

نسخة تجريبية



مخطوط

C++

فصل كتاب المبرمجين

تم تحميل هذا الكتاب من موقع

[WWW.CB4A.COM](http://WWW.CB4A.COM)

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

وَقُلْ تَابِعُوا نَبِيَّكُمْ يَكْفُرْ عَلَيْكُمْ  
وَقُلْ تَابِعُوا نَبِيَّكُمْ يَكْفُرْ عَلَيْكُمْ



مُخْتَصِر C++ فِي زَادِ الْمُبْرَجِينَ

من تأليف:

نمر مذيّب النصيرات

# الفهرس

٥	مقدمة الكتاب.....
٨	نظرة مختصرة حول الكتاب:.....
١٠	المصاه عام.....
١١	المصاه خاص.....
١٢	الجزء الأول:.....
١٢	١- مبدأ عمل الكمبيوتر:.....
١٢	١- النظام العشري (DECIMAL):.....
١٣	٢- النظام الثنائي (BINARY) :.....
١٤	طريقة التحويل من النظام العشري إلى النظام الثنائي:.....
١٤	طريقة التحويل من النظام الثنائي إلى النظام العشري:.....
١٥	٣- النظام الثماني (OCTAL) :.....
١٥	- النظام الست عشري (HEX) :.....
١٦	- كيف نحول من العشري إلى الست عشري:.....
١٦	النظام الثنائي العشري المشفر:.....
١٧	العمليات المنطقية في سي بلس بلس:.....
١٧	١- عملية الجمع AND (&&) :.....
١٧	:الأختيار OR ٢- العملية المنطقية (  ).....
١٨	٣- عملية النفي (!)Not.....
١٨	الأدوات الحاسبية:.....
١٩	أدوات المقارنه :.....
١٩	رموز لغة C++.....
٢٠	أنواع الكلمات في C++.....
٢٠	١- المتغيرات:.....
٢١	٢- الكلمات المحجوزه:.....
٢٢	تنصيب فيجوال سي بلس بلس :.....
٢٢	العمل في بيئة فيجوال سي بلس بلس :.....
٢٣	أنشاء أول تطبيق لنا في فيجول سي بلس بلس :.....
٢٧	كتابة برنامجك الأول في C++.....
٣٩	تطبيق العمليات الحسابية في C++ تسلسل.....
٤٢	شرح خطوات تنفيذ برنامجنا السابق بالنسبة إلى الذاكرة :.....
٤٤	الجزء الثاني :.....
٤٤	بني التحكم.....
٤٤	جمل إتخاذ القرارات في C++ :.....
٤٥	أ- الجملة if (إذا):.....
٤٥	١- أحادي الاتجاه.....
٥٠	٢- ثنائي الاتجاه:.....

٥٤	متعدد الاتجاهات:
٧٤	ب - الجملة <b>switch</b> :
٧٦	شرح كيفية استخدام الجملة <b>switch</b> :
٩٤	جمل تنفيذ الحلقات في C++:
٩٤	أ- الجملة <b>for</b> :
١٠٢	عمليات الإسناد في C++:
١٠٤	عمليات الزيادة و الإنقاص في C++ :
١١٩	الجملة <b>while</b> :
١٢١	مبدأ عمل الجملة <b>while</b> :
١٣٠	كيفية التعرف على قيم المتغيرات من خلال <b>Visual C++</b> :
١٤٥	ت- الجملة <b>do- while</b> :
١٤٧	مبدأ عمل <b>do/while</b> :
١٦٢	التعليمة <b>break</b> :
١٦٥	التعليمة <b>continue</b> :

## ملاحظة :

لقد بدأت في كتابة هذا الكتاب منذ حوالي عشرة أشهر بعون الله عز و جل و تأليف هذا الكتاب لم يكن بالأمر الهين بالنسبة لي فكتابة فكرة واحد أو مثال واحد أحياناً قد يستغرق مني من الوقت ثلاث أيام، و في بداية الأمر عندما بدأت في هذا العمل قررت على عدم نشر هذا الكتاب على شبكة الإنترنت قبل الإنتهاء كلياً من كتابة كل أجزاء الكتاب المقررة.

ولكني بعد أن انتهيت من الجزء الثاني قمت في مرجعة الجزئين الأول و الثاني فوجدت بعض الأخطاء، فالكمال لله عز و جل، لذلك تراجعت في قرارتي و قررت على نشر الجزء الأول و الثاني على شكل نسخة تجريبية، لكي يتمكن جميع الأخوة من قرأت ما تم كتابته و من ثم يتكرموا عليّ بالنقد و بالملاحظات حول الأخطاء الموجودة و حول كيفية الأسلوب في الكتابة من النواحي السلبية و الإيجابية حتى أكتشف أخطائي و أحاول على عدم تكررها في الأجزاء اللاحقة إن شاء الله.

**طلب خاص:** لا تبخلوا عليّ بنصائحكم، و أرجوا أن ترسلوا لي بملاحظتكم على البريد الإلكتروني التالي:

**[namer315@yahoo.com](mailto:namer315@yahoo.com)**

و سأكون شاكراً لكم. جزاكم الله خيراً

# مقدمة الكتاب

الحمد لله حمده و نستعينه و نستغفره و نستهديه و نعوذ بالله من شرور أنفسنا و سيئات أعمالنا من يهده الله فلا مضل له و من يضلل فلا هادي له. الحمد لله هو أهل الحمد و الشناء فله الحمد في الأولى و الآخرة و له الحمد على كل حال و في كل آن، نصلي و نسلم على خاتم الرسل و الأنبياء نبينا محمد بن عبد الله و على آله و صحابته و من اتبع هداه و اقتفى أثره إلى يوم الدين و عنا معه بعفوك و رحمتك يا أرحم الراحمين.

أما بعد، أيها الأخوة الكرام و الأحبة سلام الله عليكم و رحمته و بركاته، إنطلاقاً من قول الله عز و جل:

( قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَ إِنَّمَا يَتَذَكَّرُ أُولُوا الْأَلْبَابِ )

سورة الزمر (٩)

و من قوله عز و جل :

( وَقُلْ رَبِّ زِدْنِي عِلْمًا )

سورة طه (١١٤)

ويقول المولى عز و جل مرغب في طلب العلم :

( يَا مَعْشَرَ الْجِنِّ وَالْإِنْسِ إِنَّ اسْتِطْعَمْتُمْ أَنْ تَتَفَدُّوا مِنْ أَقْطَارِ السَّمَوَاتِ  
وَالْأَرْضِ فَانفَدُوا لَا تَتَفَدُّونَ إِلَّا بِسُلْطَانِ )

سورة الرحمن (٣٣)

و قال رسول الله صلى الله عليه و سلم :

( إذا مات الإنسان انقطع عنه عمله إلا من ثلاث  
إلا من صدقة جارية أو علم ينتفع به أو ولد صالح  
يدعو له )

أخرجه مسلم و النسائي و أحمد

انطلاقاً من ذلك كله و طمعاً في إيصال الكتاب الناطق بلغة القرآن إلى جميع  
الإخوة طلبة العلم و الراغبين في تعلم لغة C++ ، و لكثرة المواضيع  
و المقالات التي قرأتها على ساحات شبكة الإنترنت من الإخوة الراغبين في  
تعلم لغة سي بلس بلس و التي تطالب في إرشادهم إلى بعض الكتب العربية  
التي تتحدث عن هذه اللغة مع كثرة الشكاوى حول قلة هذه الكتب أو  
ارتفاع ثمن الكتب التي تتكلم عن هذه اللغة و عدم قدرة جميع الإخوة  
الراغبين في التعلم في الحصول على هذه الكتب سواء بسبب السعر المرتفع  
أو عدم توفرها في جميع الدول العربية أو الأجنبية بالنسبة إلى الطلاب  
العرب الذين يدرسون في الخارج ، قررت كتابة هذا الكتاب و الذي أسأل الله  
سبحانه و تعالى بأن يُعني على كتابته بالطريقة و الأسلوب المناسبين و  
الشرح المبسط و الوافي مع إيراد الأمثلة المناسبة للمواضيع الواردة في هذا  
الكتاب .

