطروحة أمامنا تنتج نتائج صحيحة.

#### 4.3 طريقة التحليل الكينوني

##### (Object-Oriented Analysis Approach (OOA

وهي الطريقة المتبعة في جميع اللغات التي تستخدم مبدأ object - oriented هنالك العديد من المصطلحات والمفاهيم الواجب معرفتها عند استخدام هذه الطريقة، وسوف نتطرق إلى هذه الطريقة بالتفصيل في القسم الثامن من هذه الوحدة .

#### الوحدة الثانية صفحة 55

##### 4. طريقة تطوير الخوارزمية المناسبة

##### Algorithmic Approach

عرفت، عزيزي الدارس، في الوحدة الثانية من هذا المقرر أن الخوارزمية (Algorithm)، عبارة عن سلسلة خطوات مرتبة منطقياً ضمن مخرطة واضحة لوصف طريقة حل يفهمها الإنسان، وأن هناك بعض الخصائص الواجب توافرها في الخوارزمية منها :

1- الوضوح والدقة في كل خطوة من خطوات الحل.

2- الترابط والوضوح في التسلسل المتلقي للخطوات،

3- محدودية الخطوات

4- تكامل الخطوات وشموليته

#### 1.44 طرق تمثيل الخوارزميات

يمكن تمثيل الخوارزميات، كما درست في الوحدة الثانية، بطريقة الرسومات والأشكال الرموز الاصطلاحية (Flowcharts) أو بطريقة تشبه لغات البرمجة (لغات الحاسوب الراقية أو العليا)، وعادة ما تعرف ب Pseudocode ، وفي البند الآتي سوف نتعرف . Pseudocode ols

#### Pseudocode 2.4

عزيزي الدارس، تعد معرفة Pseudocode الخطوة الأولى لتعلم أي لغة من لغات البرمجة. إن السؤال الذي يطرح نفسه: ما هو المقصود به pseudocode؟ إن Pseudocode هو تلخيص للخطوات الواجب اتباعها لحل أي مسألة (مشكلة) حيث تكتب بلغة سهلة تشبه إلى حد كبير لغات البرمجة الفات الحاسوب الراقية كما ذكرنا سابقاً)، وعادة ما تكون اللغة الإنجليزية (أو أي لغة منطوقة أخرى)، وبصيغة يمكن تحويلها بسهولة إلى شكل البرنامج باستخدام أي لغة برمجة، ومن الجدير ذكره أن Pseudocode لا يمكن كتابتها وتنفيذها مباشرة على الحاسوب إلا بعد تحويلها إلى برنامج باستخدام إحدى لغات البرمجة

56

#### 3.4 الثوابت والمتغيرات Constants all Variables

يستعمل الحاسوب الثوابت (Constants) والمتغيرات (Variables) لتعريف البيانات، وهي الحقائق غير المنظمة أو المعالجة (Data)، والمعلومات وهي الحقائق المنظمة أو المعالجة (Information) التي تستخدم في المعادلة حل المسألة كما ذكرنا في الوحدة الأولى. أما الثابت فهو عبارة عن قيمة عددية أو أبجدية محددة أو أي مزيج بينهما دون فراغ، ولا تتغير هذه القيمة في أثناء تنفيذ الحمل. أما المتغير فهو بمثابة اسم له دلالة يشكل من رموز أو مزيج من الرموز والأرقام دون فراغ بينهما، ويحمل قيمة عددية أو رمزية تتغير في أثناء تنفيذ الحل. إن قيمة الثابت لا تتغير في حين أن المتغير يمكن أن تتغير قيمته من لحظة إلى أخرى لكن في أي لحظة يحتوي على قيمة واحدة فقط. يمكن تصنيف الثوابت إلى نوعين كما يأتي:

- الثوابت العددية

تشتمل الثوابت العددية على الأعداد الحقيقية (Real Numbers) التي تضم بدورها الأعداد الصحيحة (Integer Numbers)، والأعداد العشرية (Decimal Numbers)، كما تحتوي الثوابت العددية على الأعداد الأسية، حيث يمكنك التعبير عن الثابت العددي باستخدام طريقة التكوين الباني (E-Notation)، وبخاصة عند استخدام الثوابت العددية الكبيرة جداً أو الصغيرة جداً، ومن الأمثلة على ذلك:  $10^{097.167} = 7.167 \times 10^{097}$  ثابت عددي من النوع الأسّي  $-7.167 \times 10^{-03} = -7.167$  ثابت عددي من النوع الأسّي. إن استخدام إشارة (-) تكون ضرورية لتوضيح أن العدد سالب في حين يتم إهمال إشارة (+) عندما يكون العدد موجباً حيث لا حاجة لظهور إشارة (1) .

## 2- الثوابت الرمزية

يتكون ثابت السلسلة الرمزية من مجموعة من الرموز المتتالية (الحروف أو الأرقام أو الرموز الخاصة بشرط أن تكون محصورة بين عاريتين " مثل " , "1234", "Computer", "Ahmed", "Is it okay", "Section 1"?") وجميعها ثوابت رمزية، حيث إن الثابت الرمزي هو مزيج من الحروف والأرقام والرموز الخاصة، ويستخدم هذا النوع من الثوابت في البرامج التمثيل البيانات والمعلومات

57

غير الرقمية مثل: الأسماء، والعناوين، وغيرها. بعبارة أخرى فهي لا تستخدم في العمليات الحسابية

من ناحية أخرى، عزيزي الدارس، فإن المتغير Variable عبارة عن اسم له دلالة رمزية لعدد أو لسلسلة رمزية، ويستخدم لتخزين أحد الثوابت، وهناك نوعان من التفسيرات: المتغيرات العددية (Numerical Variables)، والمتغيرات الرمزية (Character String Variables) يتكون المتغير العددي في العادة من حرفين مجموعين من الحروف الأبجدية (الحروف الإنجليزية) مثل (A , B , X , Axy) أو من حرف أو مجموعة من الحروف الأبجدية متبوعة برقم مثل (A1 , B3 , X1 , Axy). وهناك عدد من القواعد الأساسية يجب اتباعها عند تسمية المتغير منها:

1- أن يبدأ اسم المتغير بحرف.

2- يمكن أن يحتوي على أرقام بشرط أن يحقق البند الأول

3- لا يحتوي على فراغ، ويمكن أن يحتوي على إشارة ( \_ ) (Underscore) للفصل بين أجزاء المتغير الواحد مثال: St\_No.

4- لا يحتوي على رموز خاصة مثل \$ أو ؟

تستخدم المتغيرات العددية في تخزين الثوابت العددية، وقد تتغير قيمة المتغير العددي عدة مرات في أثناء تنفيذ البرنامج، وللمتغير الواحد قيمة محددة في لحظة ما بغض النظر عن عدد مرات التغير في قيمته. أما متغيرات السلسلة الرمزية فتستخدم التخزين الثوابت الرمزية (النص) التي كنا أشرنا إليها سابقا

4.4 التراكيب الأساسية المستخدمة في تمثيل الحل

تتوافر في حقل علم الحاسوب قوانين يطلق عليها عادة مصطلح (التجزئة)، ويجب اتباعها لتصميم حل مسألة، وأهم هذه القوانين التراكيب ما يأتي

1- الجزء عبارة عن وحدة كاملة قائمة بذاتها لها مدخل واحد ومخرج واحد ،



2- الجزء يؤدي وظيفة معينة مثل طباعة بيانات أو إدخالها .

3- الجزء قشير إلى حد معقول لتسهيل قراءته وتعديله.

4- يقرر طول الجزء مهمته وعدد الجمل اللازمة لتنفيذ هذه المهمة .

## 58 الوحدة الثانية

فيما يأتي الأجزاء المستخدمة بشكل عام في حلول أغلب المسائل على سبيل المثال لا الحصر

ا- جزء تحكم Control Module:

يظهر هذا الجزء سريان البيانات الكلي في الحل، لذلك تكون باقي الأجزاء في الحل تابعة له

2- جزء بداية Initialization Module

ينفذ هذا الجزء الجمل مرة واحدة في بداية الحل مثل: فتح الملفات، وتحديد قيم المتغيرات الأولية، وطباعة ترويسات التقارير أو عناوينها.

3- جزء معالجة البيانات Data Processing Module

يعالج هذا الجزء مرة واحدة أو عدة مرات من خلال التكرار، وتشمل هذه الأجزاء:

\* أجزاء حسابية (Calculation Modules).

\* أجزاء طباعة (Print Modules)

\* أجزاء قراءة وتحقق بيانات (Read and Data Validation Modules) .

\* أجزاء تعديل وصيانة ملفات (File Amendment and Maintenance Modules) .

4- جزء نهاية Wrap - up Module:

ينفذ هذا الجزء الجمل مرة واحدة في نهاية الحل مثل: إغلاق الملفات، وطباعة المجموع من جهة أخرى، عزيزي الدارس، هنالك شكل آخر من التراكيب التي يطلق عليها التراكيب المنطقية، وتستخدم هذه التراكيب للتأكد من أن حل المسألة يسير من جملة إلى أخرى بدلا من أن يقفز من نقطة في الحمل إلى أخرى، وتشمل هذه التراكيب 1- تركيب التابع المنطقي Sequential Logic Structure: تنفذ جمل الحل جملة بعد جملة بشكل متتابع منطقية.

## 59

2- تركيب القرار المنطقي Logic Decision Structure:

الانتقال لتنفيذ أحد خيارين جملتين محتملتين أو مجموعتين محتملتين من الجمل.

### 3- تركيب التكرار المنطقي Logic Loop Structure

تنفيذ جملة أو مجموعة من الجمل مرات عديدة .

### 4- تركيب الانتقاء المنطقي Logic Case Structure

تنفيذ جملة أو مجموعة جمل منتقاة من جمل أو مجموعات حمل متعددة .

عزيزي الدارس، إن هذه التراكيب تستخدم جملاً أو تعليمات عامة، وهي كما يأتي

أولاً: جمل الباء Start Statements

جملة البدء BEGIN / START : تستخدم هذه الجملة للبدء بتنفيذ الحل أو جزء من الحل كما هو الحال في Pseudocode ، والصيغة المستخدمة لهذه الجملة هي كما في الحالات الآتية:

- تستخدم دون اسم جزء، وتعني بداية حل المسألة كلها .

#### 1- START

- تعني بداية جزء ما اسمه (MODULE\_NAME) من حل المسألة.

#### 2- START MODULE NAME

- تعني بداية جزء ما رقمه (MODULE\_NUMBER) من حل المسألة.

#### 3- START MODULE\_NUMBER

ثانياً: حمل الإدخال Iptic Statements

جملة الإدخال ENTER / READ / INPUT : تستخدم هذه الجملة لإدخال البيانات إلى مواقع مختلفة من الذاكرة عن طريق أسماء المتغيرات المستخدمة في الجزء أو الحل

كما يأتي

- تدخل معلومة واحدة من لوحة المفاتيح على سبيل المثال إلى الذاكرة باستخدام المتغير

Variable\_name

#### 1 - INPUT Variable name

- تدخل بيانات متعددة من لوحة المفاتيح على سبيل المثال إلى الذاكرة باستخدام المتغيرات

60 الوحدة الثانية

variable name1, variable\_name2, variable namen

INPUT Variable\_name1, Variable\_name2, Variable\_name

ثالثا: حمل الإخراج Output Statements

جملة الإخراج أو الطباعة (WRITE / OUTPUT / PRINT) : تستخدم هذه الجملة الإخراج البيانات من مواقع مختلفة من الذاكرة عن طريق أسماء المتغيرات المستخدمة في الجزء أو الحل إلى الشاشة أو الطابعة كما يأتي

- تخرج بيئة واحدة إلى الشاشة على سبيل المثال من الذاكرة وهي قيمة محتوى المتغير

Variable

.name

PRINT Variable\_name -1

- تخرج بيانات متعددة إلى الشاشة على سبيل المثال من الذاكرة وهي قيم محتويات المتغيرات

Variable\_name1, Variable\_name2, Variable\_name

PRINT Variable\_name1, variable\_name2, Variable\_name -2

رابعا: حمل الإسناد Assignment Statements

حمل الإسناد Assignmient تستخدم هذه الجملة لإسناد قيمة التعبير (Expression) إلى المتغير كما يأتي:

- تسند قيمة التعبير بعد تنفيذها إلى المتغير Variable\_name

[Variable\_Name= [expression

خامسا: حمل الإنهاء Stop Statements

جملة الإنهاء EXIT / END / STOP: تستخدم هذه الجملة لإنهاء تنفيذ الحل أو جزء من الحل كما في الحالات الآتية:

- تستخدم دون اسم جزء، وتعني نهاية حل المسألة كلها .

STOP-1

تعني نهاية جزء ما اسمه (Module\_Name) من حل المسألة.

STOP Module\_Name-2

- تعني نهاية جزء ما رقمه (Module\_Name) من حل المسألة

## التركييب التتابعية Sequential Structures

يتوافر في عمل الحاسوب العديد من التراكيب أو الجمل المنطقية الأساسية منها التراكيب التتابعية، وفيها تعتمد الخوارزمية على بنية التركيب التتابعي وهو أبسط أنواع التراكيب المنطقية وأكثرها شيوعاً، ففي هذه الحالة تعتمد الخوارزمية على ترتيب جمل الحل تتابعية أو تسلسلية من أعلى إلى أسفل، كما هو موضح في المثال الآتي:

## مثال (1)

أدخل اسم موظف ما في جامعة القدس المفتوحة وعمره من لوحة المفاتيح إلى الحاسوب، ثم أطبعه على لوحة المفاتيح. سبق وأن درست، عزيزي الدارس، مثل هذا النوع من التراكيب عند دراستك للمخططات التتابعية البسيطة في الوحدة الثانية.

START -1

INPUT EMP\_NAME, EMP\_AGE -2

OUTPUT EMP\_NAME, EMP\_AGE -3

STOP-4

عزيزي الدارس، تبدأ الخوارزمية في الخطوة الأولى بجملـة START لحل المسألة وليس لجزء معين منها، وذلك لأنه لا يوجد اسم أو رقم الجزء بعد جملة البداية. الخطوة الثانية في هذه الخوارزمية تدخل اسم الموظف باستخدام جملة INPUT من لوحة المفاتيح وذلك عن طريق المتغير EMP\_NAME، والذي يدل على اسم الموظف، كذلك في الخطوة نفسها من خلال جملة INPUT تدخل عمر الموظف من لوحة المفاتيح باستخدام المتغير EMP\_AGE. ومن الجدير بالذكر، عزيزي الدارس، أنه يمكننا استخدام جملة ENTER / READ، وهذه الميزة تتميز بها Pseudocode من خلال سهولة الاستخدام. أما الخطوة الثالثة فهي طباعة اسم الموظف وعمره (EMP\_NAME, EMP\_AGE). كما يمكننا استعمال أية جملة من جمل الطباعة الأخرى مثل: WRITE PRINT، والجملة STOP تدل على نهاية الخوارزمية. كما يمكن استخدام جملة END أو جملة EXIT بدلا منها

بعد دراستنا للمثال السابق، عزيزي الدارس، يمكن ملاحظة النقاط الآتية:

1- تسلسل الخطوات المنطقية في الخوارزمية السابقة، فعلى سبيل المثال خطوة الطباعة تأتي

وتنفذ بعد خطوة إدخال البيانات وليس العكس

2- لا تبدأ أية خطوة من خطوات الحمل قبل الانتهاء من الخطوة التي تسبقها.

## 62 الوحدة الثانية

### أسئلة التقويم الذاتي (1)

- 1- اكتب خوارزمية للطباعة معدل 3 أرقام يتم إدخالها من المستخدم .
- 2- اكتب خوارزمية لحساب الرسوم الدراسية حيث إن الرسوم عدد الساعات  $\times$  سعر الساعة

### 6. التراكيب التفرعية Branch Structures

هناك صفتان من التراكيب المستخدمة في كتابة الخوارزميات وصياغتها، منها ما يعرف بالتتابعي (غير الشرطي)، ومنها ما يعتمد على شرط معين في تسلسل تنفيذ الجمل وصياغتها.

#### 1.6 التراكيب غير الشرطية

### Unconditional Branch Structures

لا يعتمد هذا النوع من التراكيب على شرط معين عند تنفيذ الجمل، فهذا النوع من الجمل يعمل وفقا لتسلسل محدد يعرف بالتسلسل التتابعي؛ أي أن الخطوات تنفذ خطوة خطوة وفقا لترتيب معين، وهنالك العديد من الأمثلة التي تعتمد على هذا الأسلوب مثل المسائل الحسابية (إيجاد مساحة دائرة إيجاد مساحة مثلث، ...).

### 2.6 التراكيب الشرطية Conditional Branch Structures

قد يتطلب الأمر منك، عزيزي الدارس، أن تنفذ خطوة أو أكثر في الخوارزمية التي تم إعدادها لحل المسألة اعتمادا على شرط معين (تنفيذ عدد من الجمل في الخوارزمية اعتمادا على شرط معين). وبعد هذا أحد الأساليب المهمة في كتابة الخوارزميات والتراكيب الشرطية، وتعني الانتقال لتنفيذ أحد خبري جملتين محتملتين أو مجموعتين محتملتين وهذا يعتمد على صياغة شرط يعتمد عليه تنفيذ (اختيار) جملة أو مجموعة من الجمل، والصيغة العامة لهذا النوع من التراكيب هي

IF <CONDITION> THEN

<STATEMENT 1>

<ELSE STATEMENT 2>

## 63 الوحدة الثانية

حيث إن الجملة رقم استنفذ فيما إذا كان الشرط صحيحة، في حين سيتم تجاهلها إذا كان الشرط غير صحيح، وسننتقل لتنفيذ الجملة رقم 2، ومن الجدير ذكره، عزيزي الدارس، أن الشرط في جملة IF (إذا) يمكن أن يكون مركبة أو بسيطة، ونعني بالمركب أن يتكون من أكثر من شرط واحد. كنا قد قمنا بكتابة مخطط سير العمليات للمثال التالي في الوحدة الثانية من هذا المقرر، وفيما يأتي نكتب الخوارزمية مستخدمين التراكيب الشرطية.

مثال (2)

اكتب الخوارزمية لإيجاد قيمة الدالة  $X| = 1$

$xif x \geq 0$

$= F = [X$

$- xif x < 0$

من الملاحظ، عزيزي الدارس، أن قيمة الدالة  $X| = F$  تعتمد في تحديد قيمتها على قيمة  $X$  حيث تكون موجبة إذا كانت  $X$  أكبر من أو تساوي صفر، وتكون سالبة إذا كانت قيمة  $X$  أقل من صفر كما يأتي

START - 1

IF  $X \geq 0$  THEN -2

OUTPUT X -3

ELSE -4

OUTPUT-X-5

STOP-6

1- اكتب خوارزمية لطباعة الأعداد الزوجية المحصورة بين 2 و 10.

2- اكتب خوارزمية الحساب قيمة الرسوم الدراسية التي سيسددها الطالب إذا علمت أن

هناك نوعان من الطلبة. الطالب المحلي، والطالب الأجنبي، حيث يدفع الطالب المحلي

15 ديناراً للساعة الواحدة، أما الطالب الأجنبي فيدفع 30 ديناراً للساعة الواحدة .

64 الوحدة الثانية

أسئلة التدوين الثاني (2)

أوجد قيمة الضريبة لكل موظف إذا علمت أن نسبة الضريبة تحدد بناء على ما

يأتي

200

\$ < الراتب فإن نسبة الضريبة = 2%.

200

\$ > = الراتب > 300\$ فإن نسبة الضريبة = 3.5%.

300

\$ > = الراتب > 500\$ فإن نسبة الضريبة = 5%.

عدا ذلك فإن نسبة الضريبة = 7.5%.

## 7. التراكيب التكرارية / Iterative loop Structures

يعد التكرار المنطقي وسيلة أو أداة مهمة جدا لكتابة الخوارزميات، ويستخدم عند الحاجة إلى تكرار أو إعادة تنفيذ جملة أو خطوة أو مجموعة جمل أو خطوات عددا من المرات، حيث لاحظنا أن جميع الخوارزميات التي تم تطويرها وكتابتها سابقة تنفذ مرة واحدة فقط.

عزيزي الدارس، قبل أن تدرس أنواع التراكيب التكرارية التي سنتناولها في الأقسام الآتية، هناك عدد من المهمات والعمليات الرئيسية التي تنجز عادة باستخدام التراكيب التكرارية وتشمل ما يأتي

العد أو الزيادة والنقصان

### Counting or Incrementing and Decrementing

تتم عملية الزيادة أو الإضافة أو العد بإضافة عدد ثابت مثل 1 أو 2 إلى قيمة محتوى المتغير، لنفرض أننا نريد زيادة محتوى قيمة المتغير Counter بمقدار ثابت وليكن يمكن أن نستخدم أي ثوابت أخرى مثل 2، 3، 20، ...). فإننا نكتب الجملة الآتية العمل هذه المهمة

Counter = Counter + 1

تنفذ هذه الجملة بإضافة قيمة 1 إلى محتوى المتغير Counter، وتخزن من جديد في المتغير نفسه (Counter)، أي أن قيمة المتغير قد زادت بقيمة الثابت وهو في هذه الحالة. تفيد هذه الجملة المبرمجين لعمل مهمة العد مثل: عد مواد أو أفراد أو درجات حرارة ، .... وكذلك الحال في حالة التناقص أو العد التناقص حيث نقص محتوى قيمة

65 الوحدة الثانية

- المسألة

المطلوب هو إيجاد معدل عمر مجموعة من الدارسين

- التحليل

تعمل على تسمية المتغيرات التي سوف نستخدمها في كتابة الخوارزمية المطلوبة وتعريفها (لا يوجد ثوابت في هذه المسألة). ففي هذه المسألة يمكننا أن نعرف المتغيرات الآتية:

مجموع أعمار الدارسين

عدد الدارسين

Sum\_of\_Ages

Number of Students

عمر الدارس

Student Age

Average\_Age

معدل أعمار الدارسين

وبعد قراءة المسألة والتمعن فيها لفهمها وتحليلها . نستطيع أن تحدد المطلوب بحساب وإظهار معدل أعمار عدد من الدارسين بالسنة (Age - Average)، أما بالنسبة إلى المعطيات اللازمة لحساب معدل الأعمار فتتخصص في معرفة عمر كل دارس، وعدد الدارسين. أما العمليات فهي تشمل جميع أعمار جميع الدارسين وعددهم بالإضافة إلى حساب معدل الأعمار.