و الله من وراء القصد

## نظرة مختصرة حول الكتاب:

في البداية أستبحكم عنراً على ضعف اللغة العربية لدي و أحياناً في عدم تناسق الجمل من الناحية اللغوية و ذلك لسببين، الأول هو أنني تلقيت هذا العلم من اللغة الروسية أي ليس من لغتي مما سبب لي ضعف في معرفة المصطلحات باللغة العربية و السبب الثاني هو بسبب المجتمع و الوسط الذي أعيش فيه و الذي هو بعيد كل البعد عن اللغة العربية.

و لكنني حاولت قدر المستطاع بالكتابة في اللغة العربية البسيطة و المفهومة لدى الجميع وكذلك الأمر بالنسبة للمصطلحات العربية المستخدمة في لغة C++ .

قمت في تأليف هذا الكتاب من خلال وجهة نظري الخاصة لكيفية التعلم الصحيح للغة C++ ، فمن وجهة نظري الخاصة حاولت التصدي للاعتقاد السائد الذي أسمعته على شبكة الانترنت و الذي يكمن في أنه من الصعب جداً إن لم يكن من المستحيل تعلم لغة C++ إن لم تكن تعرف من قبل لغة C مع العلم أنا أعرف الكثير من المبرمجين الذين تعلموا هذه اللغة من دون معرفة مسبقة للغة C .

و من وجهة نظري الخاصة أيضاً أنه إذا أردت أن تتعلم لغة C++ بالشكل الذي تصبح فيه كمبرمج لهذه اللغة و ليس كهواي برمجة، عليك أولاً بدراسة الأوامر و التعليمات الأساسية لهذه اللغة بشكل عميق و قوي و محاولة أن تتعلم كل تعليمة من تعليمات هذه اللغة على حدى و دراستها جيداً و معرفة كيفية استخدامها و موضعها في المكان المناسب بالبرنامج و معرفة كل ملاحظة أو تنبيه أو تعليق بالنسبة لهذه التعليمة ، و لكن هذا في البداية فقط، فبعد أن تتعرف على التعليمات و الجمل الأساسية لهذه اللغة بشكل جيد فإن ذلك سوف يسهل عليك المشوار قُدماً و يوفر عليك الكثير من الوقت و العناء لكي تصبح مبرمج لهذه اللغة.



و خذ مثال على ذلك فإذا أرت أن تبني منزل جديد فإن أول و أكثر ما تهتم به هو الأساس القوي و المتين لهذا المنزل و إذا تمكنت من ذلك فإنك بعد ذلك ستبدأ في اختيار الجدران و الأبواب المناسبة لك و تزين بيتك كما تشاء و من دون أي قلق أو خوف من سقوط للمنزل أو ما شابه ذلك من مشاكل البناء التي تأتي من أثار الأساس الضعيف الغير متين.

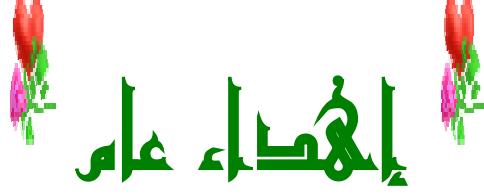
لذلك فقد شرعت أولاً في شرح المبادئ و المفاهيم الأساسية لهذه اللغة و بشكل عميق و شرح مفصل لكل جملة أو تعليمة أو أمر من أوامر هذه اللغة الأساسية، إلى الحد الذي قد أكون فيه أسهبت و أكثرت في الشرح، و لكن هدفي في هذين الفصلين الأول و الثاني هو\_ كما قلت سابقاً\_ تكوين بنية و قاعدة قوية و متينة للمبرمج حول لغة C++ حتى يسهل عليه الطريق فيما بعد بإذن الله عز وجل.

ففي الجزء الأول من هذا الكتاب بدأت في ما قبل لغة سي بلس بلس، بدأت في إعطاء فكرة مبسطة و واضحة عن النظم العددية و التي يقوم عليها أساس و مبدأ عمل الكمبيوتر، و الهدف من وراء ذلك هو تكوين صورة مبسطة عن كيفية عمل الحاسب الذي بين يديك.

ثم بعد ذلك شرعت في شرح العمليات و الرموز و الأدوات الأساسية المستخدمة في لغة C++ ،

و بعد ذلك قمت في كتابة البرنامج الأول من خلال المحرر Visual C++ مع الشرح المفصل لكيفية كتابة و تنفيذ الكود بالنسبة للكمبيوتر و الذاكرة. أما في الجزء الثاني قمت في شرح لكل من جمل اتخاذ القرار و جمل تنفيذ الحلقات و طبعاً قمت في شرح كل واحدة على حدى مع إعطاء الأمثلة المناسبة.

**نسال التوفيق**



أهدي هذا الكتاب إلى أرض فلسطين الحبيبة... إلى القدس الشريف... إلى مسرى نبينا محمد صلى الله عليه و سلم... إلى كل فلسطيني يدافع و يجاهد عن مسرى حبيبنا و نبي الهدى خاتم المرسلين عليه أفضل الصلاة و السلام.

و أهدي هذا الكتاب إلى بلاد الرافدين... إلى أرض العراق الحبيبة... أرض الخلافة الإسلامية... إلى جزء آخر فقدناه من العالم الإسلامي، لا حول و لا قوة إلا بالله العلي العظيم.  
و إلى المجاهدين الأبطال الغيورين على دينهم و أوطانهم نصرهم الله.

و أهدي هذا الكتاب إلى كل أخ مسلم من مشرق الأرض إلى مغربها.



إلى والدي الغالي على قلبي حفظه الله، الذي كان دائماً يقدم لي كل ما بوسعه من أجل أن أبلغ الدرجات العلى في العلم، فجزاك الله كل الخير يا أبي.

و إلى أُمي الحنونة حماها الله من كل شر، التي ما بخلت عليّ يوماً قط من دُعائها فجزاها الله عني كل خير.

# مختصر C++ في زاد المبرمجين

## Language (C++)

### الجزء الأول:

قبل إن نبدأ في شرح أوامر لغة سي بلس بلس يجب علينا أن ندرس بعض المفاهيم و المبادئ الأساسية التي يعمل على أساسها الكمبيوتر و لغات البرمجة التي تتعامل مع الأله مثل لغة الأله و هذه اللغة C++

١- مبدأ عمل الكمبيوتر:

أن كل ما تراه أمام عينك على الشاشة من كلمات أو صور أو صوت في الحقيقة هو ليس كذلك أي أقصد بالنسبة إلى الكمبيوتر فهو لا يعرف مثلا جملة (السلام عليكم) مثل أنت ما تقرأها بل هذه الجملة تخزن في أذاكره على شكل أرقام أذاكره هي عبارة عن مجموعة من البايتات و البايت الواحد يتكون من ثمانية بيت مثال: إذا كان لدينا ذاكره سعتها ١٢٨ بايت فهذا يعني انه لدينا  $128 * 8 = 1024$  بيت و البايت الواحد يكون على الشكل التالي:

بايت 1

--	--	--	--	--	--	--	--

و في كل بيت تخزن قيمه و أنا كل قصدي من هذه المقدمه ماذا تخزن في البيت الواحد يخزن في البيت الواحد قيمه عدديه ٠ أو ١ فإذا كانت القيمة ١ فهذا يعني عمل و إذا كانت ألقيمه ٠ فهذا يعني لا عمل و سوف أوضح هذا في التفصيل أن شاء الله و لكن في البداية يجب أن نتعرف على النظم الرقمية التي يتعامل بها الكمبيوتر

١- النظام العشري (DECIMAL): وهو أقدم نظام و يعرف أيضا في النظام الرقمي العربي و ذلك نسبة إلى مخترعه العرب و أدخلوه إلى أوروبا قبل ٨٠٠ سنة و هو أكثر نظام نستخدمه في حياتنا اليومية و هو أيضا أقرب إلى تصورنا الذهني من غيره و ذلك بسبب الاستخدام اليومي له و النظام العشري سمي بذلك لانه يستخدم عشرة أرقام و هي {٠،١،٢،٣،٤،٥،٦،٧،٨،٩}

**ملاحظته:**

( المشكلة قديما كانت أن هذا النظام غير كامل حيث كان ينتقص إلى العدد صفر حتى أوجدوه أجدادنا العرب )

و التعداد يبدأ من الصفر ٠ ثم ١ ثم ٢ .....٩ و العدد ٩ هو الأكبر في النظام العشري ثم أي عد إضافي يجب الانتقال إلى اليمين درجه و احده أي ينتج لدينا الرقم ١٠ و نلاحظ هنا أن العدد ١ يحتل خانة العشرات و العدد صفر يحتل خانة الأحاد و هكذا نتابع العد حتى نصل إلى العدد ٩٩ ثم يحدث لنا انتقال إلى اليمين درجه أخرى فينتج لدينا الرقم ١٠٠ و نلاحظ أن ١ يحتل خانة المئات .

$$\begin{aligned} \text{مثال العدد } 254 \text{ يعني } &= 1 * 4 + 10 * 5 + 100 * 2 \\ \text{و يمكن أن نكتبه في طريقة أخرى } &= 2 * 100 + 5 * 10 + 4 * 1 \\ &= 200 + 50 + 4 = 254 \end{aligned}$$

الرمز ٨ يعني بالقوى أو أس  
و نلاحظ أن النظام العشري له قاعدة ١٠ أعداد و بشكل عام أن كل الانظمة الرقمية التي ندرسها لها نفس القاعدة بالنسبة إلى التسمية المنسوب له فالنظام الثنائي له قاعدة ٢..... الخ

**٢- النظام الثنائي (BINARY) :**

و هو النظام الأكثر تعاملًا مع الكمبيوتر و بالذات مع الذاكرة . و هو يستخدم عددين فقط هما { ٠ ، ١ } وكل من هذين الرقمين يدعى {بت} و هذا البت تتغير قيمته حسب موقعه مثلا:

١ بايت = ٨ بت هذا يعني

و البت يبدأ ترتيبه من الأصغر إلى الأكبر أي من الأخف إلى الأثقل و هذا الترتيب سيكون مهم في عمليات التحويل من نظام إلى آخر

بايت 1

2 <sup>7</sup> =128 الأكبر	2 <sup>6</sup> =64	2 <sup>5</sup> =32	2 <sup>4</sup> =16	2 <sup>3</sup> =8	2 <sup>2</sup> =4	2 <sup>1</sup> =2	2 <sup>0</sup> =1 الأصغر
-------------------------------	--------------------	--------------------	--------------------	-------------------	-------------------	-------------------	-----------------------------

و المجموع = ٢٥٥ و أرجوا أن تتذكروا هذا الرقم ففي الدروس القادمة أن شاء الله سيكون مهم

تتذكروا معي أن في النظام العشري كان لدينا الخانات التالية {أحاد . عشرات . مئات.....}

ففي النظام الثنائي يصبح لدينا { أحاد . اثنان . أربعيات . ثمانيات ..... }

و في النظام العشري إذا وجد الرقم ٥ في خانة ثالثة فهذا يعني ٥ \* ١٠٠  
ففي النظام الثنائي يصبح لدينا إذا وجد الرقم ٥ في خانة ثالثة فهذا يعني ٥ \* ٤

## طريقة التحويل من النظام العشري إلى النظام الثنائي:

إذا كان لدينا العدد العشري ٥٧٣ كيف نستطيع أن نكتبه في النظام الثنائي				
نقسم العدد ٥٧٣ على ٢ فينتج لدينا الرقم ٢٨٦ والباقي ١	١	٢	٢٨٦	٥٧٣
نقسم العدد ٢٨٦ على ٢ يعطي ١٤٣ و الباقي ٠	٠	٢	١٤٣	٢٨٦
نقسم العدد ١٤٣ على ٢ يعطي ٧١ و الباقي ١	١	٢	٧١	١٤٣
نقسم العدد ٧١ على ٢ يعطي ٣٥ و الباقي ١	١	٢	٣٥	٧١
نقسم العدد ٣٥ على ٢ يعطي ١٧ و الباقي ١	١	٢	١٧	٣٥
نقسم العدد ١٧ على ٢ يعطي ٨ و الباقي ١	١	٢	٨	١٧
نقسم العدد ٨ على ٢ يعطي ٤ و الباقي ٠	٠	٢	٤	٨
نقسم العدد ٤ على ٢ يعطي ٢ و الباقي ٠	٠	٢	٢	٤
نقسم العدد ٢ على ٢ يعطي ١ و الباقي ٠	٠	٢	١	٢
نقسم العدد ١ على ٢ يعطي ٠ و الباقي ١	١	٢	٠	١

و الذي يهمنا من تلك العملية السابقة هو الباقي من عملية القسمة ثم نبدأ في عملية الترتيب من الاصغر إلى الاكبر و ذلك بالنسبة إلى البت كما تذكرون أي ينتج لدينا الرقم التالي: ١٠١١١١٠٠٠١

ملاحظه لا يجوز قرائت العدد في النظام الثنائي كما في النظام العشري بل يقرأ هكذا واحد صفر واحد واحد واحد واحد واحد صفر صفر واحد أرجو أن تتذكروا هذا و أن تميزوا في قرائت العدد الثنائي عن العشري

## طريقة التحويل من النظام الثنائي إلى النظام العشري:

ناخذ العدد الثنائي في المثال السابق ١٠١١١١٠٠٠١

في عملية التحويل نبدأ من الاصغر إلى الاكبر فنأخذ الرقم الاول و نضربه في القاعده التي ينتمي اليها النظام أي في العدد ٢ مرفوعا إلى القوه (الاس) بالنسبه إلى موقعه في أي خانه في البت ثم نتابع عملية التحويل في نفس الطريقه و عندما نتهي ما يبقى علينا سوى جمع الاعداد التي نتجت لدينا ونكون قد أنتهينا من عملية التحويل

هل فهمت ؟ لا تقلق سوف أشرح لك كيف يتم ذلك نأخذ الرقم الاصغر و هو الواحد ونضربه في قاعدة النظام أي نضربه في العدد ٢ مرفوعا إلى القوه حسب موقعه في البت و العدد ١ يقع في خانه الاولى و لا تتسوا خانه الاولى في البت هي قيمه صفر أي مرفوعا إلى العدد ٠

فتنتج لدينا المعادله التاليه

$$1.2^0 + 0.2^1 + 0.2^2 + 0.2^3 + 1.2^4 + 1.2^5 + 1.2^6 + 1.2^7 + 0.2^8 + 1.2^9 = 1 + 0 + 0 + 0 + 16 + 32 + 64 + 128 + 0 + 512 = 573$$

اليس هذا هو رقمنا 573

### ٣- النظام الثماني (OCTAL) :

و سيمي في بالنظام الثماني لانه يتألف من ثمانية أعداد و هي {٠،١،٢،٣،٤،٥،٦،٧}

بايت 1

8 <sup>7</sup> =.. الأكبر	8 <sup>6</sup> =..	8 <sup>5</sup> =..	8 <sup>4</sup> =..	8 <sup>3</sup> =512	8 <sup>2</sup> =64	8 <sup>1</sup> =8	8 <sup>0</sup> =1 الأصغر
------------------------------	--------------------	--------------------	--------------------	---------------------	--------------------	-------------------	-----------------------------