نستطيع الآن أن نوجز فهمنا للمسألة بطريقة توثيقية وذلك بتحديد المعطيات والمخرجات والعمليات بشكل رسمي كما يأتي:

STUDENT AGE

:INPUT

OUTPUT: AVERAGE AGE

:PROCESSES

SUM OF\_AGES=SUM\_OF\_AGES STUDENT\_AGE

NUMBER\_OF\_STUDENTS-NUMBER\_OF\_STUDENTS+1

AVERAGE AGE-SUM OF\_AGES/NUMBER\_OF\_STUDENTS

3- الحل



الصباغة طريقة الحل ابتداء من المعطيات (Sturtlent\_Age) إلى المخرجات (Average Age) مروراً بإجراء العمليات الحسابية اللازمة، علينا ، عزيزي الدارس، أن نتذكر أن عدد الدارسين غير معروف؛ أي أن عدد المرات التي يجب على الخوارزمية تنفيذها غير معروف، لذلك يجب اعتماد جملة أو بنية WHILE\_END / WHILE التكرارية كما هو مطلوب في السؤال، أما عن عدد الدارسين فيجب حسابه في إحدى

## 67 الوحدة الثانية

خطوات الخوارزمية كما تعلمنا باعتماد مبدأ أو عملية الزيادة (Incrementing) . يضاف عدد واحد إلى المتغير كلما أدخلنا عمر دارس. كذلك علينا التفكير بكيفية تجميع أعمار الدارسين حين معرفتهم، وتتم هذه العملية باعتماد عملية الجمع كما وردت سابقة ، إضافة إلى ذلك علينا ، عزيزي الدارس، تحديد القيمة الأولى للمتغيرين Sunn\_of\_Ages Number\_of\_Students . كذلك يجب أن تحدد كيفية وقف الخوارزمية أو قطعها عن التنفيذ وخصوصاً أننا لا نريد تحديد عدد الدارسين، وإلا وقعنا في مشكلة التكرار اللانهائي Infinite Loop. أخيرة علينا أن نحدد ترتيب الجمل المنطقي والجمل داخل جسم التكرار وخارجه، والآن باتباع الأسلوب نفسه والطريقة نفسها نستطيع أن نكتب الخطوات اللازمة (الخوارزمية التالية، ولإيجاد قيمة معدل أعمار الدارسين نبدأ بقراءة المعطيات وإجراء العمليات الحسابية اللازمة إلى أن نحسب ونظهر المطلوب:

START -1

SUM OF AGES = 0 -2

NUMBER OF STUDENTS = 0 -3

INPUT STUDENT AGE -4

WHILE STUDENT\_AGE 0 -5

SUM\_OF\_AGES = SUM\_OF\_AGES + STUDENT\_AGE

NUMBER OF STUDENTS = NUMBER OF STUDENTS + 1

INPUT STUDENT AGE

WHILE END

AVERAGE AGE = SUM OF\_AGES / NUMBER OF STUDENTS .6

OUTPUT AVERAGE\_AGE .7

STOP .8

تبدأ الخوارزمية في الخطوة الأولى بجملة START لحل المسألة. تحدد الخطوتان الثانية والثالثة قيمتي المتغيرين SUM\_OF\_AGES و NUMBER\_OF\_STUDENTS الأولية بالصغر، أما في الخطوة الرابعة فإن الخوارزمية تقرأ عمر الدارس وتحفظه باسم المتغير STUDENT\_AGE. أما في الخطوة الخامسة فتبدأ الخوارزمية بتنفيذ جملة WHILE / WHILE END التكرارية. في بداية الجملة التكرارية تفحص الخوارزمية الشرط، وهو عدم مساواة التغير STUDENT\_AGE للصفر. فإذا كان الجواب نعم تنفذ الجمل الواردة في جسم البنية التكرارية، ويبقى تنفيذ التكرار حتى تصبح نتيجة الشرط لا. وإلا تنتقل الخوارزمية إلى تنفيذ الجمل التالية بعد نهاية التكرار

68 الوحدة الثانية

### تركيب الدوران LOp Structure / For

تعمل هذه الجملة على تنفيذ تكرار تنفيذ الجملة أو الجمل المحصورة بين بداية جملة For ونهايتها عددا معروفة من المرات، حيث تزيد أو تنقص العداد في كل مرة، والصيغة العامة لهذه البنية التكرارية هي

For counter = start value to last value steps

Statement

Loop end Counter

في بداية جملة العداد التكرارية تعين الخوارزمية أو تحدد القيم الأولية للعداد

، start value=Counter

ثم تنفذ الخوارزمية جملة أو جمل جسم التكرار مرة واحدة قبل تنفيذ نهاية الجملة التكرارية (نهاية for)، ثم تزيد الخوارزمية أو تنقص العداد بقيمة الخطوة (steps) المعرفة في الثابت أو المتغيرة عندما تكون قيمة العداد أقل أو تساوي القيمة المعرفة last value . تنفذ الخوارزمية الجملة أو الجمل المكونة لجسم التكرار مرة ثانية وهكذا، حتى تصبح قيمة العداد أكبر من القيمة المعرفة في last value حيث ينتهي تنفيذ جملة التكرار وتنفذ الخوارزمية الجملة التي تلي نهاية التكرار (Tor). لقد فرضنا أن عملية اختبار الشرط (وصول قيمة العداد إلى القيمة النهائية في نهاية جسم جملة التكرار كما في بعض لغات الحاسوب مثل لغة الباسكال، علما بأن بعض لغات الحاسوب الأخرى مثل لغة السي التي سندرس بعض أجزائها في الوحدة التالية تختبر هذا الشرط في بداية جملة التكرار. فإذا تساءلنا، عزيزي الدارس، عن قيم step , last value , start value ، فيمكن أن تكون ثوابت أو متغيرات أو تعابير حسابية تستخدم جملة العداد التكرارية في حالة أن عدد مرات التكرار معروفة

منال (1)

طور خوارزمية باستخدام For لحساب معدل أعمار خمسة من الدارسين يمكن اتباع خطوات الحل في المثال السابق لتحديد المعطيات والنتائج، لكن بخلاف المسألة السابقة فإن عدد الدارسين المراد حساب أعمارهم معروف. ولإيجاد قيمة معدل أعمار الدارسين يجب البدء بقراءة الملاحظات وإجراء العمليات الحسابية اللازمة إلى أن نحسب معدل أعمار الدارسين .

69 الوحدة الثانية

1 - Start

Sum\_of\_Ages = 0 -2

Number of Students = 0 - 3

For counter =1 to 5 step 1 -4

Input Student\_Age -5

Sum\_of\_Ages = Sum\_of\_Ages + Student Age -6

Number of Students = Number\_of\_Students +1 -7

Loop\_end counter -8

Average\_Age = Sum\_of\_Ages / Number\_of\_Students -9

Output Average\_Age -10

Stop-11

3.7 تركيب الدوران / Repeat / Stricture Until

تعمل جملة (كرر / حتى ) على تكرار ( REPEAT ) تنفيذ الجملة أو الجبل المحصورة بين REPEAT و UNTIL حتى (UNTIL) يتحقق الشرط (نعم). تأخذ هذه البنية التكرارية الشكل الآتي

REPEAT

<STATEMENTS

<UNTIL <CONDITION في بداية الجملة التكرارية تنفذ الخوارزمية الجملة أو الجمل المكونة لجسم التكرار مرة واحدة، بعدها تختبر الشروط أو الشروط لتقرر إذا ما تكرر تنفيذ جملة أو جمل التكرار أو لا. فإذا كان جواب الشرط (لا) فالخوارزمية تنفذ جملة أو جيل التكرار جميعها، ويبقى التكرار حتى تصبح نتيجة الشرط (نعم) ، أي يتحقق الشرط وتنتقل الخوارزمية بعدها إلى تنفيذ الجملة التي تلي نهاية جسم بنيسة التكرار UNTIL . تستخدم بنية REPEAT /

UNTIL التكرارية في حالة أن عدد مرات التكرار غير معروف، أر في حالة أن جمل التكرار ستنفذ على الأقل مرة واحدة، أو في حالة أن الشريط لتكرار الجمل غير معروف حتى تنفذ جمل التكرار. حتى توضح الفرق بين جملتي RTPI.AT/UNTIL و WTIILE WITILE\_IND التكراريتان سوف نقوم ، عزيزي الدارس، بكتابة خوارزمية المثال السابق باستخدام جملة UNTIL / ILITAT التكرارية.

## 20 الوحدة الثانية

### مثال (5)

طور خوارزمية باستخدام UNTIL / REPEAT التكرارية لحساب معدل أعمار مجموعة من الدارسان .

هذا المثال يشابه المثال السابق، لذلك علينا، عزيزي الدارس، أن نتبع الخطوات الرئيسة والمحددة لتطوير الخوارزمية المطلوبة في المثال السابق والأقسام السابقة والتي تشمل المسألة والتحليل، والحل حتى كتابة الخوارزمية التالية المطلوبة لإيجاد قيمة معدل أعمار الدارسين بدأ بقراءة المعطيات وإجراء العمليات الحسابية اللازمة إلى أن نحسب وتظهر المطلوب.

START - 1

SUM OF AGES = 0 -2

NUMBER OF STUDENTS = 0 -3

INPUT STUDENT\_AGE -4

REPEAT -5

SUM OF AGES = SUM OF\_AGES + STUDENT\_AGE

NUMBER\_OF\_STUDENTS = NUMBER\_OF\_STUDENTS + 1

INPUT STUDENT\_AGE

UNTIL STUDENT AGE=0

AVERAGE AGE=SUM OF AGES/NUMBER\_OF\_STUDENTS -6

OUTPUT AVERAGE AGE -7

STOP -8

44.7 تركيب الدوران المتداخلة Nestcdl Loop Structure

يمكن للجمل التكرارية أن تتداخل كما هو الحال في التراكيب السابقة، فجميع القوانين المتبعة في الجمل والتراكيب التكرارية الفريدة (غير المتداخلة) من ناحية موقع تنفيذ الشرط تتبع في الجمل التكرارية المتداخلة، كذلك يمكن تداخل تراكيب تكرارية مختلفة فمثلا يمكن أن تكون البنية التكرارية الخارجية من نوع ، بينما تكون البنية التكرارية الداخلية من نوع UNTIL / REPEAT ، وبالعكس. ولتوضيح فكرة الجمل التكرارية المتداخلة تتبع المثال WHILE/WHILE END

## 71 الوحدة الثانية

تتبع خطوات تطوير الخوارزميات الكتابة خوارزمية لإيجاد معدل علامات ثلاثة امتحانات المجموعة من الدارسين بحددهم من قبل مستخدم الخوارزمية.

تشمل الخطوات الرئيسة والمحددة لتطوير الخوارزمية المطلوبة النقاط الآتية:

- المسألة

المطلوب هو إيجاد معدل ثلاثة امتحانات المجموعة من الدارسين يحدد عددهم من قبل المستخدم .

## 2- التحليل

تعرف الثابت:

عدد الامتحانات = 3 NUMBER OF TESTS

أما بالنسبة إلى المتغيرات التي سوف نستخدمها في كتابة الخوارزمية المطلوبة فهي كما يأتي:

مجموع علامات كل دارس

عدد الدارسين

SUM\_OF\_MARKS

NUMBER OF STUDENTS

STUDENT MARK

علامة الدارس

معدل علامات الدارس

AVERAGE MARK

عداد جملة التكرار الخارجية

OUTER COUNTER

عداد جملة التكرار الداخلية

INNER COUNTER

وبعد قراءة المسألة والتمعن بها لفهمها وتحليلها ، تستطيع أن نوجز فهمنا لها بطريقة توثيقية وذلك بتحديد المعطيات والمخرجات والعمليات بشكل رسمي كما يأتي:

NUMBER\_OF STUDENTS, STUDENT\_MARK

:INPUT

AVERAGE MARK

:OUTPUT

PROCESSES: SUM OF MARKS-SUM OF MARKS+STUDENT\_MARK

INNER COUNTER = INNER COUNTER + 1

AVERAGE\_MARKESUM OF MARKS/NUMBER\_OF\_TESTS

الوحدة الثانية 72

أسئلة التقويم الذاتي (3)

1- اكتب خوارزمية لإيجاد معدل الطالب في 5 مقررات، وحدد فيما إذا كان الطالب ناجحاً أم راسب، إذا علمت أن علامة النجاح في

2- أوجد عدد الطلاب الناجحين وعدد الطلاب الراسبين في الفصل الدراسي يتكون من 30 طالبة مستخدمة الجزء (1) من التقويم ( 50% الذي

8. الملفات وقواعد البيانات

1.8 مقدمة عن الملفات وأنواعها : الملف الرئيس والحركات

Master File Versus Transaction File

إن البيانات التي تتعامل معها كثيرة ومتعددة وتختلف من تطبيق إلى آخر، ومن المستحيل معالجة هذا الكم من البيانات دون تنظيم أو ترتيب، وتلجأ في كثير من الأحيان إلى تصنيف البيانات حسب نوعها، وأحياناً ندمج بعض البيانات لنخرج بتركيبات جديدة تناسب التطبيق المطلوب، وتستخرج أحياناً جداول بنماذج مختلفة وهكذا. إن السؤال الذي يطرح هو: كيف تتم عملية ترتيب البيانات والاحتفاظ بها على وسائط التخزين المختلفة المعالجتها عند الحاجة؟

لقد تعرفت، عزيزي الدارس، من خلال دراستك لمقرر الحاسوب بالإضافة إلى الوحدات السابقة من هذا المقرر كيفية معالجة البيانات من خلال كونها موجودة داخل ذاكرة الحاسوب. لكن هناك عدة عوامل تجعل استخدام الذاكرة الرئيسية أمرا صعبا ومنها:

1- إن حجم البيانات في كثير من الأحيان يكون كبير جدا بحيث لا تستطيع الذاكرة الرئيسية استيعابه .

2- إن تكلفة التخزين باستخدام الذاكرة الرئيسية عالية جدا مقارنة بالذاكرة المساعدة لذا لا بد من تخزين البيانات على ذاكرة مساعدة كالقرص المرن (المغناطيسي)، والشريط المغناطيسي، وأقراص الليزر الضوئية، وغيرها من وسائل التخزين المختلفة. وعملية تخزين البيانات على الذاكرة المساعدة بحاجة لتنظيم معين وطريقة تخزين فعالة بطلق عليها اسم تنظيم الملفات.

### وحدة الثانية 73

عزيزي الدارس، قد يتبادر إلى ذهنك العديد من الأسئلة مثل: ما المقصود بالملف؟ ما أنواع الملفات؟ كيف يمكن معالجة الملفات؟ .... وفيما يأتي الإجابة عن هذه الأسئلة. مفهوم الملف: يعرف الملف بأنه مجموعة من السجلات التي تربطها ببعضها علاقة خاصة لخدمة غرض معين. ومثال على ذلك ملف علامات الطلاب، وملف شؤون الموظفين وغيرها، ويسمى أغلب الأحيان ملف البيانات، ومن الجدير ذكره أن الملف يتكون من مجموعة من السجلات Records. وأن السجل يتكون بدوره من عناصر أصغر هي حقول البيانات كما جاء في الوحدة الأولى من هذا المقرر. وتصنف الملفات وفقا لطبيعتها واستخدامها وطبيعة البيانات المخزنة فيها إلى الأنواع الرئيسية الآتية:

#### 1- الملف الرئيس Master File

وهو ذلك الملف الذي يحتوي على البيانات الأساسية عن موضوع معين (مجموعة من السجلات)، والتي تكون ذات طبيعة دائمة، ومن الأمثلة على هذا النوع من الملفات: ملف سجل الطالب الذي يمكن أن يحتوي على البيانات الرئيسية الآتية: رقم الطالب، اسم الطالب، تاريخ ميلاده، الجنس، الجنسية، الدين، ... ، فهذه البيانات تكون على الأغلب دائمة وغير قابلة للتغيير، ولكن يمكن تحديثها بإضافة سجل للملف أو حذف سجل من الملف أو تغيير محتوى سجل في الملف. ففي المثال السابق (ملف سجل الطلاب) يمكن حذف سجل طالب ما من الملف، ومن ناحية أخرى يمكن تغيير رقم طالب معين، وهناك العديد من الأمثلة على هذا النوع من الملفات مثل: ملف الزبائن، وملف الموردين، وملف المنتجات، وملف الموظفين وجميعها تتصف بطبيعة البيانات الدائمة.

#### 2- ملف الحركات Transaction File

وهو ذلك الملف الذي يحتوي على بيانات غير دائمة (أي تحتوي على بيانات بصورة مؤقتة). ويسمى أيضا ملف الأحداث أو ملف الوقائع أو ملف العمليات المتعلقة بموضوع معين. ومن الأمثلة على ذلك في الملف الرئيس يمكن وضع جميع السجلات المقترح إضافتها، أو حذفها، أو تغييرها في ملف منفصل يسمى ملف الحركات. ويتم غالبا في هذه الملفات تجميع البيانات

المتعلقة بالمبيعات (عمليات البيع والمشتريات (أوامر الشراء) والفواتير وحركات المخزون وغيرها. إن هذا النوع من الملفات يستخدم بشكل رئيس في ترحيل البيانات الخاصة بالأحداث أو العمليات إلى الملفات الرئيسية وذلك بعد إجراء العديد من عمليات المعالجة عليها

صفحة 74

## 2.8 : طرق معالجة الملفات

تعتمد الطرق المتبعة في معالجة الملفات على طبيعة تعريف المخرجات التي بدورها تعتمد على المدخلات التي سيتم التعامل معها في النظام كما أشرنا سابقاً من ناحية أخرى يمكن تعريف طرق معالجة الملفات بأنها الأنماط أو الأساليب المستخدمة للتعامل مع العمليات Transactions الي أنظمة المعلومات الحاسوبية، وبشكل عام يوجد نوعان من طرق معالجة الملفات هما :

### - طرق معالجة الملفات بالدفعه Batch processing

يتم وفقاً لهذه الطريقة تسجيل البيانات الخاصة بالعمليات وتجميعها بشكل دفعات مجموعات حيث تعالج بصورة دورية حسب طبيعة هذه العمليات (يومية أو أسبوعية) . تستخدم طريقة المعالجة هذه في معالجة العمليات المحاسبية (فواتير، سندات، أوامر دفع) وفي معالجة عمليات الشراء، وحركات المستودع، وعمليات البيع، وحسابات الرواتب والأجور، وغيرها .

### 2- طرق معالجة الملفات بالوقت الحقيقي Real Time Processing

وفقاً لهذه الطريقة تتم معالجة العمليات بصورة مباشرة وفور تسجيلها وإدخالها إلى النظام (ويطلق عليها المعالجة المباشرة Online Processing) وذلك بعكس أسلوب معالجة الدفعات. وتمتاز هذه الطريقة بالتحديث الفوري للملفات الرئيسية بحيث تعكس بياناتها وضع النظام الحالي يستخدم هذا الأسلوب في العديد من التطبيقات وخاصة أنظمة الحجزات في شركات الطيران بالإضافة إلى أنظمة الرادار التي تتطلب سرعة استجابة عالية. وبالرغم من أن طريقة معالجة الملفات بالوقت الحقيقي تقدم عدداً من المزايا مقارنة معالجة الملفات بالدفعه، إلا أن هذا لا يعني ضرورة استخدامها في جميع الحالات، حيث يعتمد اختبار الطريقة المناسبة في كل تطبيق على حدة في أثناء تصميم النظام أو حل مسألة معينة، ويمكن توضيح ذلك بالعوامل الآتية

- سرعة الاستجابة المطلوبة (tes) ( P1st Tiiii ) ويقصد بها أقصى مدة زمنية مسموح بها بين حدوث العملية وإدخال بياناتها إلى الحاسوب لتحديث الملفات، فعندما تكون هذه المدة قصيرة جداً (بالثواني أو الدقائق) يجب استخدام طريقة المعالجة بالوقت الحقيقي، أما إذا كانت أطول من ذلك فيمكن استخدام طريقة المعالجة بالدفعه.

75 الوحدة الثانية

2. مناجم المدخلات input valume



عندما يكون حجم المدخلات كبيرة كما هو الحال في عمليات معالجة فواتير الكهرباء أو الهاتف أو إعداد جداول الرواتب فإن معالجة الدفعات هي الطريقة الأنسب،

#### 1- طريقة استخدام الملف file using

تعد معالجة الملفات بالدفعات طريقة فعالة عندما تتطلب معالجة ملف ما الوصول إلى معظم سجلات ذلك الملف وتحديثها، فمثلا عند إعداد جدول الرواتب والأجور يتم التعامل مع سجلات جميع الموظفين في الشركة، ولذلك فإن الطريقة الأنسب لمعالجة ملف الرواتب هي طريقة الدفعات، أي تجميع العمليات والأحداث الخاصة بكل موظف (إجازات، ترقية، خصومات، عقوبات، وغيرها) وإدخالها مرة واحدة كل شهر قبل إصدار جدول الرواتب من ناحية أخرى فإن معالجة ملفات المتعاملين مع البنك تتطلب أن يتم إدخال العمليات (الأحداث) وتحديث حسابات هؤلاء المتعاملين فور حدوث هذه العمليات (الحركات)، ولذلك فإن الطريقة المناسبة هنا هي المعالجة المباشرة لهذه العمليات.

#### أسئلة التقوية الذاتي (1)

وضح الفرق بين الملف الرئيس وملف الحركات.

#### 3.8 تنظيم الملفات File Organization

عزيزي الدارس، لكي نتمكن من الاستفادة من جهاز الحاسوب والمميزات التي يوفرها في معالجة البيانات وتخزينها لا بد من إدخال البيانات إليه ليتسنى إجراء عملية المعالجة. وفي معظم الأحيان عندما يكون حجم البيانات كبيرة، فإننا نحتاج إلى تنظيم هذه البيانات على شكل ملفات. وعادة ما يتم إنشاء الملفات بغرض تخزين البيانات. تنظم البيانات في الملفات على أسس عملية وتخزن على وحدات التخزين المساعدة بحيث يسهل الرجوع إليها عند الحاجة. ومعالجة البيانات تتم بتنفيذ مجموعة من البرامج بحيث يكون الملف هو المصدر الأساسي للبيانات التي تحتاجها البرامج أثناء مرحلة المعالجة إذ يحتوي الملف على بيانات تفصيلية عن موضوع معين ولكنها قابلة للتحديث من وقت

#### 76 الوحدة الثانية

الأخر كما أشرنا في الفقرة المتعلقة بالملف الرئيس وملف الحركات. فعلى سبيل المثال ملف الموظفين في إحدى الشركات الذين يتقاضون أجورهم على حساب الساعة ويتضمن الملف الرئيس الذي يحتوي على البيانات الثابتة عن كل موظف مثل: رقم الموظف، اسم الموظف الجنس. بينما يخزن في ملف الحركات جميع البيانات القابلة للتغيير اعتمادا على عامل الزمن من أسبوع لآخر مثل: رقم الموظف (وذلك لتحديد الموظف)، عدد ساعات العمل، أجر ساعة العمل. وتسمى البيانات الخاصة بموظف واحد في هذه الشركة السجل (Record). والسجل هو الوحدة المنطقية المكونة للملف: أي أن ملف الموظفين يتكون من سجلات الموظفين على أساس وجود سجل واحد لكل موظف. ويتكون السجل من مجموعة من العناصر (Items) أو الحقول (Fields) كما أشرنا في الوحدات السابقة. يطلق على الطريقة التي ترتب بها السجلات داخل الملف اسم تنظيم الملف. وتوجد عدة طرق لتنظيم الملفات منها:

## 13.8 الملفات التتابعية Sequential Files

تعد هذه الطريقة من أقدم الطرق المستخدمة في ترتيب البيانات، حيث توضع البيانات بشكل متجاور داخل الملف. ترتيب السجلات في هذا النوع من الملفات بشكل تتابعي، سجلاً تلو الآخر حسب تسلسل إدخالها للحاسوب طبقاً لمفتاح السجل (Recoral Key) (مفتاح السجل: هو حقل أو مجموعة من الحقول تحافظ على عدم تكرار البيانات حيث إن كل سجل يكون له مفتاح وحيد، مثال على ذلك رقم الطالب الجامعي). إن السجلات التي تريد إضافتها في هذا النوع من الملفات تضاف إلى نهاية الملف دائمة، وبالتالي فإن استرجاعها أو قراءتها من مكان التخزين ووحدة التخزين (المساعدة) سوف يتم بالتسلسل نفسه الذي تمت به عملية الإضافة. وهذا يعني أنه لو كنا نتعامل مع ملف يحتوي على 280 سجلاً وأردنا قراءة السجل (26) فإننا لا نستطيع الوصول إلى السجل الذي نريده إلا بعد قراءة جميع السجلات السابقة وعددها 259؛ أي أن طريقة الوصول إلى السجلات تتم بطريقة متسلسلة (تتابعية) فقط.