و كما نلاحظ فإن الخانات في النظام الثماني كالتالي

أحاد. ثمانيات. أربع و ستينات.....الخ

أي أن التحويل من النظام الثماني إلى العشري هو سهل جدا

مثال : لو كان معنا هذا العدد في النظام الثماني ٣١٥ و أردنا تحويله إلى النظام العشري

فكل ما علينا هو أن نأخذ الرقم الأول و نضربه في القاعده ٨ مرفوعا إلى القوه ٠ (أي

حسب موقعه في أي خانه من البت) و هكذا بالنسبه إلى العدد الثاني و الثالث ثم بعد ذلك

نجمع الأعداد التي لدينا و يكون العدد الناتج هو العدد نفسه و لكن في النظام العشري

$$5*8^0+1*8^1+3*8^2=$$

$$5+8+192=$$

$$=205$$

اما التحويل من العشري إلى الثماني فهو نفس طريقة التحويل من العشري إلى الثنائي و لكن

يختلف عنه أننا لا نقسم العدد العشري على ٢ بل على ٨ لأننا نريد التحويل إلى النظام

الثماني

و للتوضيح نأخذ العدد العشري في المثال السابق ٢٠٥

٢٠٥ نقسمه على ٨ يعطي ٢٥ و الباقي ٥

٢٥ نقسمه على ٨ يعطي ٣ و الباقي ١

٣ نقسمه على ٨ يعطي ٠ و الباقي ٣

و نبدأ في ترتيب الأرقام باقي القسمه من الأسفل إلى الأعلى ٣١٥ عشري

أليس هذا هو رقمنا

### - النظام الست عشري (HEX) :

و سمي بذلك لأنه يتألف من ١٦ عدد و هي

{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

قد يتساءل البعض و يقول انت قلت بأن النظام الست عشري سمي بذلك لأنه يتألف من ١٦

عدد و لكننا لا نرى سوى عشرة أعداد و الباقي ما هي إلا أحرف . و أنا أقول نعم هذا

صحيح لمن ينظر للوهلة الأولى و لكن لمن يتعمق قليلا فسيستنتج أن الأحرف بعد العدد ٩

ما هي إلا أعداد

$$\{A=10 , B=11 , C=12 , D=13 , E=14 , F=15\}$$

فالجمع في النظام الست عشري يكون على الشكل التالي

$$A=1+9$$

$$D=8+8$$

$$F=6+9$$

و هذا النظام هو الأكثر استخداما في معظم الأجهزة الالكترونيه و بالذات في الحاسب و في ترجمة الكود من لغة سي بلس بلس إلى لغة الآله و سوف أشرح ذلك في التفصيل في الدروس القادمة عندما نبدأ في كتابة البرامج على برنامج VC++  
 أن شاء الله عز و جل  
 و هذا النظام موقعه في البت

بايت 1

8 <sup>7</sup> =.. الأكبر	8 <sup>6</sup> =..	8 <sup>5</sup> =..	8 <sup>4</sup> =..	8 <sup>3</sup> =4056	16 <sup>2</sup> =256	16 <sup>1</sup> =16	16 <sup>0</sup> =1 الاصغر
------------------------------	--------------------	--------------------	--------------------	----------------------	----------------------	---------------------	------------------------------

اما التحويل من الست العشري إلى العشري نتبع نفس الطرق السابقه في التحويل من نظام إلى آخر و لناخذ مثال فورا  
 لدينا العدد

A3D أي لدينا عشر ٢٥٦ و ثلاث ١٦ و ثلاث عشر ١

و لذلك لأنه كما قلنا بأن

A=10 and D=13

أي تصبح المعادله على الشكل التالي

$$256*10+3*16+13*1= 2621$$

**- كيف نحول من العشري إلى الست عشري:**

و لناخذ فوراً على ذلك

١٩٩١

نقسم ١٩٩١ على ١٦ جواب ١٢٤ والباقي ٧

نقسم ١٢٤ على ١٦ جواب ٧ والباقي ١٢ أي C

نقسم ٧ على ١٦ جواب ٠ والباقي ٧

$$1991 = 7C7$$

**النظام الثنائي العشري المشفر:**

## Binary-Coded-Decimal

في الحقيقة هذا النظام ليس نظاما رقميا بحد ذاته أي أقصد ليس نظام مستقل كالنظام الثنائي أو الثماني... بل إنما يمزج ما بين العشري والثنائي ، وتعتمد عليه بعض الآلات الحاسبه . وكثيرمن الشرائح الإلكترونية ، فكيف يعمل هذا نظام ؟  
 و لناخذ مثال على ذلك:

و ليكن هذا الرقم العشري و نكتبه في النظام العشري الثنائي المشفر  
 ٤٦٥

فا الأربعة تساوي في النظام الثنائي ١٠٠

و الستة تساوي ١١٠

و الخمسة تساوي ١٠١



إذا العدد ٤٥٦ يكتب على الشكل التالي

**100 110 101**

أذا نستنتج أن النظام العشري المشفر هو عبارته عن النظام العشري لكن الأرقام تكتب فيه على شكل نظام ثنائي

**العمليات المنطقية في سي بلس بلس:**

يوجد في لغة C++ بعض من العمليات المنطقية logical operators التي أستخدمها لتشكيل أوامر و شروط معقدة و ذلك عن طريق الجمع بين عدة شروط بسيطة و من أبرز هذه العمليات هي

١- عملية الجمع (&&)AND :

لنفرض أننا نريد أن نكتب برنامج و قبل تنفيذ هذا البرنامج يجب التأكد من صحة شرطين معاً قبل أن ننفذ البرنامج ثم إذا تحقق الشرطين معاً نفذ المطلوب  
مثال:

لنفرض أنه لديك حساب في أحد البنوك و لديك بطاقة و أردت أن تسحب مبلغ معين فقبل أن تطلب المبلغ الذي تريده فإن المصرف الألي يطلب منك شرطين وهذان الشرطان يجب أن يتحققا معاً حتى يتم لك صرف المبلغ الذي تريده

١- أدخل أسمك.....

٢- أدخل رقم حسابك.....

أي وصف لغاريتم البرنامج سيكون على النمط التالي

إذا كان الأسم المدخل هو أحمد و كان رقم الحساب المدخل ١٧٢٥٤

إصرف المبلغ المطلوب

```
If ( name == ahmad && number == 17254)
```

```
{
```

```
نفذ
```

```
}
```

و سوف نشرح فيما بعد العملية الشرطية if في الدروس الأتية إن شاء الله

٢- العملية المنطقية (||) OR الاختيار :

لنفرض أننا عند نقطه ما ضمن برنامج و نريد التأكد من صحة شرطين أو من صحة

أحدهما قبل أن نختار المسار التالي لمتابعة التنفيذ

مثال :

لدخول كلية الهندسه يجب أن يتحقق أحد الشرطين

١- مجموع العلامات الكلي أكثر من ٨٠

أو

٢- مجموع العلامات في مادة الرياضيات أكثر من ٩٠

أي يكفي تحقق أحد الشرطين لدخول كلية الهندسه

و يكتب هذا الشرط على النحو التالي

```
If (sum_grade > 80) || (sum_mat > 90)
```

```
{
```

```
نفذ
```

```
}
```

### ٣- عملية النفي (!)Not :

عملية النفي ما هي إلا عباره عن عكس الشرط و نعبر عن هذه العمليه بالعلامه !  
و على سبيل المثال

إذا كان مجموع علامة أحد الطلاب لا يساوي ٥٠

لا يحق له دخول أي فرع يختاره الطالب

```
If (sum_grade != 50)
```

```
{
```

```
نفذ
```

```
}
```

### الأدوات الحسابيه:

في معظم لغات البرمجه يوجد أدوات حسابيه ( و في الغالب موحده ) للأجراء العمليات الحسابيه كالجمع و الطرح و الضرب و القسمة و هذا جدول في أهم الأدوات الحسابيه المستخدمه في سي بلس بلس

أسم الأداة	رمز الأداة في C++	طريقة كتابة الأداة في C++
الجمع	+	A+3
الطرح	-	a-8
الضرب	*	A*2
القسمة	/	8/4
باقي القسمة الصحيحه	%	9%2

جدول الأدوات الحسابيه

كل العمليات الحسابيه الموجوده في الجدول تستخدم في لغة سي بلس بلس بنفس الطريقه التي تستخدم في الطريقه الجبريه العاديه إلا الأداة % باقي القسمة الصحيحه التي تعطي باقي قسمة عددين صحيحين على شكل عدد صحيح و هذه الأداة لا تستخدم إلا مع الأعداد الصحيحه

مثال :  $5 \% 2 = 1$

### أدوات المقارنه :

يوجد في لغة سي بلس بلس العديد من الأدوات التي تستخدم للمقارنه بين شرطين

الشكل الجبري	رمز الأداة في C++	طريقة كتابة الأداة C++ في	معنى الأداة
<	<	$X < y$	X أكبر من y
>	>	$x > y$	X أصغر من Y
$\leq$	$\leq$	$X \leq y$	X أكبر من أو تساوي y
$\geq$	$\geq$	$x \geq y$	X أصغر من أو تساوي Y

جدول أدوات المقارنه

### رموز لغة C++:

الرموز المستخدمه في C++ هي عبارته عن

1- الحروف الأنجيزيه الكبيره ..... A,B,C,D

2- الحروف الأنجيزيه الصغيره ..... a.b.c.d

3- الأرقام العربيه الأصلية : 1.2.3.4.....

**ملاحظه :**

إن لغة C++ تميز بين الأحرف الأنجليزيه الكبيره و الصغيره  
لذلك فإن لغة C++ تعد من اللغات الحساسه للحروف  
مثال: أن المترجم في اللغه يميز الكلمه

for عن For

رغم أنه بالنسبه لنا لا يوجد أي فرق بينها لكن المترجم في لغة C++ عندما يبدأ  
في عملية الترجمة لتحويل كود البرنامج المكتوب إلى لغة الآله يفرق بين الحروف الكبيره و  
الصغيره

**أنواع الكلمات في C++:**

تنقسم الكلمات المستخدمه في هذه اللغه إلى نوعين

**١- المتغيرات:**

و هي الكلمات التي يكتبها المبرمج نفسه لكي يعبر عن المتغيرات في برنامج كإن  
يسمي مثلاً المتغير X يطلق عليه اسم sum  
و يمكن أن نعرفها على أنها أماكن في الذاكرة يمكن تخزين بيانات فيها بشكل مؤقت  
ليستعملها البرنامج أثناء مرحلة التنفيذ . وهذه المتغيرات تمتلك **إسماً و نوعاً و قيمة** في  
أماكن تخزينها

و سوف نتعرف على المتغيرات في نفسك عند تبدأ في كتابة البرامج  
و لكن يوجد بعض الشروط يجب أن نتقيد فيها للأطلاق أسماء المتغيرات وهي  
١- أن يكون الاسم المكتوب بالأحرف الأنجليزيه

٢- أن لا يحتوي الاسم على فواصل مثال:

Num

٣- أن لا يحتوي الاسم على أي رمز أو أشاره

مثل ، . ؟ " : } ÷ × ؛ < > .....

و يجوز استخدام فقط الخط التحتي "\_" فقط لا غير

٤- ان لا يكون الاسم إحدى الكلمات المحجوزه.

بعض الأمثلة الغير صحيحه لأسماء المتغيرات:

١- A fool خطأ بسبب وجود فاصل

٢- name. خطأ لأن الأسم إحتوى على رمز

٣- Типы خطأ لأن الأسم كتب في لغه غير الأنجليزيه

٤- class خطأ لأن الأسم هو إحدى الكلمات المحجوزه

## ٢- الكلمات المحجوزه:

هي الكلمات المعرفه مسبقا من قبل المترجم ل C++ و التي كما سبق و قلنا بأنه لا يجوز أستخدامها كأسماء للمتغيرات لأنها حجزت مسبقا من قبل المترجم و لكل كلمه من هذه الكلمه معنى خاص و ترمز إلى شيء معين سوف تتعرف إلى معانيهم عند البدء في كتابة كود البرنامج  
ملاحظه: أرجوا أن تنتبهوا بأن هذه الكلمات المحجوزه تكتب في الأحرف الصغيره

Sizeof	near	Static	asm	Double	long
else	do	int	While	new	auto
if	For	This	Void	Delete	Goto
Case	const	try	char	Class	Public
Private	Continue	Extern	struct	inline	float
Static	Virtual	Volatile	Frinde	enum	near
Typedef	soid	Default	inline	Overload	Unsigned
Union	Signed	Pascal	Operator	Switch	Template
Const	Register	Protected	far	Catch	char
	break	Return			

جدول الكلمات المحجوزه

**تنصيب فيجوال سي بلس بلس :**

إن عملية أعداد و تنصيب برنامج فيجوال سي بلس بلس سهله و غير معقده و هي تشبه طريقة تنصيب أي برنامج آخر

**العمل في بيئة فيجوال سي بلس بلس :****مقدمة:**

في البدايه أحب أن أقول للأخواننا الذين قد يتسائلون عن سبب إختيار برنامج فيجول سي بلس لكتابة C++ مع العلم أنه كان في إمكاننا أن نكتب برامجنا على أي برنامج آخر بلس برامج

Borland أو أي برنامج آخر محرر للغة و العمل على تلك المحررات قد يكون

مثل C++

أسهل من العمل على فيجول سي بلس بلس.

أقول لك أنت محق في ذلك و لكن أنا اخترت هذا المحرر للغة لعدة أسباب:

١- و ذلك بسبب القوه الكبيره التي يتمتع فيها هذا المحرر فيجول سي بلس بلس و قدرته على التعامل مع أكواد البرامج التي سوف تكتبها فيه إن شاء الله

٢- عندما فكروا مطوري Visual C++ قاموا في استخدام برنامج التجميع C++ العلمي التصنيف و أضافوا بيئة تطوير مع مجموعة أدوات تتيح للمطور إنتاج تطبيقات بسهولة و بسرعة لم تعرف من قبل في بيئات تطوير C++. ويندوز

٣- و أيضاً بسبب سهولة تقسيم كتابة أكواد البرامج الكبيرة التي تود كتابتها إلى عدة أجزاء و إعطاء كل جزء إلى مبرمج ليكتب كود البرنامج الجزء و من ثم تجميع هذه الأجزاء ليتكون لدينا برنامج كبير بمعنى الكلمة و هذه الطريقة تتم فقط في برمجة الكائنات الموجه و طريق الصفوف classes التي سوف نتعرف عليه لاحقاً إن شاء الله ذلك عن