الوحدة الثانية 77

مثال 7 صفحة 78

من الواضح، عزيزي الدارس، أنه حتى يتسنى لنا الوصول إلى السجل رقم 3 يجب قراءة السجلات أو 2 ويستخدم هذا التنظيم عادة عند تخزين الملفات على وحدات التخزين التتابعية مثل: الأشرطة المغناطيسية (Magnetic Tapes). كما يمكن أن يستخدم التنظيم التتابعي في تخزين الملفات التتابعية على وحدة الأقراص المغناطيسية، ويطبق هذا التنظيم في أسلوب المعالجة على شكل دفعات (Batch Processing)، ولا يستخدم هذا التنظيم في أسلوب معالجة الملفات بالوقت الحقيقي بسبب بطء استرجاع سجل معين في الملف التتابعي الذي يحتوي على مئات السجلات

هنالك بعض المميزات والعيوب الموجودة في التنظيم التتابعي يمكن توضيحها كما

### • مزايا ملفات التنظيم التتابعي

1- كثافة التخزين حيث تخزن السجلات دون أي فراغ بينها، الأمر الذي يساعد على توفير في مساحات التخزين

2- سهولة معالجة الملف التتابعي

78 الوحدة الثنية

### • عيوب ملفات التنظيم التتابعي

1- يستخدم الملف التتابعي في الحالات التي تجري فيها عمليات المعالجة على كل السجلات أو على أغلبها إذ يتم قراءة جميع السجلات في الملف التتابعي بصرف النظر عن عدد السجلات المطلوب التعامل معها .

2- يجب فرز سجلات الملفات قبل إجراء عملية التحديث وفقا لفتح السجل وحسب ترتيب سجلات الملف الرئيسي

3- تحتاج إلى ملفات متعددة في عملية الفرز أو التحديث

### 2.3.8 الملفات التتابعية المفهرسة Indexed sequential file

ذكرنا في البند السابق، عزيزي الدارس، أن استرجاع البيانات (السجلات) في الملف الذي يعتمد طريقة التنظيم التتابعية يحتاج إلى وقت كبير وبخاصة في حالة استرجاع محلات موجودة في نهاية الملف، حيث يتطلب ذلك المرور على جميع السجلات الموجودة قبل هذا السجل. ومعنى ذلك أن هذا النوع من التنظيم لا تتوافر فيه إمكانية استرجاع السجلات بشكل مباشر، وسنتعرف الآن على طريقة أخرى لتنظيم الملفات لحل هذه المشكلة، وهي طريقة الملفات التتابعية المفهرسة.

تعد هذه الطريقة من أشهر الطرق وأكثرها استعمالاً، حيث تمكنا من الوصول إلى السجلات الموجودة داخل الملف التتابعي المفهرس بطريقة مباشرة تتابعية، وذلك باستخدام حقل يسمى مفتاح السجل (Key ، Recor)، وفي الملفات التتابعية المفهرسة تخزن حسب السجلات في ملف دي فهرس (جدول)، وعادة تكون السجلات مرتبة داخل الملف المفهرس ترتيباً تصاعدياً أو تنازلياً حسب قيمة مفتاح السجل، ويقوم نظام الملفات المفهرسة بتكوين فهرس لهذه السجلات، ويتكون من قيمة مفتاح السجل، وعنوان السجل المخزن على وحدة الأقراص المغناطيسية. وعنوان السجل يتكون من :

- رقم الأسطوانة clinder number

- رقم الوجه surface number

- رقم السجل على الوجه record number

والشكل (1) يوضح الفكرة العامة لتنظيم السجلات في الملفات المفهرسة التتابعية

### 79 الوحدة الثانية

الشكل (1) : تنظيم السجلات في الملفات المفهرسة التتابعية

المجموعات هي

كما هو موضح في الشكل السابق فإن البيانات السجلات مرتبة ترتيباً تصاعدياً بناءً على مفتاح السجل، وهو في هذه الحالة رقم البضاعة. ويتم تقسيم السجلات في الملف إلى مجموعات متساوية، وفي المثال السابق قسمت إلى مجموعتين متساويتين كل مجموعة تتكون من ثلاثة سجلات، وتعطى كل مجموعة رقماً وفي هذه الحالة أرقام ويتم تكوين الفهرس لهاتين المجموعتين من 3 و 6 حيث تشير هذه الأرقام إلى المجموعات 01 و 02، والرقم 3 هو مفتاح

السجل الأخير في المجموعة الأولى في حين أن الرقم 6 هو مفتاح السجل الأخير في المجموعة الثانية.

وعندما نريد سجلاً معينة من السجلات المذكورة آنفاً وليكن السجل الذي يحمل المفتاح رقم 5 فإن الاسترجاع يتم كما يأتي:

1- نقارن مفتاح السجل رقم 5 بالرقم 3 في الفهرس، فنجد أن الرقم 5 أكبر من الرقم 3 وبالتالي فإن السجل المطلوب غير موجود ضمن المجموعة الأولى

2- نقارن مفتاح السجل نفسه (مفتاح السجل رقم 5) بالرقم 6 في الفهرس فنجد أن الرقم 5 أقل من الرقم 6، وبالتالي فإن السجل الذي يحمل مفتاح رقم 5 موجود ضمن المجموعة الثانية، وفي هذه الحالة تقوم بقراءة سجلات المجموعة الثانية تتابعية حتى تصل إلى السجل المطلوب وهو السجل الذي يحتوي على البيانات الآتية: رقم البضاعة = 5، والكمية الموجودة = 520، والكمية المطلوبة : 170

## 80 الوحدة الثانية

عزيزي الدارس، إن أهم مزايا ملفات التنظيم التتبعي المفهرس هي إمكانية الوصول إلى السجلات بالطريقة التتبعية أو الطريقة العشوائية كما هو في المثال السابق حيث إن الجزء الأول من عملية الوصول الحديد المجموعة التي تحتوي على السجل) بعد وصولاً عشوائية، في حين أن الجزء الثاني هو تتبعي

### • مزايا الملفات التتبعية المفهرسة:

1- يمكن تداول السجلات بطريقة التتابع أو بالطريقة العشوائية

2- سهولة كتابة هذا النوع من الملفات

3- سهولة إعادة تنظيم الملف.

عيوب الملفات التتبعية المفهرسة

1- يؤدي إضافة سجلات إلى التنظيم المفهرس إلى تقليل معدل التجاوب .

2- يحتاج الملف المفهرس إلى إعادة تنظيم ضمن فترات زمنية معينة وذلك بسبب المساحة المحدودة للسجلات الفائضة .

3- يخزن الملف في وحدات التخزين المساعدة وليس في ذاكرة الحاسوب الرئيسية، الأمر الذي يتطلب قراءة الفهرس ونقله إلى ذاكرة الحاسوب الرئيسية عند البحث عن أي سجل بالإضافة إلى ذلك قراءة السجل نفسه بعد البحث عن عنوانه (أي أننا نحتاج إلى عمليتي قراءة: قراءة عنوان السجل وقراءة محتويات السجل نفسه)

### 3.3.8 الملفات التحتية hashing file

يعتمد نظام الملفات التحتية hashing file على تخزين السجلات في أماكن تحدد بناء على علاقة رياضية يتم اختيارها من قبل مصمم الملف المبرمج)، وبالتالي يمكن الوصول للسجلات في الملفات التحتية اعتمادا على العلاقة الرياضية التي تم استخدامها في تحديد مكان تخزين السجل (الشكل (2)).

عزيزي الدارس من خلال دراستك لأسلوبين من أساليب تنظيم الملفات (المتابعة والتتابعة المفهرسة)، وبناء على أسلوب عمل الملفات التحتية، يمكن القول إن هناك فرقين أساسيين بين هذا النوع من الملفات والملفات التتابعية هما:

#### 81 الوحدة الثانية

1- تخزن السجلات في الملفات التتابعية واحدة تلو الأخر، بينما تخزن السجلات في الملفات التحتية بناء على علاقة رياضية.

2- يتم الوصول إلى أحد السجلات المخزنة وفق النظام التتابعي بعد المرور على جميع السجلات التي سبقتها بينما يتم الوصول إلى هذا السجل بطريقة مباشرة في الملفات التحتية (الشكل (2): تحديد عنوان السجل اعتمادا على مفتاح السجل باستخدام علاقة رياضية معينة

صفحة 82

#### 4.3.8 الملفات المباشرة Direct Files

تعتمد الملفات المباشرة في ترتيب السجلات في الملف على إيجاد علاقة بين مفتاح السجل والمرضع (الموقع) التخزيني للسجل. فالسجل الذي يكون قيمة مفتاحه 9 سوف يخزن في الموقع التاسع في المساحة المخصصة للملف على وسط التخزين، في حين أن السجل الذي يكون قيمة مفتاحه 67 سوف يخزن في الموقع السابع والستين في المساحة المخصصة للملف على وسط التخزين، وفي بعض الأحيان يطلق على هذا النوع من الملفات التنظيم النسبي؛ فإذا كان مفتاح السجل يتكون من منزلة واحدة فإن الملف يحتوي على عشرة مجلات بحيث تقع قيمة مفتاح السجلات ما بين صفر وتسعة، وبالتالي فإن قيمة السجل تعني ترتيب موضع السجل في المساحة المخصصة للملف على وسط التخزين .

وبناء على ما سبق من مفاهيم الملفات المباشرة يمكن القول إن هناك فرقا أساسية واضحة بين هذا النوع من الملفات والملفات التحتية التي سبق الحديث عنها. وبكمن هذا الفرق، عزيزي الدارس. في طبيعة توزيع السجلات داخل المنطقة المخصصة للملف على وسط التخزين المستخدم

مزايا ملفات التنظيم المباشر

1- لا تحتاج التعديلات إلى فرز قبل التنفيذ.

#### 82 الوحدة الثانية

2- يتم استرجاع سجل معين بصورة مباشرة دون قراءة أو استرجاع أي مجالات أخرى في الملف كما هو الحال في الملفات التتابعية

3- يمكن التعامل مع أكثر من ملف واحد وتعديل هذه الملفات في الوقت نفسه .

عيوب ملفات التنظيم المباشر

1- لا يستغل وسط التخزين الاستغلال الأمثل بسبب الفراغات التي تترك بين السجلات للسجلات غير المدخلة

2- تكون عمليات المعالجة بكفاءة أقل بسبب التشابك بين سجلات الملف الواحد ،

تدريب (2)

ما المقصود بطرق تنظيم الملفات

أسئلة التنويم الذاتي (5)

ملخص مزايا و عيوب كل نوع من أنواع الملفات.

4.8 قواعد البيانات DataBases

تطور استخدام الحاسوب تطورا سريعة في السنوات الأخيرة، وقد شمل هذا التطور الأفراد والمؤسسات سواء الحكومية أو الخاصة. وكانت البرامج التي تتعامل مع قواعد البيانات وإدارتها من أهم العناصر التي تطورت تطورا سريعة.

كانت قواعد البيانات تستخدم في نطاق محدود على مستوى الشركات التنظيم المعلومات الخاصة بالمخازن كهدف أساسي، لكننا في هذه الأيام نعيش عصرا أصبحت المعلومات فيه كالماء والهواء تصل عبر الأثير وأنت جالس إلى جهاز الحاسوب، فإذا كنت مشتركة في إحدى الشركات الحاسوبية الواسعة فإنك تستطيع الاتصال بالعديد من قواعد البيانات التي تحتوي على الأخبار اليومية، ودوائر المعرفة من مكتبات وبنوك المعلومات، وأسعار البورصة، كما يمكنك أداء الكثير من الخدمات دون أن تغادر مكانك مثل التعامل

83 الوحدة الثانية

مع حجوزات المطارات والفنادق وشراء السلع المختلفة، علاوة على مراجعة حسابك في البنك عزيزي الدارس، يحتاج كل تاجر إلى بناء قاعدة بيانات الإدارة مخازنه من مشتريات ومبيعات، ويحتاج الطبيب إلى قاعدة بيانات لمرضاه، والمحامي لموكليه، ....

ومع ازدياد حجوم البيانات المتسارعة، ازدادت كفاءة معدات أجهزة الحاسوب، وتطورت كفاءة البرامج المستخدمة في بناء قواعد البيانات وإدارتها، سواء أكانت على مستوى المستخدم الواحد أم على مستوى الشبكات

## 1.8 مقعدة لقواعد البيانات و مميزاتها

برزت أهمية استخدام نظم إدارة قواعد البيانات في المؤسسات والشركات الكبيرة من أنها توفر سيطرة مركزية على حفظ البيانات واستخدامها وإدارتها مقارنة بالأسلوب القديم أنظمة الملفات البسيطة التي لا توفر هذه الميزة، ويمكن إيجاز الأهمية التي تتمتع بها قواعد البيانات بالهدف الأساسي لوجودها وهو توزيع أو تقسيم البيانات إلى مجموعة من الجداول مترابطة منطقية فيما بينها. للوصول لهذا الهدف يوجد أسس وقواعد يتم اتباعها لإنجاز هذه العملية، وفيما يأتي توجز بعض الفوائد والخصائص الرئيسية لاستخدام قواعد البيانات:

1. تجنب التكرار غير اللازم للبيانات المخزنة (Retdunclancy) .

2- سهولة المشاركة في استخدام البيانات (Slaring of Dala1) .

3- تحسين الشروط الأمنية ( Security ) .

4- التأكد من صحة البيانات (Integrity) .

5. تحسين دقة توافقية البيانات (Consistency)

تعريف قواعد البيانات (Databases) DB قواعد البيانات (Database ) تعني جميع البيانات ذات العلاقة التي غالبا ما تخص مؤسسة أو مشروعا متكاملأ أو شركة كبيرة، وبعبارة أخرى يمكن تعريف قواعد البيانات على أنها «مجموعة من البيانات المعلومات المرتبطة منطقية موضوع معين» من خلال هذه البيانات المعلومات)

## 84 الوحدة الثانية

1- مشتركة بين مجموعة من المستخدمين

2- مسؤولة عن تعريف نفسها

3. ممكن تشيع المعلومات من خلال مصادر متنوعة بحيث يمكن ترتيب هذه البيانات المعلومات وتنظيمها

عزيزي الدارس، تعني بكلمة البيانات (Data) : مجموعة من الحقائق التي يمكن تسجيلها لتوضيح شيء معين. مثال: رقم الهاتف، العنوان، اسم الطالب .... وتعني كلمة المعلومات (Information) البيانات بعد إجراء عملية المعالجة لها . الأهداف الأساسية لاستخدام نظام قواعد البيانات هو تبسيط تصميم الأنظمة وتطويرها وبرمجتها وتطبيقها من ناحية، وتقليل المدة الزمنية اللازمة لذلك من ناحية أخرى، ومن خلال أنظمة قواعد البيانات تستطيع المؤسسات تحويل أنظمتها من الأسلوب اليدوي إلى الأسلوب الآلي في مدة قصيرة مقارنة بالأسلوب القديم المبني على استخدام الملفات

## الكيانات والخصائص Attributes & Entities

تعرق قواعد البيانات على أنها مجموعة من البيانات المترابطة منطقية لغرض تلبية حاجة المستخدمين لهذه البيانات بكل دقة وكفاءة وسرعة، تقوم أنظمة قواعد البيانات بالعديد من العمليات منها: البحث، والإنسانية، والتعديل، والحذف. تستند قواعد البيانات في مكوناتها على مفهوم النموذج الذي تعد الكيانات Entities جزءاً رئيسية منه. ولكل كيان مجموعة من النصائح Attrillites

مثال (8)

الطالب عبارة عن كيان في حين أن المعلومات الخاصة به عبارة عن الخصائص (الصفات)

الطالب اسم الطالب، رقمه، الحالة الاجتماعية، العنوان التخصص، ،،،، (٠٠)

,Student (Student Name, Student number, Marital Satus, Address

(... ,Major

85 الوحدة الثقبه

ولكل كيان قيم متعددة قسمى عنصره، تشترك كلها بالصفات ولكنها تختلف بالقيم، ومثال ذلك

الطالب (عبدالله 935202, أعزب، غزة، حاسوب) .

الطالب (حسن، 132001، متزوج، باليا ، رياضيات)

أمثلة للكيانات وبعض خصائصها :

المقررات (رقم المقرر، اسم المقرر، عدد الساعات) .

القسم (رقم القسم، اسم القسم، العنوان، رقم الهاتف ) .

عزيزي الدارس، إن الكيان يمكن أن يكون عبارة عن شخص، أو مكان، أو جهاز ما . أو بضاعة .....

المفاتيح العلائقية Relational Keys

Super Key - 1

هو عبارة عن حقل أو مجموعة من الحقول تحافظ على عدم تكرار السجلات في الجدول.

مثال (9)



أفرض وجود قاعدة بيانات للقبول والتسجيل في الجامعة تحتوي على جدول الطالب ( SI , Address ) , ( Number, SL Name, Gender, Juill wla JULI Jy's Spix (Student Major , Tel\_Nunber . نلاحظ هنا أن رقم الطالب ( St\_Number ) يصلح أن يكون Super Key ، حيث يحافظ على وحدانية السجلات (عدم تكرار) من خلال القيم التي يمكن أن يحتويها هذا الحقل. لا يمكن تخصيص رقم الطالب لأكثر من طالب واحد.

(Super Keys (Student

,St\_number - 1

,St\_number+ St\_name-2

,St\_number+ Gender -3

,St\_numbert Address -4

,St number+ Tel number-5

,St\_number+ Major-6

,St\_number+ St\_name+ Gender -7

-8

86 الوحدة الثانية

ج- المفتاح المرشح Candlicate Key

هو عبارة عن Super Key ، لكن لا يوجد أي جزء منه يصلح أن يكون Super Key

مثال (10)

من المثال السابق نجد أن St\_number يصلح أن يكون Candidate key ، فهو يتكون من حقل واحد، وهو عبارة عن Saiper Key. لكن لو نظرنا إلى الاحتمالات الأخرى فسنستبين أنه لا يوجد أي منها يصلح أن يكون Canaliciate Key . فعلى سبيل المثال St\_nuinbert St\_naine طبقاً لتعريف Candidate Key يجب أن يحقق خاصيتين

- أن يكون Super Key.

- لا يوجد أي جزء منه يصلح أن يكون Super Key.

نلاحظ أن St\_Name St\_Number يصلح أن يكون Super Key. ولكن أن St\_name لا يصلح أن يكون Super Key ؛ لأنه من الممكن أن يشترك أكثر من طالب في الاسم نفسه، أما

St\_nuinner يصلح أن يكون Super Key لهذا فإن #St\_name لا يمكن أن  
يمتاز بخاصية Candidate Key

### 3- المفتاح الرئيس Primary key

هو عبارة عن ال Candidate Key الذي تم اختياره جدول معين أثناء عملية التصميم  
للمحافظة على عدم تكرار السجلات في الجدول

### 4- المفتاح الاجنبي Foreign key

هو عبارة عن حقل أو عدة حقول موجودة في جدول ما على شكل Primary Key وهذا الحقل  
نفسه، موجود في جدول آخر لكنه ليس Primary Key.

### 5- المفتاح البديل Alternate Key

هو عبارة عن Candidate Key لجدول معين والذي لم يتم اختياره ليكون Primary Key  
لهذا الجدول.

## 87 الوحدة الثانية

### 6 - المفتاح المركب، (Composite Key)

وهو أي نوع من الأنواع السابقة يتكون من أكثر من حقل واحد

عزيزي الدارس، تذكر أن الجدول يحتوي على Trinary Kay واحد فقط، لكنه من الممكن أن  
يحتوي على أكثر Alternate Key واحد. Super Key, Candidate Key, Foreign Key, in

## العلاقات Relationships

ذكرنا سابقا أن قواعد البيانات هي مجموعة من البيانات المرتبطة بعضها ببعض بشكل منطقي،  
وأن الكيان هو نموذج أو مجموعة من المعلومات تحدد بمجموعة من الختان. يمكن تمثيل أي  
نظام مجموعة من الكيانات؛ فمثلا كيان الطالب وكيان المدرس وكان المساقات يشكلون نظام  
الجامعة، ويطلق على الهيكل العام الذي يرتبط هذه الكيانات العلاقة Relationship ، وعلى  
عدد من العلاقات مجموعة العلاقات ترتبط البيانات بعضها ببعض مجموعة من العلاقات حيث  
يمكن تصنيف العلاقات على النحو

### 1- واحد إلى واحد One To One 1:1

ترتبط هذه العلاقة بين كيانين على أن يرتبط كل عنصرين (عنصر من كل كيان) من عناصر  
البيانات معاً بحيث يتم تحديد قيمة العنصر الثاني عندما تكون قيمة العنصر الأول معلومة عند  
لحظة زمنية معينة. بعبارة أخرى: إذا كان الكيان A مرتبطاً بالكيان B فإن كل قيمة (عنصر)  
من قيم الكيان A ترتبط بقيمة واحدة (عنصر) من قيم الكيان b

نعني بهذا النوع من العلاقات أنه يوجد لكل سجل ( Record ) في الكيان الأول سجل واحد فقط في الكيان الثاني. فمثلاً: إذا كان الكيان A يمثل المسابقات في قسم الحاسوب، والكيان يمثل المدرسين 11 يقابله A2 B1 يقابله 12، وهكذا. (المساق المقرر يدرس سه مدرس واحد، وكل مدرس يدرس مقررة واحدة).