٤- فيجوال سي بلس بلس تعطيك القدرة على مشاهدة خطوات تنفيذ برنامجك خطوة بخطوة

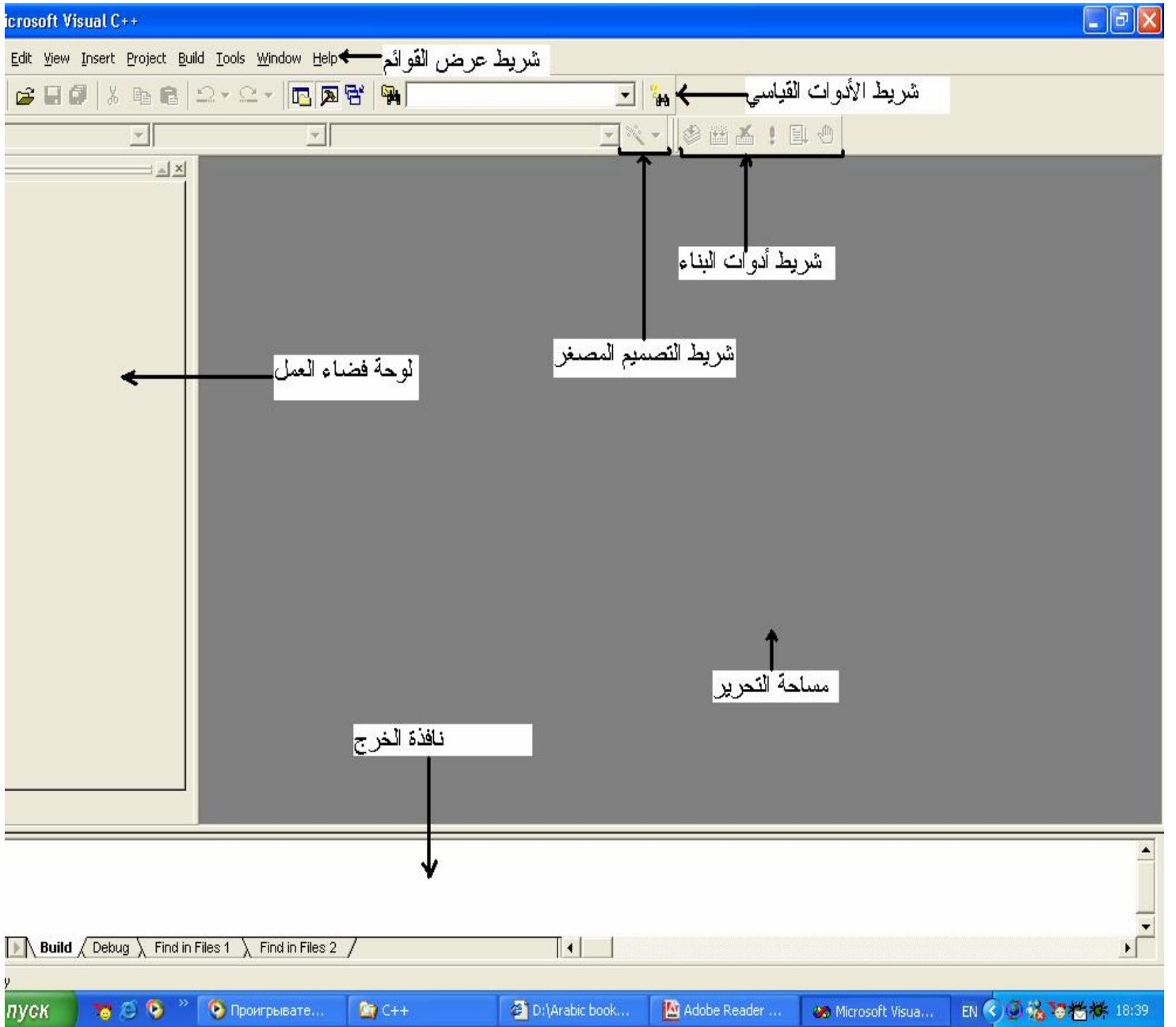
و ذلك في الضغط على F10 بعد كتابة الكود.

٥- و أيضاً بسبب سهولة الانتقال من C++ الي فيجول سي بلس بلس إذا كنت قد بدأت مع فيجول سي بلس بلس في كتابة أكواد برامجك.

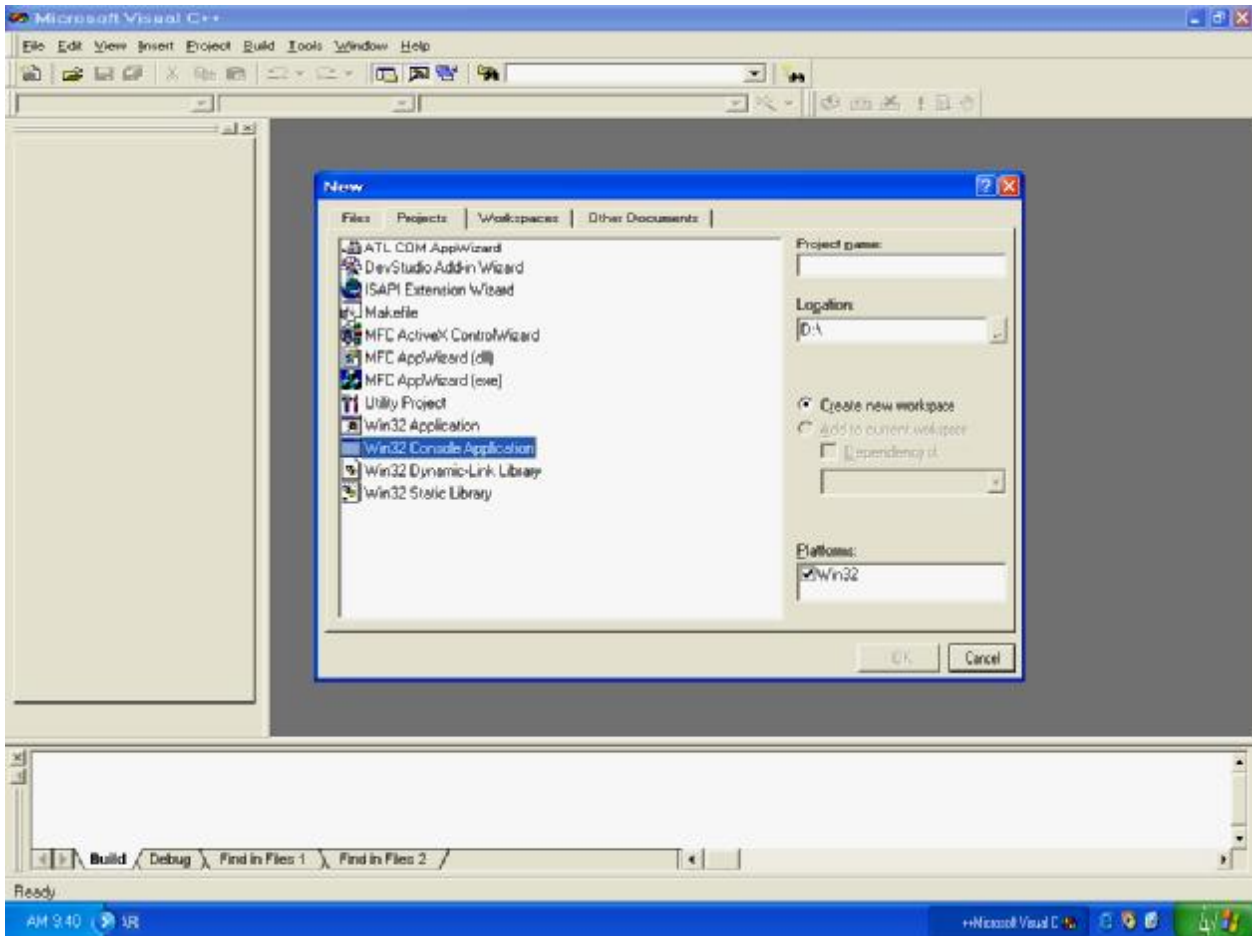
## أنشاء أول تطبيق لنا في فيجول سي بلس بلس :

سوف نتعرف في هذه الفقرة أن شاء الله عز و جل على كيفية فتح محرر اللغة فيجول سي بلس بلس و التعرف على النواظف فيه بعد عملية أعداد و تنصيب فيجول سي بلس بلس إذهب إلى  
**ابدأ < البرامج < Visual C++**  
 سوف تظهر لك و اجهة البرنامج على الشكل التالي:

### واجهة برنامج Visual C++



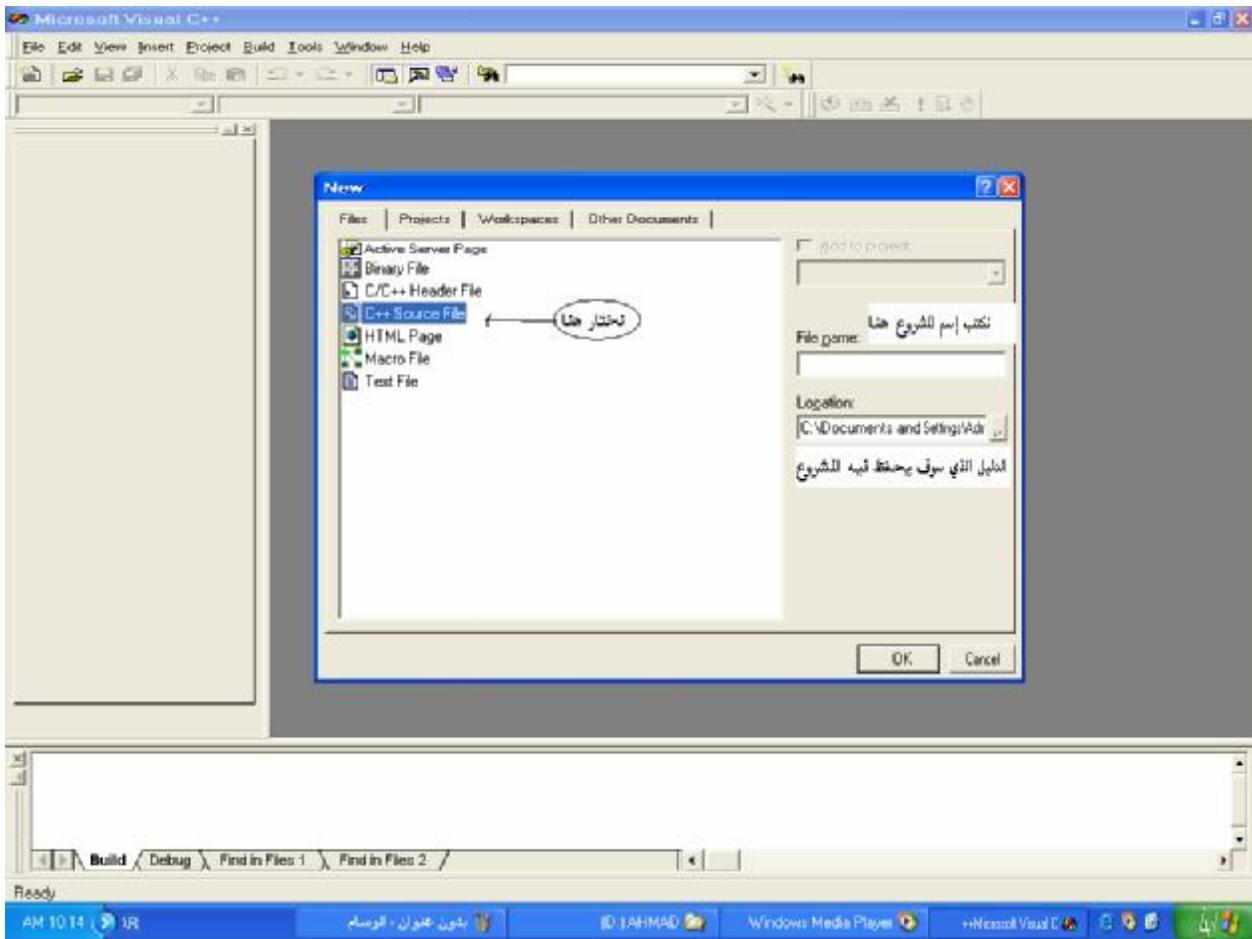
و من ثم إذهب إلى شريط عرض القوائم و أختار من هذا الشريط `new < file` حينئذ سوف يظهر لك بما يسمى في الصندوق الحواري و الذي سوف يظهر لديك على الشكل التالي :



عرض للصندوق الحواري

و هذا الصندوق الحواري يحتوي على كافة المشاريع التي يمكن عملها في  
**Visual c++**  
 و نختار من هذا الصندوق **Files** القائمة  
 فيظهر لنا الصندوق الحواري لهذه القائمة  
 على الشكل التالي:





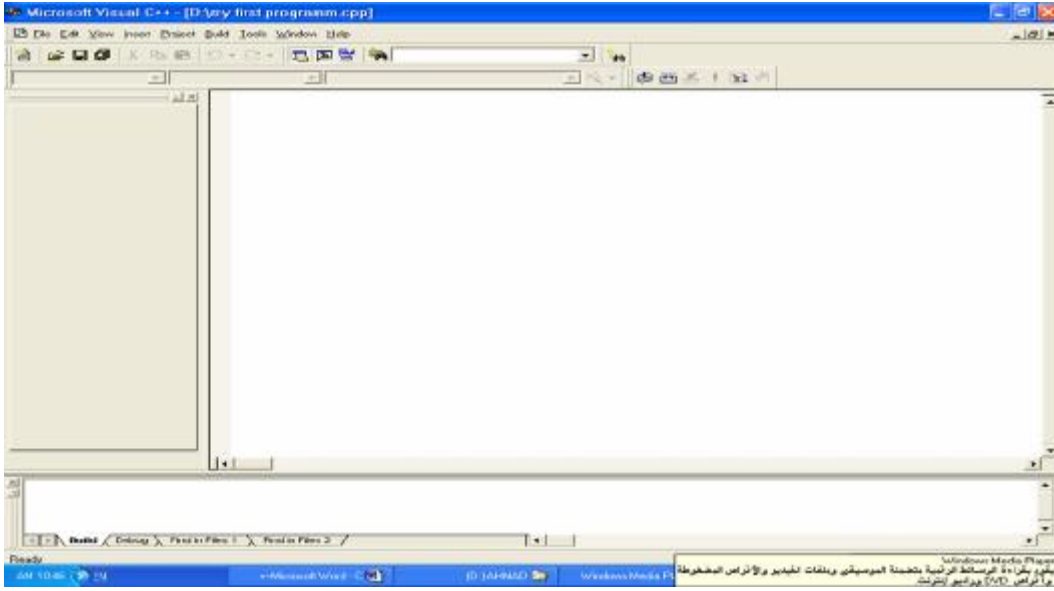
الصندوق الحواري لـ file

و نختار من هذه القائمة

## C++ source file

و ثم نكتب اسم المشروع الذي نود File name و نختار مثلاً My first programm القيام به في المربع

و بعد ذلك نكتب الدليل أي العنوان و هو المكان الذي سوف يحفظ فيه المشروع ثم نختار الزر ok فيظهر لدينا محرر كود لغة C++ أو بيئة عمل C++



محرر أكواد لغة C++

## كتابة برنامجك الأول في C++:

لنبدأ على بركة الله في كتابة أول برنامج لنا في بيئة عمل C++

```
# include <iostream.h>
main()
{
    cout<<"welcome arab to C++";
    return 0;
}
```

بعد كتابة هذا الكود لبرنامجنا الأول ننتقل إلى المرحلة الثانية و هي مرحلة الترجمة أي مرحلة اكتشاف الأخطاء إن وجدت و ذلك بالانتقال إلى شريط عرض القوائم و اختيار build و من هذه القائمة نختر