## 2- واحد إلى متعدد one to many 1:m

هذا النوع من العلاقات أكثر الأنواع استخداماً في الحياة العادية وفي معظم نظم قواعد البيانات. نعني بهذه العلاقة أنه يوجد لكل سجل (1,Record) في الجدول الأول سجل واحد أو أكثر في الجدول الثاني

## 88 الوحدة الثانية

## 3- متعدد إلى متعدد many to many m:m

مثل هذه العلاقة بين كيانين في حالة وجود علاقة عكسية حيث أن كل سجل في الكيان الأول يقابله (اله) عدد من السجلات في الكيان الثاني، وكل سجل في الكيان الثاني يقابله (اله) عدد من السجلات في الكيان الأول

## تعريف: العلاقة Relationship

هي الربط بين كمانين أو أكثر مختلفين في بياناتها (تفاصيلها) لكنهما يشتركان في جزء محدد من هذه البيانات.

## منال 11 صفحة 89

## 89 الوحدة الثانية

عزيزي الدارس، يتضح لنا من الأمثلة السابقة وجوب وجود حقل مشترك بين أي حد ولن يتم إنشاء علاقة بينهما

## 248 برمجيات إدارة نظم قواعد البيانات

## Database Management System Software

عزيزي الدارس، لكي نتمكن من تصميم قاعدة البيانات وبنائها بجد تحديد واختيار البرمجيات والتجهيزات الحاسوبية التي سيتم استخدامها لبناء قاعدة البيانات وتقسيمها ويتم بناء قواعد البيانات وتشغيلها وتحديثها باستخدام برمجيات حاسوبية خاتة تسمى أنظمة إدارة قواعد البيانات (Datalhase Malangenient Systems ) DBMS التي تقوم بتعريف البيانات المخزنة في قواعد البيانات وتنظيمها ومعالجتها وتخزينها واسترجاعها، وتزود هذه الأنظمة البرمجية المستخدمين بأدوات فعالة لإضافة البيانات وحدها وصيانتها وعرضها ، والبحث في قواعد البيانات والتعديل فيها

## (Database Management System (DBMS

هو مجموعة من البرامج التي تقوم بإدارة البيانات ومعالجتها بطريقة سهلة وسريعة والهدف من نظام إدارة قواعد البيانات والعمل على جمع البيانات الخاصة بالمؤسسة وتصنيفها وحفظها ، واسترجاع المعلومات الصحيحة عند الحاجة، ومن الأمثلة على نظم قواعد البيانات DIMS نظام تسجيل الطلاب، وأنظمة البنوك،... هذه البرامج تتم تمكنا من إنشاء البيانات واسيانتها ومعالجتها .

ومن الأهداف الأساسية لنظام ادارة قواعد البيانات: إنشاء قواعد البيانات، وتزويد المستخدم بواجهة تطبيق مناسبة، وترتيب البيانات وتنظيمها بشكل مناسبة بحيث دكن المستخدم من تخزين البيانات واسترجاعها بسهولة وبأقصى سرعة ممكنة. يتم التعامل مع أنظمة إدارة قواعد البيانات من خلال الواجهات Interfaces التي توفرها هذه الأنظمة لكي يتمكن المستخدم من التعامل معها، فهي تمكن المستخدم من إنشاء البنية أو الهيكل المنافي الذي يستخدمه التخزين البيانات، وتمكنه أيضا من ادخال هذه البيانات وتحريرها وصيانتها وتحديثها واسترجاعها بالطريقة والشكل المناسب

### 90 الوحدة الثانية

ناحية أخرى، عزيزي الدارس، هنالك العديد من الحزم البرمجية التنوعة المستوى، فمنها الحزم المستخدمة في الحواسيب الشخصية، وهي بسيطة التكوين وسهلة الاستخدام ومصممة للمستخدم العادي غير المتخصص في مجال الحاسوب والبرمجة، بينما توجد حزم أخرى ذات إمكانات كبيرة يمكن برمجتها وأتمتة العديد من التعليقات بوساطتها ، وهي مصممة لتستخدم كأداة لتطوير التطبيقات الإدارية المختلفة بكفاءة وإنتاجية عالية، ومن الأمثلة المشهورة على هذه الحزم، حزمة Paradox لشركة Borland، وحزم Microsoft Access 7, Access 2000, FoxPro المستخدمة مع بيئة البرمجة Delphi، وبالإضافة إلى هذه الحزم المذكورة أعلاه هناك حزم أخرى تستخدم للتطبيقات الكبيرة مثل حزمة أنظمة إدارة قواعد البيانات Informix وحرمة Oracle ، وغيرها، وتمتاز هذه الحزم الأخيرة بكونها تعمل في بيئات التشغيل المختلفة مثل Unix بالإضافة إلى إمكانية عملها في بيئات تشغيل أخرى مثل: Windows NT و Windows 95, 98, 2000. والشكل (3) يوضح دور أنظمة إدارة قواعد البيانات كواجهة ربط البرمجيات التطبيقية مع أنظمة التشغيل

الشكل (3) أنظمة إدارة قواعد البيانات كواجهة ربط البرمجيات التطبيقية مع أنظمة التشغيل

### صفحة 91

### 3.4.8 تنظيم نظم إدارة قواعد البيانات

يمكن. عزيزي الدارس، تقسيم قواعد البيانات لبقعة للطريقة التي تنظم بها البيانات على النحو الآتي

## - قواعد البيانات الهرمية hierarchical databases

بسم التنظيم الهرمي في بعض الأحيان التنظيم الشجري Tree Structure (ir) وتستخدم قواعد البيانات الهرمية بشكل كبير وفعال مع المراسيمه الكبيرة ذات الحجم الكبير للذاكرة ورسائط التخزين، وتتطلب إحدى اللغات الراقية لبرمجتها وتخزن البيانات على شكل تركيب هرمي في العقد حيث تنمو الشجرة معمودية ، وقد يتفرع من العقدة Notle فروع متعددة تنتهي بعقدة وليدة، وتسمى العقدة المولدة الوالد Parent والجديدة الأبناء Children ، ويكون لكل ابن والد واحد فقط. أما الوالد فقد يكون له أكثر من ابن

ترتبط كيانات هذا النوع على شكل شجرة، ودائما يتم البحث في الشجرة ابتداء من الجذر Root باتجاه تفرع الشجرة. لذا يعد هذا النوع هو الأفضل من ناحية البحث عن البيانات، والشكل (1) يوضح كيفية ترابط كيانات قواعد البيانات الهرمية

الشكل (11): التنظيم الهرمي لقواعد البيانات صفحة 92

2- قواعد البيانات الشبكية network databases أما قواعد البيانات الشبكية فتشبه قواعد البيانات الهرمية إلى حد كبير إلا أن كل عقدة أو فرع قد يكون لها أكثر من والد، وهذا بدوره ينعكس على طبيعة معالجة البيانات، حيث يسمح بالوصول إلى العنصر الواحد من البيانات بأكثر من طريقة كما هو موضح في الشكل (5):

صفحة 93

## 92 الوحدة الثانية

(7)

الشكل (5): التنظيم الشبكي القواعد البيانات

## - قواعد البيانات العلائقية

يمكن في هذا النوع تنظيم البيانات على شكل جداول ذات بعدين. وتعد قواعد البيانات العلائقية من أشهر أنواع قواعد البيانات في الوقت الحالي لما توفره من كفاءة عالية في المرونة، من ناحية أخرى، يمكن استخدام قواعد البيانات العلائقية على معظم أنواع الحواسيب، وخاصة الحواسيب المسفرة ، حيث إن هذا النوع من قواعد البيانات يحتاج إلى حجم تخزين قليل نسبيا في الذاكرة أو وسائط التخزين المساعدة مقارنة بالأنواع الأخرى من قواعد البيانات، بالإضافة إلى أنها لا تحتاج إلى لغة برمجة من اللغات الراقية وانما تشتمل في داخلها على لغة برمجة خاصة بها مثل لفية الاستعلامات الهيكلية

## (Structured Query Language SQL)

عزيزي الدارس، إن إنشاء الجداول يحتاج إلى جهد كبير بالإضافة إلى أن كل جدول مرتبط بالجدول الآخر من علاقة معروفة مسبقاً، ويتميز هذا النوع من التنظيم بالسهولة في النواحي الآتية:

أ- تعديل الجداول من ناحية تغيير ختان المنزل، وإضافة حقول جديدة ، وحذف حقول موجودة .

ب- اناقة العلاقات التي تربط الجداول وحذفها

ج- استرجاع البيانات المخزنة بأي صبغة يريد المستفيد

و- سهولة تعلمها واستخدامها

### 93 الوحدة الثانية

عزيزي الدارس، إن التنظيم العلائقي يلقي اهتماماً واسعاً على أعلى المستويات من قبل الشركات المنتجة الأمر الذي يجعله مرشحة ليحل محل الأنواع السابقة، والجدول (1) يوضح تنظيم البيانات في أحد الجداول في قواعد البيانات العلائقية

#### 4.4.8 أنواع قواعد البيانات ( Types of Dalilis ) s'

أولاً: قواعد البيانات الشخصية |Personal Databases

عزيزي الدارس، إن مفهوم قواعد البيانات الشخصية يطلق على قاعدة البيانات المستخدمة من جهة واحدة فقط ، والمقصود بالهة الواحدة (شخص، مؤسسة و...) .

ثانياً: قواعد البيانات الموزعة [ Distribute ] Dilal1sts عزيزي الدارس، ذكرنا سابقاً أن التقنيات والأجهزة المستخدمة في مجال الحاسوب وتكنولوجيا المعلومات تطورت تطوراً كبيراً، مما أثر بدوره في الاستخدامات المختلفة للحاسوب فقد حصلت ثورة في شبكات الحاسوب، وانتشار أجهزة الحاسوب حيث أصبح بالإمكان تخزين قواعد البيانات في أماكن عدة حسب استخدامها بدلاً من تخزينها في مكان واحد، ولولا عنصر الاتصالات الذي يؤدي دوراً مهماً في نقل البيانات وتوصيلها من مكان إلى آخر لما تمكنا من توزيع قواعد البيانات وتخزينها في أماكن مختلفة والشكل (6) يوضح أسلوب توزيع قواعد البيانات على أجهزة الحاسوب المختلفة والمتصلة بعضها

### 94 الوحدة الثانية

بعض عن طريق شبكة الاتصال، يستطيع أي مستخدم في هذا النظام الحصول على أي جزء من قواعد البيانات بغض النظر عن مكان وجود المستخدم والبيانات

الشكل (6): نظام قواعد البيانات الموزع صفحة 95

يتم الاتصال بين الحواسيب عن طريق إرسال رسائل ( Messages ) فيما بينها، وهذه الرسائل تحتوي على تعليمات أو بيانات من ناحية أخرى يمكن تعرب قواعد البيانات الموزعة على أنها مجموعة من قواعد البيانات المخزنة على أكثر من جهاز حاسوب واحد وفي أماكن مختلفة ويتم الربط فيما بينها من خلال شبكة الاتصالات

إن الوقت اللازم للحصول على البيانات هو الوقت الذي يتطلبه نقل البيانات خلال الشبكة. ويعد هذا الوقت كبيراً إذا ما قورن بوقت استرجاع البيانات من وحدات التخزين في كل حاسوب أو رقت معالجة هذه البيانات في الحواسيب المختلفة، لذا يجب تقليل هذا الوقت قدر الإمكان للحصول على كفاءة جيدة للنظام

## 95 الوحدة الثانية

عزيزي الدارس، إن البرمجيات التي تنظم عمل قواعد البيانات الموزعة ويمكن المستخدمين من تخزين البيانات واسترجاعها بصورة سهلة تسمى أنظمة إدارة قواعد البيانات الموزعة Distributed Database Management Systems، ومن أهم وظائف هذه البرمجيات ما يسمى شفافية النظام Tritisparency التي يمكن تقسيمها إلى ثلاثة أنواع هي:

1- شفافية التكرار ( Repetition ) | Ta1115pareficy .

2- شفافية مكان التخزين Localioli Traillsp411lly

3- شفافية تجزئة البيانات Fragilentition Transparency

هنالك العديد من المزايا (المحسنات التي توفرها قواعد البيانات الموزعة منها :

1- سهولة التوسع في النظام مقارنة بالنظام المركزي

يتم التوسع في النظام المركزي القديم باستبدال الأجهزة القديمة في حالة الزيادة في قواعد البيانات. أما حالة النظام الموزع فيتم التوسع وربطها بالشبكة بدلاً من استبدال جميع الأجهزة القديمة، وبذلك فإن فكرة التوسع في قاعدة البيانات الكلية تعتمد على إضافة مراكز جديدة Sites نقعد فقط

- اعتمادية النظام System Reliability

توزيع البيانات على المراكز التي تستخدمها، وتحسن سرعة الوصول إليها، وتصبح متوافرة بشكل أفضل للمستخدم. كما أنه بتكرار البيانات في مراكز مختلفة تصبح هذه البيانات أقرب إلى المستخدم وتقل كلفة الوصول إليها. في حال تعطل أي جهاز في النظام المركزي فإن النظام بأكمله يتعطل، أما إذا تعطل أحد مراكز النظام (قواعد البيانات الموزعة) فإن جزءاً من قواعد البيانات يمكن الحصول عليها من المراكز الأخرى عن طريق الاتصالات.

من ناحية أخرى هنالك بعض المساوي، لقواعد البيانات الموزعة ومنها :

1- صعوبة تحديث البيانات update Problem

## 2- صعوبة إجراء الاستفسارات المعقدة Complex Query Processing

### 3- الكلفة العالية في تصميم أنظمة قواعد الموزعة وبرمجتها .

#### 96 الوحدة الثانية

### قواعد البيانات الكينونية Object - Oriented Databases

ترجع فكرة استخدام (Object - Oriented Programming (oop إلى ما قبل 25 عاما، إلا أن استخدام Object - Oriented قد أحرز تقدما كبيرا في السنوات القليلة الماضية. وقد برزت في السنوات الأخيرة فكرة قواعد البيانات الكينونية بشكل كبير وعملي؛ لأنها تعمل على حل العديد من المشاكل التي كانت تواجهها في الأساليب القديمة ولم تتوافر لها الحلول المناسبة. وكما ذكرنا سابقا ، عزيزي الدارس، فإن قواعد البيانات تعتمد على السجلات الصغيرة نسبية، والبيانات المتجانسة والمنتظمة بالإضافة إلى الإجراءات القصيرة، وهناك الكثير من التطبيقات التي تمتاز بصفات تختلف عن هذه الصفات المذكورة أعلاه منها

\* أنظمة الرسم الهندسي (CAD) الذي يحتوي على وحدات كثيرة غير متجانسة مثل: الدائرة، والمربع، والقطع الناقص، والباب، وتلع الماكينات، وغيرها .

\* برامج معالجة النصوص Worul Processing التي تمتاز بطول السجلات

\* أنظمة الوسائل المتعددة Multimedia مثل: الصوت والصورة، والنص، والرسم، وهي تختلف في طبيعتها عن بعضها، وكذلك تختلف طرق معالجتها . الكائن Object في لغات البرمجة يعرف بأنه مجموعة من برامج الحاسوب Software التي تحتوي على مجموعة من المعلومات المتعلقة ببعضها ببعض، ويطلق على هذه الإجراءات المناهج Meliods في OOP، كما يطلق على عناصر المعلومات الخصائص Properties .

تعتبر مفهوم Object سهلا لكنه ذو قدرة ومرونة كبيرة. تعبر الخصائص عن كل شيء يعرفه الكائن (Object) في حين تعبر المناهج (Methods) عن كل شيء يمكن عمله.

أما مفهوم الكينونة في قواعد البيانات فهي وحدة تحتوي على البيانات والإجراءات التي يمكن أن تتم عليها، وهو قريب جدا من المفهوم الذي ذكر في الفقرة السابقة. إن الأسلوب القديم كان بني البيانات بشكل منفصل عن الإجراءات التي تتم عليها حيث كان يستخدم لغة وصف البيانات Data Definition Langtige (DDI) لبناء هذه البيانات ويستخدم لغة معالجة البيانات DML (Data Manipulation Language) لإجراء

#### 97 الوحدة الثانية

العمليات المختلفة عليها، أما في أسلوب الكينونات نتوصل البيانات والإجراءات التي تتم عليها في وحدة تسمى الكينونة Object كما ذكرنا سابقا إن ظهور اللغات التي تتعامل مع الكينونة مثل: ++ C , I , Delphi , Vistia , Basic , Javil وغيرها كانت العامل الرئيس لنهوض



قواعد البيانات الكينونية، ومن الجدير ذكره ، عزيزي الدارس، أن الكينونة تتركب من الأجزاء الآتية

#### 1- المتغيرات varibal

التي توضع فيها البيانات

#### 2- الرسائل Messages

التي تتبادلها مع الكيتونات الأخرى حيث يتم الاتصال بين الكيتونات بوساطة رسائل

محددة ترسلها بعضها البعض

#### 3- المناهج Methods

وهي العمليات التي يتم إجراؤها على الرسائل التي أرسلت إلى الكائن

هنالك أيضاً، عزيزي الدارس، بعض المفاهيم الأساسية التي يجب معرفتها عند التعامل مع الكينونة وهي

#### - الصنف Class

وهو قالب أو تعريف يجمع الصفات التي تخص مجموعة من الكيتونات المتشابهة مثال Student عبارة عن صنف Class لكن الطالب محمد أو الطالب عبدالله عبارة عن كينونة متشابهة في الصفات تسمى مثيل Installice

#### 2- الوراثة inheritance

وهي من أهم صفات هذا الأسلوب من قواعد البيانات، حيث إن الصنف الفرعي أو الكينونة ترث من الصنف الأعلى جميع صفاتها الخاصة بالمتغيرات والطرق، وسيتم توضيح هذا المفهوم بعمق عند دراسة UML

#### 98 الوحدة الثانية

وهناك بعض الصفات والاعتبارات الأخرى التي تنطبق على هذا الأسلوب من بناء

قواعد البيانات يمكن الرجوع إلى بعض المراجع الخاصة و « 383 | مقرر قاعدة البيانات وإدارتها، في حالة الحاجة إلى التوسع في هذا المجال.

#### أسئلة التنويم الذاتي (1)

ما الفرق بين ربعا مكونات قواعد البيانات في النموذج الهرمي والنموذج الشبكي؟

#### 9. التحليل الكينوني

## Object-Oriented Analysis Approach

### 1.9 مقدمة إلى طريقة التحليل الكينوني

الكائن Object هو أجزاء مترابطة تضم بيانات وكودا Cole يعمل في هذه البيانات. بعبارة أخرى يمكن تعريف الكائن Object بأنه مجموعة من برامج الحاسوب (Software) التي تحتوي على مجموعة من المعلومات والإجراءات المتعلقة ببعضها البعض ويطلق على تلك الإجراءات Methods في بيئة Object - Oriented ، كما يطلق على عناصر المعلومات Properties خدائع. أما المصنف Class فيمكن أن نعهه قالباً يمكن انشاء كائنات من خلاله. من ناحية أخرى بعد الأصناف هي أوصاف الكائنات دائماً، ويمكن القول إن السف بصف الهيكل الداخلي للكائن البيانات التي يحتويها، والعمليات الأنشطة التي يقدر تحقيقها على هذه البيانات، فالكائن شر حالة لصفة، وهناك العديد مین المزايا التي تتوافر عند استخدام هذه الطريقة أهمها Encapsulation هنالك العديد من لغات البرمجة التي تستخدم هذه الطريقة منها C++ , Java , Basic ويمكن توضيح كيفية استخدام هذه الطريقة وتمثيلها للبيانات في الشكل (7) المثال (12)

### 99 الوحدة الثانية

#### مثال (12)

الشكل (7): نموذج مركبة باستخدام التحليل الكينوني

### صفحة 102

عزيزي الدارس، في هذا المثال قدمت المركبة على شكل حزمة رموز النوع، وقيمتين صحيحتين: السرعة القصوى، والسرعة الدنيا. كل هذه البيانات تخص المركبة أيا كانت، بمعنى آخر، كل كائن من نوع مركبة سيكون له نسخته الخاصة من هذه البيانات، ويطلق عليها خصائص المثلث (InstaTice Attributes). إن عملية خلق المثلث Instantillion هي التي تسمح إنشاء كائن انطلاقاً من صنف تنص على منع قيم مميزة لكل خصائص المثلث

### 100 الوحدة الثانية

#### مفهوم الكائن Object

كنا قد نوهنا ، عزيزي الدارس، في بداية هذه الوحدة إلى ماهية الكائن، وفيما يأتي تذكير بالكائن وبعض المفاهيم الخاصة بعد الكائن Object وحدة شفرة ومعلومات يمكن الوصول إليها واستخدامها ببراعة كما يمكن أيضاً إعادة استخدامها، ويتكون Object من مكونين أساسيين: الأول يملك عدداً من الإجراءات أو الأعضاء التي تسمى مناهج (Methods) وتقوم بتعريف المهام التي يستطيع (Object) تأديتها، والثاني يحتوي على عدد من المتغيرات تسمى خصائص Properties تستخدم لاسترجاع أو تنظيم المواد الخاصة به Object. وسبق أن ذكرنا أن ال Object عبارة عن مثلث (Instance) لصنف Class. تصمم الأمثلة متعددة النوع من طريق

استخدام المناهج Methods والخصائص Properties نفسها التي استخدمت في تصميم  
الصف Class

أسئلة التقويم الذات (7)

ما المقصود بالكائن؟

3.9 رموز يو أم أل UML Notation

1.3.9 مقدمة إلى لغة UML Unilical Modeling Language

أصبحت UML لغة النمذجة الموحدة المعتمدة لترميز العمليات البرمجية لدى الوسط الصناعي. وبالرغم من أنها خرجت من تحت عباءة ثلاثة بعدون من أهم أصحاب المنهجيات؛ إلا أنها لقت قبولا واسعا لدى المهتمين ببناء البرمجيات على اختلاف مشاربهم ومنهجياتهم

كما ذكرنا سابقا فإن UML تستخدم رموزا ميسيلة للتعبير عن مختلف نماذج العمل البرمجي يسهل بواسطتها على ذوي العلاقة - من محللين ومصممين ومبرمجين ومستفيدين - التخاطب فيما بينهم، وتمرير المعلومات في صيغة مالية موحدة وموجزة ، تقنيهم عن الوصف اللغوي المعتاد. إنها مثل مخططات البناء التي يتبادلها الساحون والمعماريون ومهندسو التشييد، أو مخططات الدوائر الكهربائية والإلكترونية التي يمكن لأي كان في هذا المجال أن يفهمها ويتعامل معها.

101 الوحدة الثانية

هنا يجب التنويه إلى بعض النقاط التي تشكل سوء فهم ارتبط با UML لدى الكبرين

\* UML ليست منهجية لبناء البرمجيات يعني أنها لن ترشدك إلى أفضل الطرق التصميم البرمجيات وتطويرها .

\* UML لا ترتبط بمنهجية محددة لتنشئة البرمجيات، بالرغم من أنها تولدت من منهجيات سابقة، وبالرغم من أنها متوافقة ومتممة المنهجية العملية الوحدة RUP التي صاغها الأفراد أنفسهم الذين قاموا بوضع UMI. هذه اللغة يمكن توظيفها على مختلف العمليات البرمجية بغض النظر عن المنهجية المتبعة، بل بغض النظر عن وجود منهجية أصلا

أول ما يلاحظ عن UML وجود العديد من المخططات المختلفة (ماذج التي يجب التعود عليها، ويعود السبب في هذا التنوع إلى أن المنظومة يحتمل أن ينظر إليها من زوايا مختلفة حسب المشاركين فيها. يشترك عدد من الأفراد في تطوير البرمجيات، وكل واحد له دور مثلا

\*المحللون-

\*المصممون

\*المبرمجون-

\* القائمون بالاختبار.

\* مراقبو الجودة.

\* المستفيدون

\*الكاتب التقنيون

كل هؤلاء الأفراد يهتمون بجوانب مختلفة من المنظومة، وكل واحد منهم يحتاج إلى مستوى مختلف من التفاصيل. فعلى سبيل المثال يحتاج المبرمج إلى فهم التصميم الموضوع للمنظومة من أجل تحويله إلى تعليمات برمجية في مستواها الأدنى، بالمقابل ينصب اهتمام الكاتب التقني (الموثق) في سلوك المنظومة ككل، فيحتاج إلى فهم كيفية عمل المنتج تحاول UML أن تقدم لغة قوية التعبير بحيث يمكن للمشاركين الاستفادة ولو من مخطط واحد على الأقل من مخططاتها

## 102 الوحدة الثانية

الرموز الأساسية basic notations

عزيزي الدارس، لغة النمذجة الموحدة Unified Modeling Language أو UML في لغة نمذجة رسومية تقدم لنا صيغة لوصف العناصر الرئيسية للنظم البرمجية هذه العناصر نسم مشغولات artifacts [1]. وفي البنود الأتية سوف نستكشف النواحي الرئيسية في UML ، ونصف كيف يمكن تطبيق UML في مشروعات تطوير البرمجيات.

تتجه UML بطبيعتها نحو بناء البرمجيات كائنية المنحي object-oriented ، وتستخدم بعض الرموز التمثيل الكائنات ومن هذه الرموز المبينة في الشكل (8)

الشكل (8): ملخص لأدوات مخططات الصفوف في UML

## صفحة 103

بعض الملاحظات الخاصة برموز لغة النمذجة الموحدة

(UML Unified Modeling Language)

1- تمتلك UML العديد من العناصر النمذجة مختلف مظاهر النظام البرمجي بكامله وذلك بشكل رسومي

2- يمكن ملائمة مخططات الصفوف UML Class Diagrams مع مخطط كيان ارتباط مع بعض الفروقات كما سنرى في البنود التالية

## 103 الوحدة الثانية

3- مثل مجموعات الكيانات بصناديق hots , يتم إظهار الوصفات بداخلها وليس كقطاع منفصلة كما في (111) (112) TR

4- تمثل مجموعات الارتباط الثنائي في UML برسم خط يصل بين مجموعتي الكيانات ويوضع اسم الارتباط قرب الخط

5- يكتب اسم الأدوار roles الموضوع لمجموعة كيانات مشتركة في ارتباط على الحمل بقرب مجموعة الكيانات الخاصة بها

6- ممكن كتابة اسم مجموعة الارتباطات في صندوق مع واصفات مجموعة الارتباطات ويرحل الصندوق بخط منقط إلى الأمل الواصل بين مجموعتي الكيانات.

7- مثل الارتباطات قوى الثنائية معينات inions، كيا في ER diagrams

3.3.9 رموز حالات الاستخدام Use Cases

يبين الشكل (9) حالة الاستخدام السلوك النظام من وجهة نظر المستخدم

الشكل (9) حالة الاستخدام السلوك النظام من وجهة نظر المستخدم

صفحة 104

حالة الاستخدام Use

Case : هي وصف السلوك النظام من وجهة نظر المستخدم. فهي ذات پاندة في مراحل التحليل والتطوير، وتساعد على فهم المتطلبات يكون المخطط سهلا للاستيعاب، مما يمكن كلا من المطورين (محللين، مصممين، مبرمجين، مختبرين) والمستخدمين (الزبون) من الاشتغال عليه، لكن هذه السهولة يجب أن لا تجعلنا ننتقص من

104 الوحدة الثانية

شأن مخططات حالة الاستخدام، فهي بإمكانها أن تحتل كامل عمليات التطوير ، بدأ من الاستهلال حتى التسليم.

• رمز (شكل) الصنف والشيء Class and object diagram

بيان الشكل (10) مخرلات لعدد من الأصناف

الشكل (10) مخطط الاصناف

صفحة 105

رسم مخططات الأسنان جانب أساسي لأي منهج للتصميم بالمنحى للكائن، لذلك ليس بالغريب أن تقدم لنا UNIL الصيغة المناسبة له، وسوف نرى أنه يمكننا استخدام مخطط الأسنان في

مرحلة التحليل، وكذلك في مرحلة التصميم وسوف نقوم باستعمال صيغ مخططات الأصناف لرسم خريطة للمفاهيم العامة التي يمكن للمستفيد (الزبون) أن يستوعبها، وسوف تسميها النموذج المفاهيمي Conceptual Model، وهي بالإضافة إلى مخطط حالة الاستخدام تحمل من النموذج المفاهيمي أداة قريبة لتحليل المتطلبات

#### 105 الوحدة الثانية

### 5.3.0 أشكال التسلسل Sequence Diagrams

#### الشكل (11)

الشكل (11): أشكال التسلسل

#### صفحة 106

الخطوط المنقطة إلى أسفل المخطط تشير إلى الزمن، لذلك فما نشاهده هنا هو وصف لكيفية تفاعل الكائنات في نظامنا عبر الزمن بعض أدوات نمذجة UML مثل برنامج راشيونال روز Rational Rose، يمكنها توليد المخطط التتابعي آلية من مخطط التعاون، وهذا ما حدث تماما عندما رسمنا المخطط أعلاه

#### 4.9 حالة دراسية - تحليل نظام باستخدام UML - (تطبيق قواعد بيانات)

ترغب إحدى الجامعات في حوسبة العمل في مكنيتها؛ حيث يتم العمل فيها بطريقة يدوية، ويتلخص في تسجيل البيانات عن موجودات المكتبة وهي: رقم الصنف، وتاريخ الملكية، والسعر، والنوع، والحالة، وعدد النسخ، في حين أن العمليات الممكن إجراؤها على الموجودات تتمثل في إدخال الكتاب إلى المكتبة وإعطائه رقم التصنيف، وإتلاف الكتاب، واسترجاع الكتاب، وإعارته. أما موجودات المكتبة فيمكن تصنيفها إلى نوعين: الموجودات التسجيلية (العنوان، ونوعية التسجيل) ويمكن تصنيف الموجودات التسجيلية إلى برنامج

#### 106 الوحدة الثانية

حاسوب (الإصدار، والشكل) والأفلام المخرج، وتاريخ الإنتاج، والموزع) أما النوع الثاني من الموجودات فهو المنشورات العنوان، ودار النشر) ويمكن تقسيم المنشورات إلى نوعين الكتب المؤلف، والطبعة، وتاريخ النشر، ورقم الكتاب الوطني والمجلات (سنة الإصدار والجهة المصدرة). ويمكن تصنيف معلومات المستخدمين للمكتبة إلى (الاسم، والعنوان، ورقم الهاتف، ورقم التسجيل). والعمليات الممكن إجراؤها على المستخدم في استصدار بطاقة مكتبة، وسحب بطاقة المكتبة. ويمكن تصنيف مستخدمي المكتبة إلى نوعين: القراء والمستعيرين، وهؤلاء يصنفون إلى الطلاب والموظفين .

#### نشاط (1)

مستخدمة صيغة UML مثل البيانات السابقة حتى يتسنى حوسبة نظام المكتبة

مستخدمة قاعدة بيانات مناسبة.

معايير تصميم واجهة المخاطبة للمستخدم

## User-Interface Design

تعد واجهة المستخدم من العناصر الأساسية لنجاح العديد من التعليمات، حيث يجب أن تلبى حاجة المستخدم وتكون سهلة الاستعمال. وفي كثير من الأحيان يتم تجاهل المعايير الواجب توافرها في تصميم User Interface واجهة المستخدم، ويتم التركيز على المكونات الأساسية للتطبيق. إن أهمية واجهة المستخدم تتمثل في جعل التطبيق سهل الاستخدام، والذي بدوره ينعكس على كلفة التدريب، ويزيد من رغبة المستخدمين في استخدام التطبيق

### 1.10 أساسيات تصميم واجهة المخاطبة للمستخدم

عزيزي الدارس، إن تصميم واجهة المخاطبة للمستخدم ( User Interface ) نقطة جوهرية في نجاح النظام، وإن تصميم واجهة مخاطبة صعبة الاستخدام سوف يؤدي إلى رفض المستخدم للنظام، لذلك يعمل كثير من مصممي النظام على توفير العناصر الأنوية في النظام عند تصميم واجهة المستخدم :

#### 1- Windows (الإطارات):

إن عرض أكثر من شاشة (إطار) في اللحظة نفسها يمكن المستخدم من عرض معلومات مختلفة في اللحظة نفسها دون إغلاق الشاشة الحالية والرجوع لها في وقت لاحق

#### 2- Icons (الأيقونات):

استعمال أشكال مختلفة من الأيقونات والرموز حيث ترمز الأيقونات في بعض التطبيقات إلى الملفات مثل (Windose) وبعضها الآخر يمثل العمليات.

#### 3- Melills (القران)

اختيار الأوامر من القوائم المنسدلة بدلا من كتابتها على شكل أوامر

#### 4- Tinting (المؤشر) :

استخدام الفأرة mouse للاختيار من القوائم المختلفة

108 الوحدة الثانية

-5

#### Cliphics (الرسومات) :

يمكن عرض الرسومات مع النصوص لتوضيح المعنى بشكل أكبر

وهناك بعض المعايير الإضافية لواجهة المستخدم:

#### 1- الفة الاستخدام User Fahniliarity

يجب أن تكون واجهة المستخدم مستندة على الشروط والمفاهيم الموجهة للمستخدم بدلاً من مفاهيم الحاسوب، على سبيل المثال، نظام المكتب يجب أن يستعمل مفاهيم مثل الرسائل، وثائق، حافظات ..، بدلاً من أدلة، معرفين، ملف، ... أي مصطلحات تتناسب ونوعية التطبيق المستخدم ( ).

#### 2- التناسق Consisteiicy:

يجب أن يعرض النظام مستوى ملائمة من التناسق، حيث يتم عرض الأوامر والقوائم بالصيغة نفسها

#### 3- أقل مفاجأة Minimal SLIprise

إذا كان المستخدم يتوقع النتائج من أمر معين فيجب أن يبتعد النظام كل البعد عن إعطاء نتائج قد تكون مفاجئة للمستخدم

4- الاسترداد Verability Rec : يجب أن يكون النظام بقدر يمكن المستخدم من استرجاع الحالة التي كان عليها قبل حدوث خطأ معين، وهذا يمكن أن يكون من خلال رسائل تأكيدية تتمثل بالغاء العملية المالية أو السابقة

#### 5- موجه المستخدم User Guilance

يجب أن يزود النظام بأنظمة المساعدة للمستخدم أو ارتباطات مختلفة مثل الارتباطات مع الإنترنت

#### 109 الوحدة الثانية

#### 6- تنويع الاستعمال Us liversity

يعتمد بعض المستخدمين على نقل معان في غرف البيانات والمعلومات في حان يفضل البعض الآخر أنواع مختلفة مثل عرض البيانات بشكل ملولي أو عرضي لذا يجب تنويع عملية الاستخدام .

#### (2.16) أساسيات تقديم الشاشات

عزيزي الدارس، عند تصميم الشاشات هناك بعض المعايير يجب الاهتمام بها وهي



1- سهولة التعامل مع الشاشات؛ ففي شاشات الإدخال برأى سهولة إدخال البيانات وتقليل تكاليف الإدخال عن طريق لوحة المفاتيح، وذلك بتقليل عدد الرموز اللازم إدخالها ، ويكون ذلك بشكل مدروس ومقبول.

2- أن تكون الشاشة مرتبة وفق آلية معينة حتى يسهل التعامل معها، مثال شاشات الإدخال للموظف (ترتيب البيانات الشخصية، ثم البيانات الأخرى مثل المؤهلات العلمية ...

3. أن يكون ترتيب الشاشات وفق النماذج المستخدمة في النظام حتى لا يكون هناك غموض أو التباس

4- تعتمد الشاشات على مبدأ القوائم والاختبار سواء بالفأرة أو إدخال رقم الاختيار من لوحة المفاتيح

أسئلة النك و بي الداني (8)

ما المعايير الأساسية الواجب اتباعها لتصميم واجهة مخاطبة المستخدم ؟

3,10 النموذج الأولي Troloypic

يعتبر النموذج الأولي الخطوة الأولى في بناء النظام. ويعد النموذج الأولي رفقة للمواصفات والمتطلبات المقترحة في النظام ويحقق احتياجات الجهة المستفيدة إلا أنه يكون قابلاً للتطوير والتعديل طبقاً لاحتياجات ذوي العلاقة، وفي بعض الأحيان يطلق على النموذج الأولي (النموذج التجريبي).

110 الوحدة الثانية

الخلاصة

تتلخص محتويات هذه الوحدة، عزيزي الدارس، بأبلى الطرق المختلفة لكل المسائل باستخدام الحاسوب في طريقة التحليل للمسألة و طريقة تطوير الخوارزمية المناسبة، وطريقة الهندسة البرمجيات التراكيب المنطقية المستخدمة في حل المسائل في التتابع والقرار (التفرعية) والتكرارية، ففي التتابع تكون خطوات الحل تسلسلية فقط ويعد هذا الأسلوب أبسط أنواع التراكيب المنطقية، أما أسلوب التفرع (القرار) فيعني الاختيار أو الانتقال لتنفيذ جزء من الخوارزمية وربما أكثر من جزء مع إهمال أجزاء أخرى اعتماداً على شرط معين ، التراكيب التكرارية التي تعتبر من أهم وسائل كتابة الخوارزميات، ويستخدم هذا النوع من التراكيب عند الحاجة إلى تكرار مجموعة من الجمل هناك عدة طرق لتنظيم الملفات منها: الملفات التتابعية، والملفات التتابعية المفهرسة ، والملفات التحتية والملفات المباشرة قواعد البيانات هي مجموعة من البيانات المعلومات المرتبطة منطقياً موضوع معين، ترتبط البيانات مع بعضها البعض بمجموعة من العلاقات حيث إنه يمكن تصنيف العلاقات على النحو التالي: واحد إلى واحد، وأحد إلى متعدد و متعدد إلى متعدد . نظام إدارة قواعد البيانات يتكون من مجموعة من البرامج التي تقوم بإدارة ومعالجة البيانات بطريقة سهلة وسريعة الهدف تنظم نظم إدارة قواعد البيانات

إلى قواعد البيانات الهرمية، وقواعد البيانات الشبكية وقواعد البيانات العنقودية تقن قواعد البيانات إلى قواعد البيانات الشخصية، وقواعد البيانات الموزعة، وقواعد البيانات الكينونية UML لغة النمذجة الموحدة المعتمدة لترميز العمليات البرمجية لدى الوسط UML. تستخدم رموز مبسطة للتعبير عن مختلف نماذج العمل البرمجي هناك معايير يجب أخذها بعين الاعتبار عند تصميم واجهة المخاطية للمستخدم مثل ألفة الاستخدام، التناسق، أقل مفاجئة، موجه المستخدم، الأسترداد ، وتنويع الاستعمال. الصناعي

#### 111. الوحدة الثانية

طريقة الوصول إلى البيانات داخل الملف، وتشمل:

أ- الطريقة التتابعية

ب- الطريقة العشوائية.

ج- الطريقة الديناميكية

#### 14. مسرد المصطلحات

- 1- التراكيب التتابعية Sequential Structures: ترتيب جمل الحل تتابعية أو تسلسلية من أعلى إلى أسفل.
- 2- التراكيب التفرعية Branch Structures : تعتمد على شرط معين في تسلسل تنفيذ وصياغة الجمل
- 3- التراكيب التكرارية Loop Structures / Iterative: تستخدم عند الحاجة إلى تكرار أو إعادة تنفيذ جملة أو خطوة أو مجموعة جمل أو خطوات عددا من المرات.
- 4- التراكيب الشرطية Conditional Structures: تعني الانتقال لتنفيذ أحد خباري جملتين محتملتين أو مجموعتين محتملتين وهذا يعتمد على صياغة شرط يعتمد عليه تنفيذ (اختبار ) جملة أو مجموعة من الجمل.
- 5- التراكيب غير الشرطية Unconalitional Structures : الخطوات تنفذ خطوة خطوة وفقا لترتيب معين.
- 6- تركيب الانتقاء المنطقي Case Logic Structure: تنفيذ جملة أو مجموعة جمل منتقاة من جمل أو مجموعات جمل متعددة .
- 7- تركيب التابع المنطقي Sequential Logic Structure: تنفذ جمل الحل جملة بعد جملة بشكل متتابع منطقي.
- 8- تركيب التكرار المنطقي Loop Logic Structure: تنفيذ جملة أو مجموعة من الجمل مرات عديدة.

9- تركيب القرار المنطقي Decision Logic Structure : الانتقال لتنفيذ أحد خيارى جملتين محتملتين أو مجموعتين محتملتين من الحمل.

10- الثوابت Constants تستخدم لتعريف البيانات الثابتة

113 الوحدة الثانية

11- جمل الإسناد Assigniillent Sitenilents: تستخدم هذه الجمل لإسناد قيمة تعبير

إلى متغير

12- العلاقات Relationships : الربط بين كيانيين أو أكثر مختلفين في بياناتهما تفاصيلهما) لكن مشتركين في جزء محدد من هذه البيانات

13- قواعد البيانات Databases: مجموعة من البيانات المعلومات المرتبطة منطقية موضوع معينه

15- قواعد البيانات الشبكية Network Data lases تشبه قواعد البيانات الهرمية إلى حد كبير إلا أن كل عقدة أو فرع قد يكون لها أكثر من والد وهذا بدوره ينعكس على طبيعة معالجة البيانات، حيث يسمح بالوصول إلى العنصر الواحد من البيانات بأكثر من طريقة

16- قواعد البيانات العلائقية Relational Databases : تنظيم البيانات على شكل جداول ذات بعدين.

17- قواعد البيانات الهرمية Hierarchical Databases: تخزن البيانات على شكل تركيب هرمي في العقد حيث تنمو الشجرة عمودية وقد يتفرع من العقدة Nucle فروع متعددة تنتهي بعقدة وليدة، وتسمى العقدة المولدة بالوالد Parent والجديدة بالأبناء Children حيث لكل ابن والد واحد فقط، في حين أن الوالد من الممكن أن يكون له أكثر من ابن

18- قواعد البيانات الشخصية Individual Databases: يطلق على قاعدة البيانات المستخدمة من جهة واحدة فقط، والمقصود بالجهة الواحدة (شخص، مؤسسة).

19- قواعد البيانات الكينونية Object Oriented Data lscs: تعرف على أنها مجموعة من برامج الحاسوب Software التي تحتوي على مجموعة من المعلومات والجراءات المتعلقة ببعضها البعض ويطلق على هذه الإجراءات ال Methods,

20- الكيان Entity : تستند قواعد البيانات في مكوناتها على مفهوم النموذج الذي تعتبر الكيانات Intities جزءا رئيسية منه

21- المتغيرات Variables : تستخدم لتعريف البيانات المتغيرة .

22- المعالجة بالدفعات Tatch Processing: تسجيل البيانات المتعلقة بالعمليات وتجميعها بشكل دفعات (مجموعات) حيث تتم معالجتها بصورة دورية حسب طبيعة هذه العمليات (يومية أو أسبوعية).

#### 114 الوحدة الثانية

23- المعالجة بالوقت الحقيقي Time Processing - Real : تتم معالجة العمليات بصورة مباشرة وفور تسجيلها وإدخالها إلى النظام .

24- ملف الحركات Transaction File : الملف الذي يحتوي على بيانات غير دائمة (أي يحتوي على بيانات بصورة مؤقتة)

25- الملف الرئيس Master File: الملف الذي يحتوي على البيانات الأساسية عن موضوع معين (مجموعة من السجلات)

26- الملفات التتابعية Sequential Files توضع البيانات بشكل متجاور داخل الملف.

27- الملفات التتابعية الفهرسة Index Sequential Files تخزن جميع السجلات في ملف ذو فهرس (جدول)، وعادة تكون السجلات مرتبة داخل الملف المفهرم ترتيبا تصاعديا أو تنازليا حسب قيمة مفتاح السجل ويقوم نظام الملفات الفهرسة بتكوين فهرس لهذه السجلات ويتكون من قيمة مفتاح السجل وعنوان السجل المخزن على وحدة الأقراص المغناطيسية وعنوان السجل

28- الملفات المباشرة Direct Files: تعتمد في ترتيب السجلات في الملف على إيجاد علاقة

بين مفتاح السجل والموقع الموقع التخزيني للسجل

29- الملفات التحتية Ishing Files: يعتمد نظام الملفات التحتية على تخزين السجلات أماكن يتم تحديدها بناء على علاقة رياضية يتم اختيارها من قبل مصمم الملف في المبرمج) .

30- نظم إدارة قواعد البيانات ( Data I ase Maningenicent Systems : هو مجموعة من البرامج التي تقوم بإدارة ومعالجة البيانات بطريقة سهلة وسريعة

31- الواجهات Interfaces: يتم التعامل مع أنظمة إدارة قواعد البيانات من خلال الواجهات Interfaces التي توفرها هذه الأنظمة لكي يتمكن المستخدم من التعامل معها

32- واجهة المخاطبة User Interface هي تصاميم على الشاشة يتم من خلالها التخاطب بين المستخدم والنظام

#### 115 الوحدة الثانية

## الوحدة الثالثة

### 120 الوحدة الثالثة

#### 2. لغات برمجة الحاسوب

تعد لغات البرمجة حلقة الوصل بين الإنسان والحاسوب، وكلما كانت هذه اللغة بسيطة وأشبه بلغة الإنسان كان التعامل مع الحاسوب أسهل وأسرع وأفضل، إن التطور الذي حصل في مجال الحاسوب وتكنولوجيا المعلومات لم يشمل التطور على أجهزة الحاسوب فقط بل شمل أيضا التطور في لغات البرمجة المستخدمة الكتابة البرامج، فقد استخدم الإنسان في بادئ الأمر لغة الآلة، وهي لغة ذات مستوى منخفض، وكانت لغة مملّة وصعبة ثم توصل إلى لغات رمزية مثل لغة التجميع، وهي أيضا لغة ذات مستوى منخفض، ومع التطوير الدائم توصل الإنسان إلى لغات البرمجة ذات المستوى العالي (الراقية) القريبة من لغة الإنسان، وبشكل عام، عزيزي الدارم، يمكن تصنيف لغات البرمجة إلى صنفين: الأول القات المستوى المنخفض، والثاني لغات المستوى العالي، وسوف نتعرض لكل نوع بالتفصيل في البنود التالية

1.2 اللغات ذات المستوى المنخفض Level Languages - Low إن جميع البرامج مثل داخل الحاسوب بالشفيرة الثنائية (1) و (0)، وهي تشبه في ذلك الدوائر الكهربائية، والعناصر الإلكترونية التي تمثل بحالة (On) أو (Off)، لذلك في البداية وقبل اختراع لغات البرمجة كانت تستخدم الرموز الكتابة البرامج للحاسوب، وسميت لغة الآلة Machine Language، وهي أبسط لغة استعملت للبرمجة ويستخدمها الحاسوب في سائر أجهزته الداخلية، ويتعامل معها لمعالجة البيانات والمعلومات الواردة إليه من وحدات الإدخال كما يتعامل بها لإخراج النتائج التي تحول فيما بعد إلى اللغة بفهمها الإنسان عبر أدوات الإخراج المختلفة. إن لغة الآلة هي اللغة الوحيدة التي يفهمها الحاسوب، وهي لغة رقمية وأمرها وتعليماتها مكتوبة بتشكيلات الأرقام الثنائية (0 و 1)، وتعتمد هذه اللغة على تركيب الحاسوب، وتتميز بصعوبة كتابة البرامج، وصعوبة اكتشاف الأخطاء وتصحيحها، كما أنه لا يمكن عمل توثيق للبرمجيات باستخدام اللغة. وفيما يأتي مثال على البرمجة بلغة الآلة

صفحة 119

مثال (1)

101001 1001001000- نقل محتوى العنوان 1001001000 إلى المسجل AX، 101110  
10 10000 000- جمع محتوى العنوان 0001000010 إلى محتوى المسجل AX 110001  
10001100001- تخزين محتوى المسجل AX في العنوان 1001100001.

نتيجة الصعوبة التي كان يواجهها المبرمجون في كتابة البرامج بلغة الآلة وخاصة عند تتبع الأخطاء الواردة في البرامج، فقد أصبح من الضروري تطوير لغة أكثر سهولة من لغة الآلة

وبعيدة ما أمكن عن الرقمين الثنائيين القائمة عليهما (0 و 1). ومن هنا جابت خطوة البداية للغة التجميع Assembly Language. كانت الخطوة الأساسية هي التعويض عن الأرقام الثنائية البحتة باستخدام رموز ذات دلالة على معنى العملية المراد تنفيذها مثل Add , Mul , Sub ، وعلى عنوان البيان المراد تنفيذ العملية عليه، وقد تم ذلك باستخدام رموز دالة على المقصود من العملية فمثلا التعليمة 0001000010 101110 في لغة الآلة تعني جمع محتوى العنوان 0001000010 مع محتويات المسجل AX، تصبح في لغة التجميع ADD AX , B؛ فقد تم التعبير عن العملية بكلمة ADD بدلا من 101110، وعن العنوان بالحرف B بدلا من 10 000 10000.

تلاحظ، عزيزي الدارس، أن العملية أصبحت أسهل عند استخدام لغة التجميع. وتصنف لغة التجميع كما ذكرنا سابقا من اللغات المنخفضة المستوى، وبما أن اللغة الوحيدة التي يفهمها الحاسوب هي لغة الآلة المكتوبة بالرموز الثنائية، لذلك كان لا بد من تحويل البرنامج المكتوب بلغة التجميع إلى لغة الآلة ليتم تنفيذه بالحاسوب، واستخدم لهذه الغاية المجمع Assembler وكما هو الحال بالنسبة للغة الآلة، وعلى الرغم من التطور الذي تحقق باستخدام لغة التجميع، إلا أن هناك بعض الصعوبات التي برزت وهي: تعلم مكونات اللغة وفهمها وخاصة لغير المتمرسين، كذلك صعوبة تعديل البرنامج أو تصحيحه. وتستخدم لغة التجميع في الوقت الحالي عند الحاجة لبرامج تمتاز بالسرعة لأنها اللغة الأكثر كفاءة للتعامل مع عناوين الذاكرة ومسجلات الحاسوب بشكل مباشر.

صفحة 120

2.2 اللغات ذات المستوى العالي High Level Languages - تتميز لغات المستوى العالي بتعدددها ، بحيث أصبح كل منها ملائمة لتطبيق مجال أكثر من غيره، لذلك ظهرت عدة لغات أعلى مستوى من لغة التجميع وأقرب إلى لغة الإنسان، وعرفت هذه اللغات باسم اللغات الراقية High Level Languages (HLL) ، وقد صممت هذه اللغات لتستخدم على أكثر من نوع من أجهزة الحاسوب المختلفة دون الحاجة إلى تغيير أو تعديل حيث لا يعتمد كتابة البرنامج على التركيب الداخلي للحاسوب. وقد ساعد ذلك كثيرة على الإفادة من الحاسوب بقدر أكبر، حيث إن عملية البرمجة أصبحت لا تتطلب معرفة كبيرة بتركيب الحاسوب أو تعليماته بل يكفي الإلمام بلغة البرمجة نفسها. وهنالك، عزيزي الدارس، عدة أنواع من لغات البرمجة عالية المستوى فمنها ما يطلق عليها اللغات التقليدية مثل: سي C، وباسكال Pascal وفورتران Fortran ، وغيرها. ومنها ما يطلق عليها اللغات البنائية Structured مثل: لغة الاستعلامات الهيكلية Structured Query Language ( SQL). ومنها ما يطلق عليه الكينونية (Object - Oriented Language ( OOP ) ومنها الأمثلة على هذا Delphi C++ و Java وغيرها .

تسمى البرامج التي تكتب باللغات الراقية البرامج المصدرية (Source Programs) ، لذلك لا بد من تحويل البرنامج إلى لغة الآلة حتى يتم فهمه من قبل الحاسوب حيث إنها تولد عددا كبيرا من الخطوات المكتوبة بلغة الآلة، ويطلق على هذه البرامج بعد تحويلها إلى لغة الآلة برنامج الهدف. والشكل (1) يوضح كيفية تحويل البرنامج المكتوب بلغة عالية المستوى إلى لغة الآلة

الشكل (1): تحويل البرنامج المصدر المكتوب بلغة عالية المستوى إلى البرنامج الهدف

(البنية الآلة)

صفحة 121

عزيزي الدارس، إن برامج الترجمة يمكن تصنيفها إلى الأنواع الآتية

- المترجمات Compilers

وهي تقوم بتحويل البرنامج المصدري (Source Program)، وهو البرنامج المكتوب بلغة من لغات البرمجة عالية المستوى إلى برنامج هدف (Object Program)، وهو برنامج لغة الآلة، وتتم عملية التحويل هذه دفعة واحدة، ويمكن أن يحتوي المترجم العالي على برامج خاصة لرصد الأخطاء واكتشافها. وتعد لغة السي C من الأمثلة على هذا النوع من المترجمات.

2- المفسرات Interpreter

وهي مترجمات فورية تقوم بعملية الترجمة من لغة البرمجة عالية المستوى إلى لغة الآلة سطرًا بعد سطر، حيث يترجم السطر الأول وينفذ، ثم يترجم السطر الثاني وينفذ وهكذا حتى نهاية البرنامج. ومن الأمثلة على المفسرات تلك المستخدمة في لغة بيرسك.

- برنامج التجميع Assembler Program

ويقوم هذا المجمع بتحويل البرنامج المكتوب بلغة الأسبلي Assembly Language إلى برنامج مكتوب بلغة الآلة (البرنامج الهدف) ليتم تنفيذه بعد ذلك

أسئلة التنويم الذاتي (1)

- وضح الفرق بين لغات البرمجة ذات المستوى العالي ولغات البرمجة ذات المستوى المنخفض

2- وضح الفرق بين المترجمات والمفسرات.

صفحة 122

3. برمجة الحاسوب بلغة سي

عزيزي الدارس، إن تعلم البرمجة يمر بمرحلتين الأولى تعلم كيفية التفكير كمبرمج، أقصد التفكير في حل المشاكل problem solving، ثم تليها تعلم لغة برمجة معينة. وقد تعلمت في الوحدات السابقة عددا من الأساليب المختلفة في كيفية حل المسائل وتحليلها. وفي هذه الوحدة سوف نتعلم البرمجة بلغة السي C عزيزي الدارس، كانت أول بداية اللغة السي في مختبرات شركة T & AT الأمريكية عام 1972م. وكانت البداية كمشروع تولى الإشراف عليه دينيس ريتشي (Dennis Ritchie)، وهو مخترع لغة السي باستخدام نظام التشغيل Unix. ولغة سي هي في الأصل لغة قديمة يطلق عليها اسم BCPL، وقد طورها مارتين ريتشارد (Martin

(Richards) في كامبردج عام 1967م، والتي بدورها كانت لغة B التي اخترعها كين تومسون. ولقد أدت لغة B إلى تطوير لغة السي في السبعينيات من القرن الماضي، حيث أخذ دينيس ريتشي من لغة B أفضل تعليماتها، وغير في كثير من نيودها، وأضاف إليها تعليمات أخرى عديدة، وجعل منها لغة برمجة وساهها لغة C وبالرغم من أن لغة C قد أنشئت لاستخدامها في برمجة النظم إلا أن استخدامها كلفة ذات مستوى عال انتشر في تنفيذ البرامج التطبيقية، ومع ذلك فإن لغة C و ++C تستخدمان كلفة ذات مستوى متدن وذلك نظرا لقدرتها على توفير المميزات التي توفرها اللغات في هذا المستوى، كالقدررة على التعامل مع العناصر الثنائية bit مثلا. وفي الوقت نفسه تستخدم لغة C و ++C كلفة ذات مستوى عال لقدرتهما على توفير جمل التحكم (Control statements) والهياكل البنائية (Data Structures) التي هي من مميزات اللغات ذات المستوى العالي ولقد تطورت لغة السي بشكل كبير واكتسبت شعبية وحظيت بإقبال منقطع النظير، ولعل السبب الرئيس هو شعبية نظام التشغيل Unix وانتشاره، وهو من أهم نظم تشغيل الحاسبات المتطورة، حيث إن هذا النظام كتب بلغة السي، وهذا الأمر بعد مصدر قوة للغة السي بين سائر اللغات الأخرى أصبحت لغة السي الآن من أشهر اللغات وأكثرها انتشارا في الدول المتقدمة، ولقد بدأ المبرمجون من جميع الاختصاصات يكتبون برامجهم بها، وعلى سبيل المثال لا الحصر ظهر كتاب (برامج المايكرو كمبيوتر للمهندسين الكيميائيين) عام 1987م عن دار ماك -

صفحة 123

جرو هيل (Hill - McGraw) الأمريكية، مكتوبة من ألفه إلى يائه بلغة السي، ومعنى ذلك أنه يفترض أن على جميع المهندسين الكيميائيين في أمريكا أن يفهموا لغة السي، ويتعاملوا بها، أضف إلى ذلك أن معظم الشركات تطلب في إعلاناتها مبرمجين بلغة سي. كانت معيارية لغة سي لعدة سنوات عبارة عن مصنف Compiler مزود على نظام التشغيل Unix. وبعد انتشار الحاسب الشخصي وشيوعه تم إنشاء العديد من المصنفات للغة سي. ومما يجدر ذكره هنا، عزيزي الدارس، أنه على الرغم من تعدد المصنفات كانت هناك توافقية عالية جدا بين أكثرها. وبسبب عدم وجود أي معيار أو مقياس Standard معتمد بين هذه المصنفات ظهرت بعض التناقضات بينها، ولمعالجة هذه المشكلة أسست لجنة في صيف عام 1983م لإنشاء معيار وتعرف اللجنة ب ANSI، وهي اختصار American National Standards Institute والتي تعني هيئة القياسات الأمريكية. وكان الدور الأساسي لهذه اللجنة هو تعريف لغة سي تعريفا واحدا مشتركا لجميع المصنفات، وقد أخذت معالجة توحيد القياس للغة سي ما يقارب 6 سنوات، وهذا أكثر مما كان متوقعا. وفي نهاية عام 1989م ثبت أول معيار للغة سي، وسمي ANSI C، ولغة سي القياسية تسمح بأن يكون الاسم التعريفي مكونة من أي عدد من الرموز حتى 31 رمزا وكانت النسخ الأولى من هذا المقياس متوافرة في عام 1990م.

### 1.3 مقدمة إلى لغة

تعد لغة السي إحدى لغات البرمجة ذات المستوى العالي، وتعد أفضل لغة لكتابة برمجيات النظم التي تتحكم في المكونات المادية للحاسوب مباشرة، ومعظم نظم التشغيل مكتوبة بلغة السي مثل:



نظام Windows 95 / 98، ونظام التشغيل UNIX عزيزي الدارس، عند اختيار لغة البرمجة المناسبة لتطوير نظام ما، فإن أول ما يجب التفكير به هو نقاط القوة ونقاط الضعف في كل لغة من اللغات المتوفرة لكي نتأكد من اختيار الأفضل منها، إن لغة السي C هي لغة سهلة وليست معقدة أو مطولة كما أنها لغة مرنة ويمكن التوسع بها بسهولة، إنها توفر ميزات اللغات الدنيا (Low Level Languages) مثل لغة الأسمبلي حيث تكون التراكيب المستخدمة بسيطة وقريبة من تركيب الأجهزة، ولذلك فإنها أصبحت بديلاً للغة الأسمبلي كما يمكن بوساطة لغة السي بناء تراكيب بيانات قوية ومعقدة؛ فمع أن لغة سي لا تحري أصلاً أوامر للتعامل

صفحة 124

مع التراكيب المركبة، لكن مترجم السي يحتوي على العديد من الدالات (Functions) التي تمكن المبرمج من تنفيذ هذه الأوامر بسهولة. وبالمثل فإن لغة السي تفتقر إلى بعض الإمكانيات الأساسية مثل الإدخال والإخراج، لكن هذه الإمكانيات متوفرة على شكل مكتبة (Library) من الدوال التي يمكن استخدامها لهذا الغرض

### 2.3 عناصر لغة سي

تحتوي لغة سي على عدد من العناصر التي من شأنها التحكم في سير عمل البرنامج. هنالك العديد من الأوامر المستخدمة للتحكم في سير عمل البرنامج مثل: جملة إذا (if)، وجملة بينما (while)، وجملة لأجل (for)، وجملة المفتاح (case - switch). وهنالك العديد من هذه الأوامر التي سنتعرفها من خلال دراستنا، إن هذه الأوامر، عزيزي الدارس، تمكن المبرمج من بناء برنامج مصمم ومركب بشكل جيد مما يمكنه من الاستغناء عن استخدام الأمر اذهب إلى (goto) الذي يسبب الكثير من المشاكل ويفضل عدم استخدامه

#### 1.2.3 لمحة عن لغة سي وميزاتها

عزيزي الدارس، نعرض فيما يأتي بعض أهم مميزات لغة السي خاصة عند

استخدامها في التطبيقات الشاملة:

أ- لغة سريعة؛ فهي تتحكم في المكونات المادية للحاسب مباشرة، ولذلك تعد نسبياً أسرع تنفيذاً من اللغات الأخرى بل أصبحت قريبة إلى حد كبير من سرعة تنفيذ لغة الأسمبلي

ب- لغة صالحة لأغراض متعددة: تجارية، علمية، فنية، أدبية.

ج- لغة لها صورة قياسية معتمدة: لأن تطويرها تم في شركة واحدة. ونلاحظ أن الفرق بين نمط وآخر من أنماط لغة السي بسيط جداً، بينما نجد أن الفروق كبيرة وكثيرة بين أشكال وصور لغة مثل لغة البيسك، لأن كل شركة صانعة لأجهزة الحاسبات تضع تعليمات خاصة ومختلفة للغة، وتختلف عما تضعه الشركات الأخرى.

د- لغة غنية بالدوال: وتوفر للمبرمج عدداً كبيراً من دوال الإدخال والإخراج والدوال الرياضية والحسابية، وهو الأمر الذي يسهل عملية بناء البرامج بها

هـ- إمكانية التنقل Portability : هذه الميزة تعني أنه يمكن نقل أو تحويل برنامج من جهاز أو من نظام تشغيل إلى جهاز آخر أو إلى نظام تشغيل آخر. إن لغة السي هي من أفضل اللغات التي تمكن المبرمج من نقل البرامج من جهاز إلى آخر بسهولة تامة وخاصة في الأجهزة الصغيرة ،

و- تصميم البرامج بأسلوب البناء (Structured Programming) : إن لغة السي تمكن المبرمج من تصميم برنامج بأسلوب البناء بسهولة كونها تحتوي على جميع الأدوات اللازمة لذلك

ز- سرعة الترجمة (Compilation): بعد مترجم لغة السي من أسرع المترجمات.

ح- دعم البرمجة الجزئية (Modular Program) : تدعم لغة السي أسلوب تطوير وترجمة وربط البرامج الجزئية المنفصلة. إن هذه الميزة من أهم صفات لغة السي حيث يمكن للمبرمج أن يعمل على تطوير أجزاء البرنامج كل على حدة، وبعد ذلك يتم ربط هذه الأجزاء بعضها ببعض.

- توافر مكتبة (Library) كبيرة من الدوال الخاصة الجاهزة في لغة السي: هذه المكتبة تحوي دوال خاصة بالرسومات (Graphics) ، ومعالجة الملفات (Files) ، وقواعد البيانات والنوافذ والاتصالات وغيرها . إن هذه الدوال تعمل على توفير الوقت والجهد عند تطوير البرمجيات

ي- الكفاءة في استخدام الذاكرة (Memory Usage): بما أن البرامج المكتوبة بهذه اللغة صغيرة ولا تحوي دالات داخلية فإن الذاكرة اللازمة لتنفيذها أصبحت صغيرة والاستفادة منها أكثر كفاءة

وهناك، عزيزي الدارس، بعض العيوب الموجودة في لغة السي ولكنها لا تذكر فيما إذا قورنت بالميزات التي ذكرنا سابقا، ومن هذه العيوب:

أ- أنها تعتمد على أسلوب المترجم الذي يترجم البرنامج بأكمله دفعة واحدة ثم يعرض قائمة من الأخطاء بعكس لغة البيسك أو الباسكال التي تعتمد على أسلوب المفسر حيث يفسر البرنامج سطراً بعد سطر مما يسهل على المستخدم تصحيح الأخطاء بكل سهولة ويسر

ب- هناك أخطاء لا يستطيع مترجم لغة السي أن يكتشفها ، وهذا قد يسبب بعض المشاكل الخطيرة التي يمكن أن تنتقل إلى أجزاء أخرى دون اكتشافها.

ج- عدم انتشار لغة سي واستخدامها في الأجهزة الرئيسية الكبيرة (Main Frame) . ولهذا السبب يصعب توافر مختصين بهذه اللغة في المراكز التي تتعامل مع الأجهزة الكبيرة، علماً بأن هذه الحقيقة بدأت تتغير قليلاً، مما يدل على زيادة انتشار هذه اللغة.

د. افتقار لغة سي إلى التنوع الكبير في أنواع المتغيرات. فمثلا في معظم اللغات المعروفة الأعداد الصحيحة (Integers)، وأعداد الفاصلة العائمة (floats) نوعان مختلفان، حيث لا يجوز نقل بيانات من نوع إلى آخر دون المرور في عملية التحويل (Conversion)، وهذا يمنع حصول بعض الأخطاء غير المتوقعة. أما لغة سي فهي تسمح بإعطاء تسمية العدد الصحيح المقابل للبيانات من نوع متغيرات حرفية (Character)، وهذا قد يتسبب في حدوث بعض المشاكل عند فحص البرامج (علمة بأن هناك هذه ميزة وليست ضعف).

### 22.3 قائمة الكلمات المحجوزة للغة سي

تتكون لغة سي من الكلمات المحجوزة بالإضافة إلى المتغيرات التي يتم تعريفها في البرنامج. وفيما يأتي، عزيزي الدارس، قائمة بالكلمات المحجوزة للغة سي:

جدول (1): قائمة الكلمات المحجوزة للغة ،

extern auto return continue case double unsigned goto else switch  
float struct static if do char short type def break while union int long  
register sizeof default for

الجدول مهم جدا صفحة 127

### 3.2.3 الثوابت وأنواعها

ذكرنا ، عزيزي الدارس، في الوحدة الثالثة من هذا المقرر أن قيمة الثابتة لا تتغير، وهي القيمة الثابتة التي يتم تخصيصها، في حين أن المتغير يمكن أن تتغير قيمته من لحظة إلى أخرى، لكن في أي لحظة يحتوي على قيمة واحدة فقط. كما ذكرنا أن هنالك نوعان من الثوابت وهما

- الثوابت العددية

تشتمل الثوابت العددية على الأعداد الحقيقية (Real Numbers) التي تتم بدورها الأعداد الصحيحة (Integer Numbers) والأعداد العشرية

صفحة 127

(Decimal Numbers) كما تحتوي الثوابت العددية على الأعداد الأسية، حيث يمكنك التعبير عن الثابت العددي باستخدام طريقة التدوين البائي (E - Notation)، وبخاصة عند استخدام الثوابت العددية الكبيرة جدا أو الصغيرة جدا، ومن الأمثلة على ذلك:  $109.167 \times 10^9 = E097.167$   $7.167 \times 10^{-10} = E-103.7.167$  ثابت عددي من النوع الأسّي.  $7.167 \times 10^{-10} = E-103.7.167$  ثابت عددي من النوع الأسّي.

عزيزي الدارس، إن استخدام إشارة (-) تكون ضرورية لتوضيح أن العدد سالب في حين يتم إهمال إشارة (+) عندما يكون العدد موجبا حيث لا حاجة لظهور إشارة (+).

## 2- الثوابت الرمزية

تشتمل الثوابت الرمزية على الحروف سواء أكانت الأبجدية أم الحروف الخاصة مثل &, #, 2... بالإضافة إلى النصوص التي هي مزيج من الأحرف والأرقام مثل "Ahmed" و "X1" بشرط أن تكون بين علامتي تنصّب

### 3.3 المتغيرات في لغة سي

المتغيرات هي أسماء (عناوين) لمواقع في ذاكرة الحاسوب يخزن بها رموز أو أعداد . بعبارة أخرى المتغيرات عبارة عن حجز مكان في الذاكرة لكي يتم ملؤها بقيمة ما. ويوجد هناك العديد من أنواع المتغيرات منها ؛

#### 1.3.3 Integer Nuinbers متغيرات عددية صحيحة

وهذه المتغيرات يمكن أن تكون القيم المسندة إليها صحيحة فقط سواء أكانت موجبة أم سالبة، ونقصد بالأعداد الصحيحة الأعداد التي لا تحتوي على فاصلة عشرية. ويتم الإعلان عن هذه المتغيرات في جسم البرنامج، حيث يتم الإعلان عنها بذكر نوع المتغير ومن ثم اسم المتغير، والمثال الآتي يوضح ذلك: (int i;) حيث إن ال (int) هي للإخبار أن هذا التغير هو عبارة عن متغير عددي من نوع عدد صحيح ولا بد. عزيزي الدارس، من ذكر الفاصلة المنقوطة في نهاية الجملة كما ذكرنا سابقة. ويمكن أن يتم الإعلان عن أكثر من متغير في سطر واحد مثل (a; b; c) وهكذا. كما يفضل أن يتم الإعلان عن (تعريف) المتغيرات من النوع الواحد في سطر واحد. لاحظ المثال (2) يتم التعامل مع المتغيرات من النوع العددي الصحيح

أما ال (a) فهو اسم المتغير.

صفحة 128

مثال (2):

```
<include <stdio.h#
```

```
()void main
```

```
}
```

```
;int a, b
```

```
;a= 5
```

```
;b = 2
```

```
;(printf("%d\n", a
```

```
;(printf("%d\n", b
```

لاحظ المثال السابق يوجد به أمران لم يتم التطرق لهما من قبل وهما  $(a = 5)$  و  $(6 = 2)$ ، ومثل هذه الجمل تعرف بجمل الإسناد (أو التخصيص)، أي إسناد أو تخصيص القيمة في الطرف الأيمن لإشارة المساواة إلى الطرف الأيسر بمعنى تخصيص قيمة للمتغير  $a$  وا على التوالي. وسيتم مناقشة هذا النوع من الجمل في قلنا سابقا، عزيزي الدارس، أن المتغير يشغل حيزا من الذاكرة، وهذا الحيز يستخدم التخزين القيم في المتغير. إذا تكون القيمة 5 قد تم تخزينها في حيز المتغيرة في حين أن القيمة 2 قد تم تخزينها في حيز المتغير  $b$ ، من ناحية أخرى يمكن إسناد القيم عند التعريف بالصورة  $(b, 5 = \text{int } a = 2)$ . أما الشيء الثاني الذي لم يتم التطرق له مسبقا وهو  $(\%d)$ ، وهذا يستخدم عند التعامل مع المتغيرات العددية الصحيحة، وهو يعرف بوصفات المتغيرات، وسيتم شرحها بالتفصيل في البنود القادمة. أي أن هذه العلامة ترمز دانية إلى أنه يتم التعامل مع المتغيرات العددية الصحيحة سواء أكان في جمل القراءة `scanf` أم جمل الطباعة `printf` بعد تنفيذ البرنامج السابق، عزيزي الدارس، سوف يتم طباعة الآتي على الشاشة:

2

### 2.3.3 متغيرات الأعداد الحقيقية التي تحتوي على القارة العشرية Float Numbers

تكون القيم المسندة لهذا النوع من المتغيرات قيما عددية تحتوي على فاصلة (نارزة) عشرية، والأعداد الحقيقية هي الأعداد التي يكون فيها فاصلة عشرية، سواء أكان العدد موجبا أم سالبة، وجميع ما ينطبق على المتغيرات العددية الصحيحة ينطبق أيضا على هذا

النوع من المتغيرات على الصورة  $(\text{float } a)$  حيث كلمة  $(\text{float})$  تعني أنه يجب حج مساحة في الذاكرة المتغير من نوع حقيقي بفاصلة عشرية. أما  $(a)$  فهي اسم المتغير، والمثال (3) يوضح كيفية التعامل مع هذا النوع من المتغيرات.

مثال (3)

```
<include <stdio.h#
```

```
()void main
```

```
}
```

```
;float a, b
```

```
;a = 1.14
```

```
;b = 8.1
```

```
;(printf("%f\n", a
```

```
;(printf("%f\n", b
```

```
llend of program{
```

من الواضح لنا، عزيزي الدارس، أن هنالك تشابهاً كبيرة بين المثال السابق والمثال الذي تم مناقشته في المتغيرات العددية من النوع الصحيح، ولكن الفرق هنا فقط في العلامة (%f)، وهذه العلامة تعني أننا نتعامل مع المتغيرات من النوع العددي الحقيقي بالفاصلة العشرية، بالإضافة إلى أن نوع المتغيرات المستخدمة هو float بدلاً من int. إذا كتبت البرنامج السابق ونفذته فإن ناتج البرنامج سيكون كما يأتي:

1.14

8.1

أي أنه سيتم طباعة قيمة المتغير ، وهي 1.14 وقيمة a وهي 8.1

### 3.3.3 المتغيرات الحرفية Character

هذا النوع من المتغيرات مختلف عن الأنواع التي سبقته حيث يتم إسناد قيم حرفية لهذا النوع من المتغيرات، وكل متغير يتسع لتخزين حرف واحد فقط ويمكن تعريف هذه المتغيرات من خلال الشكل الآتي: (Char ch;)، حيث إن char لإخبار جهاز الحاسوب أن يحجز حيزاً لمتغير من نوع حرفي. والمثال (4) يبين كيفية التعامل مع المتغيرات الحرفية

صفحة 130

مثال (4):

```
<include<stdio.h#
```

```
()void main
```

```
}
```

```
;char ch
```

```
;"ch="a
```

```
;(printf("%c \n", ch
```

صفحة 131

إذا نفذ البرنامج السابق، عزيزي الدارس، فإنه سيتم طباعة القيمة المخزنة في المتغير ch وهي a, ومما يجدر ذكره أن جميع ما ذكر في أمثلة المتغيرات السابقة ينطبق على المثال (4) مع الاختلاف في العلامة (%c). عزيزي الدارس، إن كل خلية محجوزة في الذاكرة لها سعة (Size)، وسعتها تعتمد على نوع المتغير، ولها عنوان (Address) يحدد مكان وجود هذه الخلية المحجوزة لهذا التغير في الذاكرة. إن السعة تحدد أعلى قيمة يمكن تخزينها من هذا النوع، حيث إن بعض الأنظمة تحدد سعة العدد الصحيح بثماني ثنائيات (bits)، وبناء عليه يمكن تخزين 256 قيمة مختلفة في هذه الخلية من هذا النوع، كما أن لغة السي تستخدم البايت (byte) كوحدة تخزين (خلية) لوضع أي رمز (character) مثل: (أرب 4. %، a، A ...).  
والجدول (2) بوضع السعة اللازمة لتخزين كل نوع من البيانات بالإضافة إلى مجال القيمة الممكن تخزينها :

الجدول (2): أنواع البيانات والسعة اللازمة لتخزينها ومجالها

Range
Size
Data Type
المجال
نوع البيانات
to 255 0
bits 8
to 127 128-
bits 8
bits -32,768 to 32,767 16
bits 0 to 65,535 16
bits-32,768 to 32,767 16
bits-32,768 to 32,767 16
bits 0 to 4,294,967,295 32
bits -2,147,483,648 to 2,147,483,647 32
to 3.4 * (10 <sup>38</sup> - 38) * 3.4

bits 32

(to 1.7 (10\*\*+308 (308-\*10) \* 1.7

bits 64

(bits 3.4 \* (10\*-4932) to 3.4(10 \*\*+4932 80

unsigned

char

enum

unsigned int

short int

int

unsigned long

long

float

double

long double

الجدول صفحة 131 مهم جدا

4.3 بنية البرامج المكتوبة بلغة

يتركب برنامج سي من مجموعة من الدوال (Functions). كل منها يعمل على أداء عمل محدد، وهي تشبه الإجراء في لغة باسكال أو البرنامج المصغر Subroutine في لغة فورتران. إن الدوال التي يتكون منها برنامج سي يمكن أن تترجم كل دالة على حدة ومن ثم يجري ربط بعضها ببعض وبأجزاء مكتبة سي لتكون البرنامج النهائي. يكون الاعتماد الأساسي في لغة سي على مكتبة الدوال الجاهزة حيث تفتقر اللغة نفسها إلى الإمكانات الذاتية. حتى أن البرنامج الرئيس نفسه مبني على دالة رئيسة اسها (main )، وهو يمثل نقطة بداية تنفيذ برنامج سي، وتكتب صيغة هذه الدالة كما يأتي

) main



ويوضع بين الأقواس ( ) المعاملات التي قد يتعامل معها البرنامج أما جسم الدالة فينحصر بين الأقواس ( ). أما الدوال الأخرى فتركيبها يشبه تركيب الدالة الرئيسة ما عدا اسم الدالة ونوعها، والصيغة المستخدمة لكتابة الدوال في هذه الحالة هي

(Type name\_of\_function (list of arguments

}

\*

عزيزي الدارس، إن السطر في برنامج السي ينتهي بالفاصلة المنقوطة (.) بالإضافة إلى أن السطر الأول يحدد نوع بيانات الدالة المستخدم type، واسم الدالة (name). وبين الأقواس تضع المعاملات التي تمرر إلى الدالة يجب ملاحظة أن جسم الدالة محدد بالأقواس ( ) التي يجب أن تكون متساوية في العدد وإلا فإن البرنامج سينتج عنه أخطاء

صفحة 132

ومشاكل كثيرة غير محسوبة. لاحظ أن اسم الدالة يجب أن يتبع بالأقواس ( ) وإن لم يستخدم أي معاملات (arguments) فمثلا في البرامج السابقة استخدمت دالة الطباعة; printf ( " % \n \ c , ch ، وعليه قمنا بطباعة قيمة المتغير ch، وحددنا نوعيه من خلال الحرف ، بالإضافة إلى استخدام \n للدلالة على سطر جديد حيث تم إحاطة هذه المعاملات بالرمزين إن وجود الأقواس ضروري، فإذا لم توضع الأقواس لا يستطيع البرنامج التمييز والتعرف على الدالة. لاحظ أن هذه الدالة غير معرفة داخل لغة سي لكنها موجودة في المكتبة الجاهزة والمتوافرة من خلال لغة سي، كما سبق ذكره، عزيزي الدارس، فإن جملة السي يجب أن تنتهي برمز الفاصلة المنقوطة ; ويمكن للجملة أن تستمر في أكثر من سطر أي عدد من الفراغات فيها. ولقد جرت العادة عند كتابة جمل لغة سي أن ترتبها بطريقة تسهل علينا قراءتها ومراجعتها .

كما يمكن وضع

5.3 إدخال برنامج بلغة سي وتنفيذه

سنقوم الآن بكتابة أول برنامج في لغة السي مهمته فقط أن يظهر على الشاشة كلية Hello World ، لإدخال برنامج بلغة السي يتم تشغيل لغة السي أولاً، وذلك بعدة طرق مختلفة اعتماداً على نوع اللغة المستخدمة سواء أكانت Turbo C++ أم Turbo C ويذكر بأن هناك عدة إصدارات اللغة السي منها ما يعمل في بيئة Windows ، ومنها ما يعمل في بيئة MS D0s لتشغيل لغة السي C افتح برنامج Turbo C++ ، واقتع ملفاً جديدة، ويجب أن يكون ذا امتداد CPP ، افترض أنك على معرفة لا بأس بها باستخدام قوائم Windows الاعتيادية كما هو موضح في الشكل (2)

صفحة 133

يبدأ أي برنامج بلغة C بتضمين ما يلزمك من مكتبات، ثم بالجزء `main()`، والذي هو بمثابة القسم الذي سيكتب ضمنه كود برنامجك؛ فالمكتبة مجموعة من التوابع المكتوبة من قبل الشركة المصنعة أو من قبلك، هذه التوابع ستلزمك في برنامجك الذي ستقوم بكتابته، ولتبسيط الأمر دعنا نفترض أنك تدرس مادة الرياضيات، وفجأة وقفت أمام حل معادلة من الدرجة الثانية، واحتجت إلى معرفة القانون الذي يمكنك من إيجاد جذور المعادلة، فما عساك أن تفعل؟ سوف تذهب إلى المكتبة القريبة منك وتبدأ بالبحث ضمن قسم الرياضيات عن القانون الذي يلزمك في حل مسألتك، وهنا وجه المقارنة هو: أنك استخدمت مكتبتك في الحل (أي تضمين قسم الرياضيات من المكتبة) ثم الاستفادة من القانون الذي ساعدك على الحل (وهو التابع الذي ستكتبه لاحقا ضمن البرنامج). كل هذا الكلام نخصره بالتعليمة الآتية

```
<include <iostream.h>
```

وتعني تضمين الملف الرأسي `iostream` في الامتداد `h`، ويحاط اسم المكتبة باقوس من الشكل ، وهناك العديد من المكتبات التي يمكن تضمينها في البرنامج وذلك حسب الحاجة مثل

```
<include <stdio.h> #include <stdlib.h> #include <math.h>
```

```
<include <string.h>
```

عزيزي الدارس، هناك نوعان من الملفات في لغة C وهما:

1- ملفات امتدادها `h` وهي ملفات بمثابة المكتبات، ونيتها تعريف التوابع، ولا تحتوي على كود إما فقط تعريفات لأسماء التوابع المحددات (Parameters).

2- ملفات امتدادها `cpp`. وهي ملفات تحتوي على الكود سواء أكان برنامج كاملا أو مرتبطة بتعريف تابع

وغالبا ما تكون أسماء المكتبات معبرة عن محتواها من التوابع، فالمكتبة السابقة (`iostream`) هي اختصار كلمة `Input Output Stream` ، وتعني مجرى الإدخال والمخرجات، أي أن هذه المكتبة تحوي ضمنها كل التوابع التي تسمح لك بالتخاطب بين الحاسب والمستخدم فمثلا لوحة المفاتيح هي جهاز الإدخال، والشاشة هي جهاز الإخراج، وهكذا ... ، ويتم استخدام هذه الملفات من قبل المترجم `Compiler` (الذي يحول البرنامج المكتوب إلى لغة الآلة) في التعرف على التوابع التي سنستخدمها لاحقا ضمن الجزء `main()`، وهو الجزء الأساسي الذي سيتم ضمنه كتابة برنامجنا ،

`)main`

```
your code here //
```

```
{ ;return 0
```

تستخدم التابع (الدالة) printf( ) لإظهار عبارة على الشاشة، ويصبح البرنامج النهائي بالشكل الآتي:

```
<include <iostream.h#
```

```
<include<stdio.h#
```

```
( )void main
```

```
}
```

```
;"printf("Hello World
```

```
{
```

صفحة 135

عزيزي الدارس، يجب ملاحظة ما يأتي:

- 1- كل الحروف السابقة هي حروف صغيرة.
  - 2- كل تعليمة ضمن الجزء ( ) main يجب أن تنتهي بفاصلة منقوطة.
  - 3- عندما تريد إظهار مقطع نصي يجب أن نضع علامات التنصيص ".".
  - 4- تستخدم الدالة ( ) printf لإظهار النص المراد طباعته على الشاشة.
  - 5- يفضل أن ينتهي الجزء ( ) main بالتعليمة ; return 0 وهي بمثابة تعليمة فحص للبرنامج والتأكد من خلوه من الأخطاء وذلك في حالة عدم استخدام void التي تعني لا شيء.
- بعد كتابة الكود السابق اضغط على الزر run لعرض النتائج الشكل (3).

Hello World

الشكل (3): شاشة تنفيذ البرنامج

صفحة 136

### 6.3 العمليات الحسابية والمنطقية في

يوجد في لغة سي C نوعان من العمليات وهما: العمليات الحسابية، والعمليات المنطقية. وفي بعض الأحيان يطلق عليها الأدوات بدلا من العمليات، وفيما يأتي شرح

## الأدوات الحسابية Arithmetic Operators

تسمع لغة C باستعمال الأدوات الحسابية من جمع وطرح وضرب وقسمة، كاللغات الأخرى، إلا أن عملية الرفع إلى الأس ليس لها أدوات مباشرة مثل الأداة \* في بيسك Basic ، وإنما تتم عملية الرفع في لغة C بطريقة أخرى، كما تختلف القسمة في بيسك عنها في السي، إذ إن أي جزء كسري ينتج عن القسمة بهمل مهما كان كبيراً، كما في الغني باسكال وكويول، فمثلاً ناتج القسمة  $\frac{8}{3}$  هود، والكسر 0.666 بهمل، ويكون ناتج القسمة باستعمال الأداة / صحيح العدد. ويمكن تلخيص الأدوات الحسابية المستعملة في لغة السي كما يأتي في الجدول (3):

الجدول (3): الأدوات الحسابية المستعملة في لغة سي

للطرح أو كإشارة سالبة -

للمجموع +

\* للضرب

/ للقسمة

% باقي القسمة للنقصان

++ للزيادة

ويختلف أداء بعض الأدوات الحسابية حسب نوع المعطيات الصحيحة، أو الحقيقية أو الرمزية فعند معاملة المعطيات الحقيقية للأدوات الحسابية، يمكن القول إن العمليات الأساسية من جمع وطرح وضرب، تجري بالطريقة التي نعرفها، إلا أن هناك حدوداً يجب أن نذكرها وهو ألا تتجاوز قيمة النتيجة لأي عملية حسابية الحدود المرسومة لها (نوع المتغير الناتج)؛ ولأن لكل نوع من أنواع المتغيرات حدوداً إذا ما تم تجاوزها فإنه ينتج عنها خطأ . وعند معاملة المعطيات الصحيحة بالأدوات الحسابية تعمل الأدوات بالطريقة التي نتوقعها، وعند تجاوز الحدود المسموح بها في القيمة الصحيحة فإن هذا يعني أن خطأ قد وقع Overflow، وفي هذه الحالة تتلقى من المترجم رسالة تدل على الخطأ ، فمثلاً إذا كان لدينا البرنامج الآتي

صفحة 137

}

;int n=33000

;n=n\* 3

{

فعند طباعة النتيجة النهائية لقيمة المتغير n نتوقع أن يكون الجواب 99000، لكن الجواب في هذه الحالة لن يتجاوز 30464، وهو الحد الأعلى المسموح به للقيمة الصحيحة ، وهناك أمر آخر يتعلق بالقسمة فعندما نقسم 8 على 3 قسمة صحيحة 8/3 فإن الناتج يكون صحيحة أهمل 2 فقط، وإذا رغبت في المحافظة على الجزء الكسري وهو وأسقط، فيمكنك أن تحول القسمة إلى قسمة حقيقية 8.0 / 3.0 حينئذ سيكون الناتج 2.667، ولهذا أدخلت الأداة التي تمثل باقي القسمة % ويسمى Modulus Operator ويستعمل على النحو الآتي: 3%7، ويكون ناتج هذه العملية الجواب (1) وهو باقي القسمة الصحيحة 3/7 ، ومن الجدير بالذكر أن كلا من باسكال وكريول تستعملان مثل هذه العملية، ففي باسكال تكتب هذه العملية على النحو 3 mod 7، وكلمة MOD هي اختصار لكلمة Modulus ، أما في لغة C فتستعمل الأداة % لتقوم بهذا العمل.

### 2.6.3 الأدوات الأحادية والثنائية Unary and Binary Operators

أدوات الجمع والطرح والضرب والقسمة وباقي القسمة أدوات ثنائية binary ؛ أي أنها تأخذ (تتعامل مع) قيمتين وتنتج قيمة واحدة، فمثلا نتيجة 2\*3 هي القيمة 6، وهناك الأداة الأحادية (-) عندما تتعامل مع قيمة واحدة، فمثلا (1999-) تمثل الإشارة السالبة، وهي هنا أداة أحادية Unary، والعملية هذه ليست عملية طرح كما نعلم. 3.6.3 أدوات الزيادة والنقصان Increment and Decrement من مميزات لغة سي C أنها تستعمل الأدائن الحسابيتين ++ و -- لزيادة القيم بمقدار 1 أو إنقاصها بمقدار 1، على النحو الآتي

;++a

;a++

صفحة 138

ومعنى الجمل السابقة إضافة العدد 1 إلى التغير ، ويمكن كتابتها بصورة مكافئة على النحو الآتي: a+=4؛ وبالطريقة نفسها يمكن إنقاص 1 من قيمة a على النحو الآتي

;a--

أو

--a

ومعنى الجمل السابقة انقاص العددا من المتغير A. ويمكن كتابتها بصورة مكافئة على النحو الآتي: a=a-1 ;

عزيزي الدارس، هناك فرق في سرعة التنفيذ، فالتعبير ++a أسرع من التعبير a = a+1 ، وهذه هي الفائدة من استخدام مثل هذه الأدوات. ومما ينبغي التنبيه له أن هناك فرق بين ++a و ++a ، صحيح أن كلا من التعبيرين يجمع 1 إلى a، لكن عند استعمال ++a في تعبير من

التعابير، فإن  $a$  تزداد قبل استخراج قيمة التعبير، أي إذا كانت  $a=5$  فإن  $a$  تكون قيمتها 6 بعد تنفيذ  $++a$ ، أما في حالة  $a++$  فإنه يتم استخدام نبية  $a$  قبل الزيادة، وبعد ذلك تتم الزيادة بمقدار 1، أي أن العملية الأولى يمكن أن نطلق عليها جمع تقديم، في حين يمكن أن نطلق على العملية الثانية جمع تأخير، وينطبق هذا الكلام أيضا على  $a--$  و  $--a$ .

### 46.3 الأدوات العلائقية والمنطقية

Relational and Logical Operations يرجع اسم الأدوات العلائقية إلى العمليات المختصة بالقيم التي بينها علاقات، وهو إجراء عملية مقارنة منطقية بين قيم رمزية أو حسابية، وتكون نتيجة المقارنة منطقية وهي انعم أو لا (False / Title). وعادة ما تستخدم التعابير المنطقية في الجمل الشرطية والأمثلة الآتية توضح ماهية التعبير المنطقي  $x == y$  و جواب هذا التعبير إما نعم أو لا، والتعبير المنطقي  $x < 1000$  يكون جوابه إما نعم أو لا. ومن الجدير ذكره أنه في لغة السي تمثل قيمة التعبير المنطقي لا false بالعدد (0) في حين تمثل القيمة نعم true بأي قيمة غير الصفره والمتعارف عليه أنها العدد (1). ويبين الجدول (4) الأدوات العلائقية والمنطقية:

### 139 الوحدة الثالثة

الجدول (4): الأدوات العلائقية والمنطقية

الأدوات العلائقية

أكبر من <

أقل من >

أكبر من أو يساوي <=

أقل من أو يساوي >=

يساوي =

لايساوي !=

الأدوات المنطقية

&& And (حرف العطف و)

|| Or (حرف العطف أو)

! Not (النفى، أداة أحادية Unary

الجدول صفحة 140 مهم جدا

وفيما يأتي مجموعة من الأمثلة يمكن من خلالها توضيح أهمية عمل كل أداة من الأدوات السابقة وأسلوبها

افرض أن  $b = \text{int } a = 3$ ;

فإن التعبير  $a < 3$  نتيجته false أي 0

التعبير  $a \leq 3$  نتيجته true أي 1

التعبير  $a > B$  نتيجته false أي 0

التعبير  $b = !a$  نتيجته false أي 0

التعبير  $b == a$  نتيجته true أي 1

أسئلة التقويم الذاتي (2)

اذكر مثالا لكل نوع من أنواع البيانات.

صفحة 140

الجمل الأساسية في لغة سي

1.4 جمل الملاحظات والتعليقات

تستعمل سائر لغات البرمجة جمة للتعليقات والملاحظات، وكذلك لغة السي C مثلا الجملة الآتية

This is a comment statement}

في جملة ملاحظة في لغة Pascal، تقابلها جملة التعليق الآتية في لغة C

This is a comment statement//

كما هو مبين في الجملة السابقة فإن النص الذي يأتي بعد الشرطين المائلتين // هو جملة تعليق أو جملة ملاحظة، وتستعمل جيل التعليق في أي مكان من البرنامج لإبداء ملاحظة ما، عند سطر ما في البرنامج، ولا تعد هذه الجملة من الجمل التنفيذية، بمعنى أنها الوحدات من البرنامج لا يؤثر فيه. وعادة ما يتجاهلها المبرمجون، والمثال الآتي يوضح الأشكال المختلفة لجمل التعليق في لغة سي C

مثال (5)

This is a C comment statement//

\*An example on comment in C language \*/

void main() //start your program

}

/int x, y, z; /\*Variable decleration

{

عزيزي الدارس، يجب ملاحظة ما يأتي

1- لا يترك أي فراغ بين الشرطة / والنجمة \* من جهتي جملة التعليق

2- الشرطتان متلاصقتان، أي // ولا يوجد أي فراغ بينهما

3- يقوم مترجم C بإهمال النصوص المستعملة في جمل التعليق، أي أنها لا تنفذ ، بل هي جمل توضيحية تظهر في البرنامج فقط دون أي تأثير

4- يمكن وضع جملة الملاحظة والتعليق في أي مكان من البرنامج ما عدا وسط اسم تعريفي

Identifier أو كلمة محجوزة Reserved Word.

2.4 جمل الإسناد

تزدنا لغة السي بالعديد من جمل الإسناد (التخصيص)، والأمثلة الآتية توضح الأشكال المختلفة لهذه الجمل

صفحة 141

;C=C + 3

ويمكن إعادة كتابة الجملة السابقة على النحو التالي

;C += 3

الأداة += تضيف النيسة على يمين إشارة المساواة إلى المتغير على يسار إشارة المساواة، وتخزن الناتج في المتغير. بشكل عام يمكن كتابة جمل الإسناد على النحو الآتي

;Variable = Variable Operator Expression

حيث إن Operator يمكن أن تكون إحدى العملات الآتية: % , , + ومن الجدير ذكره أنه يمكن إعادة كتابة الجملة السابقة على النحو الآتي:

;Variable operator= expression

والجدول (5) يوضح الأشكال المختلفة لجمل الإسناد أو التخصيص



int C =3, D=5, E=4, F=6, G=12

الجدول (5): جمل الإسناد (التخصيص) في لغة سي

صفحة 142 معم جدا

Assigns Sample Assignment Operator Expression Explanation C = C +7 C  
العدد 7 D=D-4 D-=4 \* = E\*= 5 == += 7 D=D-4 D-=4  
(10) طرح العدد 4 من المتغير D وتخزين الناتج في المتغير D وهو (1) ضرب المتغير في  
العدد 5 وتخزين E = E \*5 الناتج في المتغير E وهو (20) قسمة القيمة الموجودة في المتغير F  
على العدد 3 وتخزين الناتج في المتغير F وهو (2) ايجاد باقي قسمة القيمة المخزنة في المتغير  
على العدد 9 وتخزين الناتج في المتغير G وهو (3) F/= 3 F=F/3 G%= 9 = F=F/3 F/= 3 G%= 9

هذا الجدول صفحة 142

### 3.4 جمل الإدخال والإخراج

تمثل جمل الإدخال والإخراج في لغات البرمجة أهمية كبرى حيث إن البرنامج الناجح يكون  
مبنية على التفاعل مع المستخدم، إن لغة سي C تزودنا بالعديد من جمل الإدخال والإخراج  
وبأشكال مختلفة، وأهمها جملة الطباعة printf() وجملة الإدخال scanf().

صفحة 142

- استخدام printf :

تستخدم هذه الدالة، وهي في الواقع ليست دالة وإنما هي إحدى الكلمات المحجوزة للغة السي  
حيث إن لكل لغة برمجة كلماتها المحجوزة الخاصة بها، وتستخدم هذه الكلمة الطباعة شيء ما  
على شاشة الحاسوب وهي تدعم مواصفات الأعداد التي سيتم مناقشتها لاحقاً، يتم تنفيذ الطباعة  
باستخدام دالة خاصة موجودة في المكتبة stdio.h تسمى printf وتستقبل هذه الدالة عدد من  
المعاملات الاختيارية من ضمنها السلاسل الرمزية (strings) ، حيث يتم طباعتها كما هي،  
وكذلك تستقبل أسماء متغيرات الطباعة القيم المخزونة بداخلها بأشكال مختلفة حسب الحاجة،  
وذلك باستخدام نموذج %X، حيث إن X تحدد نوع الشكل المطلوب (يطلق عليها الموصفات)  
فمثلاً :

%c يظهر القيمة على شكل رمز (Character).

%s يظهر القيمة على شكل سلسلة رمزية (String) .

%d يظهر القيمة على شكل عدد عشري (Decimal)،

%f يظهر القيمة على شكل عدد حقيقي (Float).

سبق أن استخدمنا جملة printf وتعاملنا معها في العديد من الأمثلة السابقة. والمثال الآتي يوضح الاستخدامات المختلفة لجملة الطباعة

مثال (6):

```
void main()
{
    int x, y;
    float f;
    char letter, string[20];
    x = 5;
    y = 7;
    f = 3.25;
    letter = 'A';
    string = "hello C";
    printf("The variable x = %d", x);
    printf("The sum = %d, the product = %d", x + y, * * y);
    printf("It's a float number %f", f);
    printf("This is [%c] a character", letter);
    printf("%s", string);
}
```

صفحة 143

في جملة الطباعة الأولى تطبع ما يأتي: The variable x = 5 وجملة الطباعة الثانية تطبع ما يأتي: the product 12 = The sun = 35 لاحظ أيضاً، عزيزي الدارس، أنه يمكن في الجمل الطويلة الكتابة على أكثر من سطر بشرط أن لا تكون هناك سلسلة في داخل شكل الطباعة موزعة على سطرين لاحظ أيضاً ميزة أخرى لدالة الطباعة printf وهي استخدام الشكل \n الذي يعد بداية سطر جديد حيث إن الرمز \ يعني للغة سي وجود رمز خاص يتبع هذا الرمز، كل رمز من الرموز التالية له معني معين كما هو موضح في الجدول (6):

الجدول (6): معاني الرموز الخاصة لدالة الطباعة Printf

الرمز Symbol

المعنى Meaning

\n new line سطر جديد

\b backspace فراغ

\r carriage return عودة إلى أول السطر

\" double quote علامة تحديد

' single quote علامة تحديد

\0 null علامة نهاية السلسلة

\\ back slash الشرطة العكسية

\65 A الرمز ذو قيمة 65 وهو يمثل حرف A

(ASCII character with value 65 (i.e. A

2- استخدام scanf

وهذه الكلمة تستخدم لأخذ المعلومات من المستخدم حتي يتم تنفيذ العملية المطلوبة. والمثال الآتي يبين كيفية التعامل مع هذه الدالة.

مثال (7):

```
;"printf("enter two number:=\n
```

```
;(scanf("%d", &a
```

```
;(scanf("%d", &b
```

```
;c= a + b
```

```
;(printf("%d", c
```

صفحة 144

المثال السابق يقوم بأخذ عددين صحيحين من المستخدم ثم يقوم بجمع هذين العددين وطبع الناتج على الشاشة. والجديد في هذا الكود هو السطر scanf); ( " % d " , & a حيث إننا استخدمنا كلمة (جملة) scant ثم بعد ذلك العلامة (%d) ، وذلك لأن المتغير الذي نتعامل معه هو من النوع العددي الصحيح (Integer) ، أما العلامة (&a) فلا بد من استخدامها عند استخدام الدالة scanf ، ومعناها أن المترجم عندما يجد العلامة (&) متبوعة بمتغير ما فإن هذه العلامة ترمز إلى عنوان المكان من الذاكرة. ويمكن أخذ أكثر من مدخل (متغير) من المستخدم باستخدام جملة

scanf في وقت واحد، أي أننا في مثالنا هذا قد أخذنا عددين من المستخدم (أي تم استخدام scanf مرتين مرة لقراءة a، والمرة الأخرى القراءة b) لذلك نستطيع أن نستخدم scanf بالصورة الآتية لكي يتم أخذ قيم المتغيرين وإسنادها إليها، وصورة scanf ستكون كما يأتي:

```
scanf("%d%d", &a,&b);
```

ويمكن لهذه الدالة أن تأخذ جميع أنواع المتغيرات، ولكن ينبغي الانتباه إلى نوع المتغير حتى تأخذ الهيئة التي تناسبه، وأقصد بالهيئة أي إذا كان المتغير متغيرة عددية حقيقية فإننا نستخدم (%f)، وإن كان المتغير متغيرة حرفية فإننا نستخدم العلامة (%c) كما ذكر سابقاً، وكما هو مستخدم في حال printf

صفحة 145

### جمل التحكم

ستعرض في هذا القسم مجموعة إضافية من جمل سي C التي تمكنا من نقل التنفيذ في البرنامج إلى أمكنة أو جمل تحددها نحن. إذا تأملنا البرامج التي وردت سابقاً فلاحظ أن تنفيذها كان ابتداء من الجملة الأولى والثانية الثالثة وهكذا حتى آخر جملة في البرنامج، وذلك دون قفزات أو إعادة بعض الأجزاء، مثل هذا النوع من البرامج يستخدم جمل التحكم التي سوف نتكلم عنها في البنود الآتية.

### 1.5 جملة الانتقال غير المشروط Goto

تستخدم هذه الجملة للتفرع الإجباري، أي الانتقال إلى مكان آخر في البرنامج دون أي شرط (الانتقال مباشرة)، وأحد الاستخدامات المفيدة لهذه الجملة هو الخروج الإجباري من حلقة دوران، والتي هي جزء من تكرارات أخرى. لاحظ أن استخدام هذه الجملة قد يسبب مشاكل كثيرة إذا لم تكن حذرين، لذا فمن النادر استخدامها حتى أن بعض المختصين قد يمنع استخدامها قطعية. لكي نستطيع التفرع إلى مكان ما يجب تحديد هذا المكان عن طريق وضع علامة (label) في ذلك المكان، واستخدام هذه العلامة في جملة التفرع. والصيغة المستخدمة هي:

```
goto label;
```

على أن يتم تعريف هذه العلامة في البرنامج كما يأتي:

```
:Label
```

والمثال الآتي بوضع ذلك:

مثال (8):

```
<include<stdio.h#
```

```
()void main
```

```
;int x =5, y= 10, a=15
```

```
;(printf("a=%d", a
```

```
;goto L
```

```
;(printf("x=%d", x
```

```
;(L:printf("y=%d", y
```

```
{
```

صفحة 146

المثال السابق سوف يقوم بطباعة قيمة 15= وينتقل مباشرة لطباعة قيمة 10=y متجاهلا طباعة قيمة x وذلك بسبب وجود جملة goto

## 2.5 جل الانتقال المشروط if

قد يتطلب الأمر في كثير من الأحيان تنفيذ جملة أو عدد من الجمل في البرنامج وذلك اعتمادا على شرط معين حيث إن جملة if تستخدم عندما يريد المبرمج أن يتأكد من شيء معين في برنامجه ليتم تنفيذ جملة أو مجموعة من الجمل اعتمادا على الشرط الذي تم تحديده في جملة if ، وذلك بفحص قيمة متغير ، هل هي القيمة التي يريد المبرمج أو المستخدم أم لا ، والصورة العامة لهذه الجملة هي

```
(if (expression
```

```
;Statement
```

حيث إن ( expression ) هو الشرط الذي يراد التحقق منه، أما (Statement) فهي الجملة التي سيتم تنفيذها في حال تحقق الشرط. ويحظ، عزيزي الدارس، أن علامة الفاصلة المنقوطة توضع فقط بعد (Statement) ولا يوضع بعد if أي فاصلة منقوطة. ننوه هنا إلى أن الجملة Statement ستنفذ إذا كان الشرط صحيحا، في حين سيتم تجاهلها إذا كان الشرط غير صحيح، وسيتم الانتقال لتنفيذ الجملة التي تأتي بعد Statement ولا تتبع لتركيب جملة if. ومن الجدير ذكره أن الشرط في جملة if (إذا) ممكن أن يكون مركبة أو بسيطاً، ونعني بالمركب أن يتكون من أكثر من شرط واحد. لاحظ المثال الآتي:

مثال (9):

```
<include<stdio.h#
```

```
()void main
```

}

;int a

;"printf("enter a number greater than 15 \n

لا بد من كتابة العلامة (8) حتى يتعرف البرنامج مكان عنوان التخزين //

;(scanf("%d", &a

// إذا تحقق الشرط سيتم تنفيذ الجملة الآتية //

;"a printf("The number you entered is not greater than 15 \n ) if<15)

#### 147 الوحدة الثالثة

لاحظ عزيزي الدارس، أننا نريد من المستخدم أن يدخل عددا أكبر من (15) لكن ماذا لو أدخل المستخدم عدد أقل من (15)، بالتأكيد سوف يؤثر هذا الإدخال في سير عمل البرنامج إذا كان البرنامج يعتمد على هذه القيمة، ولكن في برنامجنا هذا لا يهم؛ وذلك لعدم اتصال البرنامج بعمليات أخرى مبنية على هذه القيمة، وإنما كان الهدف هو التحقق فقط هل هذه القيمة أكبر من 15 أم لا. وكما قلنا سابقاً، عزيزي الدارس، أن الجملة التي تلي الشرط تنفذ فقط في حال تحقق الشرط، فلو أدخلنا للمثال السابق العدد (7) ولاحظ أنه أقل من العدد (15). أي أن الشرط تحقق فتنفذ الجملة التي تليه ويتم طباعة الجملة الآتية :

The number you entered is not greater than 15

#### 3.5 جملة إذا المتداخلة Nested if

يوجد شكل آخر لجملة إذا (if) يتم فيه تنفيذ إجراء معين في حالة تحقق الشرط

وتنفيذ إجراء آخر في حالة عدم تحقق هذا الشرط. هذه الجملة هي:

(if (expression

;Statement 1

else

;Statement 2

ومعنى ذلك إذا تحقق الشرط يتم تنفيذ الجملة الأولى (Statement1)، وإذا لم يتحقق يتم تنفيذ الجملة الثانية (Statement2). إن هذه الصيغة للجملة الشرطية هي الشائعة في جميع أنواع البرمجة. لاحظ هنا أن الجملة بعد if و بعد else تنتهي بالفاصلة المنقوطة (3)، والمثال الآتي يوضح عمل هذه الجملة

مثال (10)

```
()void main
{
    ;float avg
    ;("printf("Enter student average
    ;scanf("%f", &avg
    (if(avg>=50
    ;("printf("The Student is Passin
    else
    ;("printf("The Student is Fail\n
    {
```

صفحة 148

وهناك شكل آخر من أشكال if ويطلق عليها if المتداخلة، وتستخدم هذه الجملة عادة للتحقق من أكثر من شرط، وتأخذ الصيغة الآتية:

```
(if (expression1
    ;Statement 1
    else
    (if (expression 2
        ;Statement 2
        else
        (if (expression n
            ;Statement n
            else
            ;Statement m
```

وتستطيع باستخدام هذه الصيغة أن تتعامل مع أكثر من شرط واحد (عدة اختبارات). لاحظ المثال الآتي الذي يقوم بطباعة تقدير الطالب اعتمادا على معدله بناء على الجدول الآتي:

التقدير

Average

avg>=90 and avg<=100 A

avg>=80 and avg<90 B

avg>=70 and avg<80 C

avg>=60 and avg<70 D

avg>=50 and avg<60 E

Avg<50 F

مثال (11):

```
<include<stdio.h#
```

```
;float avg
```

```
()void main
```

```
;"printf("enter student average
```

```
;(scanf("%f", &avg
```

```
(if (avg>=90 && avg<=100
```

```
;"printf("A\n
```

```
else
```

```
(if (avg>=80 && avg<90
```

```
;"printf("B\n
```

```
else
```

```
(if (avg>=70 && avg<80
```



#### 149 الوحدة الثالثة

```
;"printf("C\n
else
(if (avg>=60 && avg<70
;"printf("D\n
else
(if (avg>=50 && avg<60
;"printf("E\n
else
;"printf("T
!15
!
```

مثال (12)

البرنامج الآتي يقوم بقراءة عددين، ومن ثم إجراء عملية الجمع أو الطرح اعتماداً على نوع العملية المراد تنفيذها بحيث إذا كانت قيمة  $op=1$  فنتم عملية الجمع كانت قيمة  $op=2$  فنتم عملية الطرح، عدا ذلك يتم طباعة رسالة خاطئة.

```
<include<stdio.h#
} ()void main
;(int i=0,j=0,,op=0
;" :printf("Enter two numbers
;(scanf("%d %d",&i,&j
;(printf("\nyou have entered [%d][%d]\n",ij
;"printf("\n1) Addition(+)\n\t2) Subtraction(-)\n
;" :printf("Enter the operation number
```

```

;(scanf("%d",&op
    (if (op=1
        }
        ;printf("\nyou select (1) addition\n
        ;ritj
        {
        (else if (op=2
        ;printf ("\nyou select (2) subtraction\n
        ;ri-j
        {
        else
        printf ("\nwhat do you mean %d, only enter 1 or 2. Intry
        ;(again\n",op
        ;(printf ("the result is%d\n",0
        {

```

صفحة 150

تدريب (1)

- 1- اكتب برنامج يقوم بقراءة 3 قيم ومن ثم جد مجموع ومعدل هذه الأعداد .
- 2- اكتب برنامجا لقراءة معدل عشرة طلاب وحده فيما إذا كان الطالب ناجحة أم راسبة.
- 3- اكتب برنامجا للطباعة معكوس العدد، أي إذا كان العدد يساوي 5 فإن معكوسه يساوي 5-.

5

أسئلة التقويم الذاتي (3)

- 1- أوجد الخطأ في كل من الجمل الآتية:

.1

```
;(If (age>=65
```

```
;"printf("Age is greater than or equal to 65 \n
```

```
else
```

```
;"printf("Age is less than 65 \n
```

2- بين مخرجات ما يأتي

-a

```
;int x=5; int y =8
```

```
}(if (x<10
```

```
(if(y>10
```

```
;"printf("***\n
```

```
}else
```

```
;"printf("#####\n
```

```
;"printf("@@@@@\n
```

-b

```
;int x=5; int y =18
```

```
}(if (x<10
```

```
(if(y>10
```

```
;"printf("***\n
```

```
}else
```

```
;"printf("#####A\n
```

```
;"printf("@@@@@\n
```

صفحة 151

#### 5.4 جملة التفرع المتعدد Switch

تستخدم هذه الجملة عادة عندما يكون برنامجك يقوم بأكثر من مهمة فمثلا برنامج الآلة الحاسبة يعمل جيدا إذا استخدمنا هذه الطريقة معه، والصورة العامة لهذه الجملة هي

```
(switch (expression  
;case constant 1: statement 1; break  
;case constant 2: statement 2; break  
;case constant n: statement n; break  
;default: statement  
{
```

تعد جملة switch من أسهل الجمل في لغة سي، ونعني به (Case constant) رقم الحالة أو الخيار الذي إذا أدخل فسوف يتم تنفيذ الجملة التي تليه (Statement) . ثم بعد ذلك جملة break ، وهي تستخدم للخروج من هذه الجملة. لأن الغرض المطلوب قد تم تنفيذه ولا يوجد حاجة لتفحص الحالات التالية. ثم تجد في النهاية الكلمة (default:) أنه في حال تفحص جميع الحالات المدرجة ولم يجد مطابقة فإنه يقوم بتنفيذ وهي تعني العملية التي تليه، ولكي تتضح الفكرة أكثر سوف نعيد كتابة برنامج الآلة الحاسبة لجمع وطرح وضرب وقسمة عددين :

```
<include<stdio.h#  
    })int main  
        ;int i  
        ;float x=0,y=0,10  
        ;("printf ("Enter two numbers: \n  
        ;(scanf("%d %d",&x&y  
        printf("\n\t%s\n\t%s\n\t%s\nli%s\nEnter the number of the  
        ,"operation  
        "Addition (1"  
        "Subtraction (2"  
        "**'Mult (3"
```

## 152 الوحدة الثالثة

```
;( " 4) Division
;(scanf("%d",&i
    (switch(i
        }
;case 1: r=x+y; break
;case 2:=x-y; break
;case 3:=x*y; break
        :case 4
        (if(y !=0
            }
            ;r=x/y
            {
            else
            }
;("printf("can't divide by zero \a\n
            ;r=0
            {
            ;break
            :default
; (printf("What do you mean by%d.\a\n",1
            {
            ;(printf("result %f\n",r
            {
```

## تدريب (2)

أكتب برنامج يقوم بقراءة نص من خلال لوحة المفاتيح وإيجاد عدد مرات استخدام  
Tab وعدد مرات Enter وعدد مرات الفراغات.

## أسئلة التقويم الذاتي (4)

بين مخرجات ما يأتي:

```
;n=15
```

```
(switch (n
```

```
 }
```

```
;case 1: printf ("the number is 1\n"); break
```

```
;case 2: printf(the number is 2 \n"); break
```

```
;"default :printf(" neither 1 nor 2\n
```

153 الوحدة الثالثة

Mostafa\_El\_Ghoul [١٥/١٠/٢٠٢٠ م، ٢:٤٦]: بعض الدوال الجاهزة وجميعها تحتاج  
math.h

Function

Description

Example

sqrt(x) Square root of x

sqrt (900.0) is 30.0

sqrt (9.0) is 3.0

exp(x) Exponential function et

exp(1.0) is 2.718282

exp(2.0) is 7.389056

$\log(x)$  Natural logarithm of x (base e)  $\log(2.718282)$  is 1.0

$\log(7.389056)$  is 2.0

$(\log_{10}(x))$  logarithm of x (base 10

$\log_{10}(1.0)$  is 0.0

$\log_{10}(10.0)$  is 1.0

$\log_{10}(100.0)$  is 2.0

$\text{fabs}(x)$  Absolute value of x

$\text{fabs}(5.0)$  is 5.0

$\text{fabs}(-5.0)$  is 5.0

$\text{fabs}(0.0)$  is 0.0

$\text{ceil}(x)$  Rounds x to the smallest integer  $\text{ceil}(9.2)$  is 10.0

not less than x

$\text{ceil}(-9.8)$  is - 9.0

$\text{floor}(x)$  Rounds x to the largest integer

$\text{floor}(9.2)$  is 9.0

not greater than x

$\text{floor}(-9.8)$  is -10

$\text{pow}(x,y)$  X

$\text{pow}(2,7)$  is 128.0

$\text{pow}(9, 0.5)$  is 3.0

$(\sin(x)$

Trigonometric sine of x (x is in radian)  $\sin(0.0)$  is 0.0

$\cos(x)$  Trigonometric cosine of x (x is in radian)  $\cos(0.0)$  is 1.0

$\tan(x)$  Trigonometric tangent of  $x$  ( $x$  is in radian)  $\tan(0.0)$  is 0.0

الجدول مهم صفحة 154