Compile

أو باستخدام اختصار لوحة المفاتيح Ctrl+f7

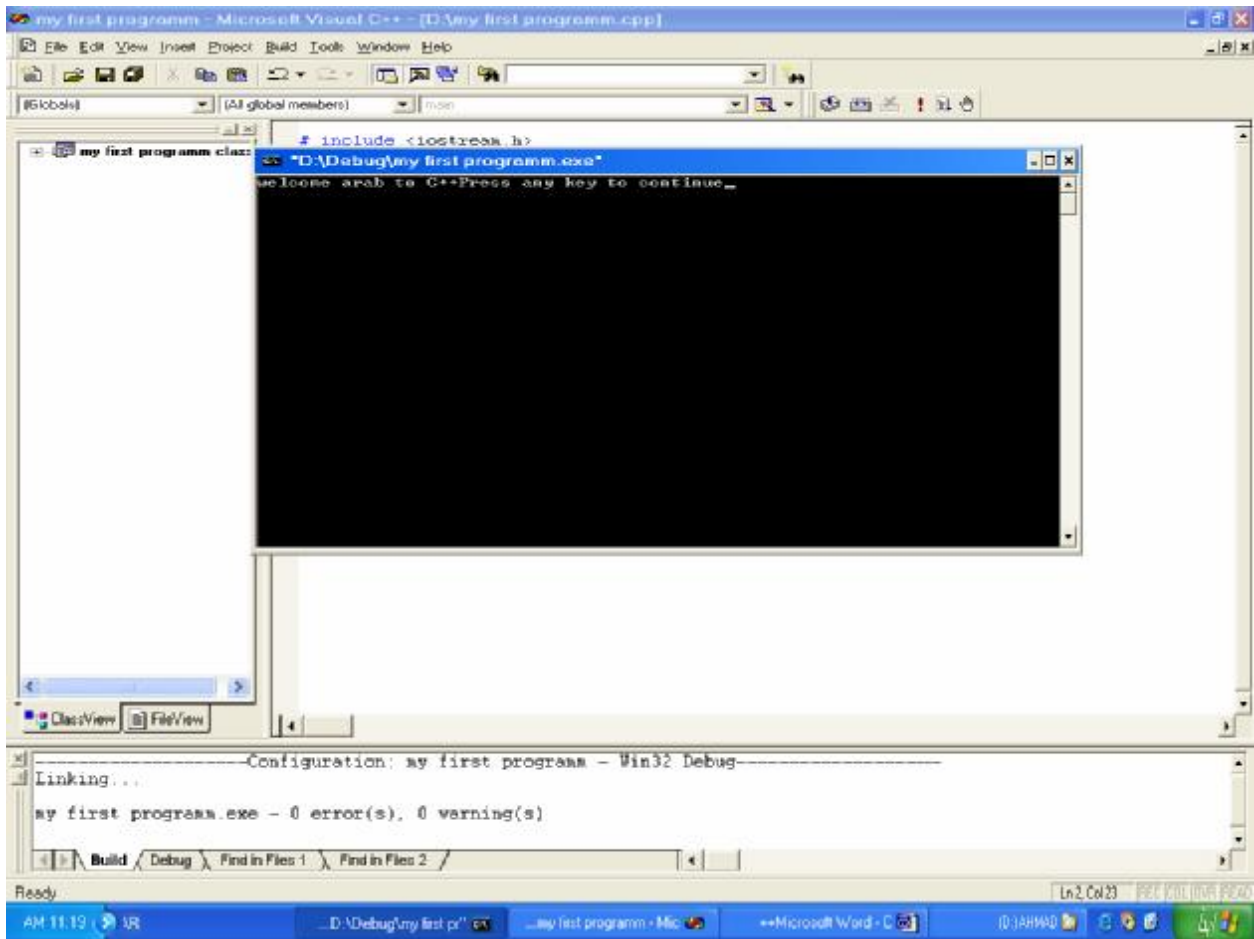
و سوف يكون مكتوب بعد هذه الكلمة اسم البرنامج الذي اخترناه .

و بعد تحقق المترجم من عدم وجود أي أخطاء في هذا الكود سوف ننتقل إلى المرحلة التالية

هي مرحلة التنفيذ و ذلك في اختيار من القائمة build < execute program

و بعد ذلك سوف تظهر لنا شاشة سوداء تظهر لنا نتيجة عملنا و ستكون على الشكل التالي

أو باستخدام إختصارات لوحة المفاتيح Ctrl+f5



تنفيذ البرنامج

أما الآن دعنا نشرح في التفصيل المراحل التي يمر فيها أي برنامج في C++ قبل أن يتم تنفيذه

#### ١- المرحلة الأولى: مرحلة الكتابة edit

في هذه المرحلة تتم فيها كتابة نص البرنامج و ذلك يتم بواسطة محرر النصوص الخاص المتوافق مع C++ الذي يستخدمه المبرمج لكتابة برامج و القيام في التعديلات و بيئة عمل

و الأضافه الحذف ثم بعد ذلك يتم تخزين البرنامج على القرص الصلب و في الذات المكان الذي حددته أنت عندما اخترت الدليل الذي حفظت به برنامجك

و عادة ما تعطي برامج C++ أسماء تنتهي في الأمتداد .c أو .ccp.

#### ٢- المرحلة الثانية: مرحلة ما قبل الترجمة preprocess

هذه المرحلة تهدف إلى التعامل مع أوامر خاصة تسمى بتوجيهات ما قبل preprocessor directives الترجمة

تحدد هذه التوجيهات جملة من المعالجات و الإجراءات الواجب تنفيذها على نص البرنامج قبل عملية الترجمة، أي قبل المرحلة الثالثة . و في هذه المرحلة تتم معالجة نص البرنامج .

#### ٣- المرحلة الثالثة : مرحلة الترجمة compile للبرنامج

في هذه المرحلة يقوم المترجم في ترجمة C++ إلى لغة الآله .حيث يولد المترجم برنامج بلغة الآله و برنامج

يخزنه على القرص الصلب

#### ٤- المرحلة الرابعة : مرحلة الوصل linking

في هذه المرحلة تتم إستدعاءات لتوابع تم تعريفها في مكان آخر مثل المكتبات المعيارية أو المكتبات الخاصة في مجموعة من المبرمجين العاملين ضمن مشروع واحد. يحتوي النص الناتج بعد عملية على فجوات بسبب هذه linking بوصل النص السابق مع التوابع الناقصة الترجمة الأجزاء الناقصة فيقوم الواصل

و كل ذلك من أجل الوصول إلى برنامج قابل للتنفيذ و ذلك سيكون تحت النظام unix الأمر الذي ينفذ عملية الترجمة و الوصل لبرنامج C++

#### ٥- المرحلة الخامسة: مرحلة الشحن load

و هذه المرحلة تتمثل في إختصار في عملية وضع البرنامج ( الذي قمنا في كتابته ) في الذاكرة و ذلك طريق الشاحن loader الذي يقوم بأخذ الملف التنفيذي و ينقله إلى الذاكرة ليتم تخزينه هناك

#### ٦- المرحلة السادسة : مرحلة التنفيذ execute

في هذه المرحلة يبدأ الكمبيوتر تنفيذ البرنامج تعليمه الواحدة تلو الأخرى حتى آخر تعليمه في برنامجنا وذلك تحت إشراف و سيطرة وحدة التحكم و المعالجة و بعد المرور في تلك المراحل يتم تنفيذ البرنامج و أظهاره لنا على الشكل الذي رأيتمه

## شرح لأسطر البرنامج :

أما الآن بعد أن تعرفنا على مراحل تنفيذ البرامج سوف نبدأ في شرح الأسطر التي قمنا في كتابتها

السطر الأول `# include <iostream.h>` يعبر عن توجيه لمرحلة ما قبل الترجمة. فالتعليمة `#` و التي `include` هي من تعليمات المعالج المبدئي يليها أما التعليمة `<iostream.h>` ذو الأمتداد `.h` تدعى هذه التعليمة بالملف الرأسي الحاوي على العمليات الخاصة للخروج و الإدخال

أما السطر التالي `Main()` فهو يعتبر جزءاً من أي برنامج C++ و هي إسم الدالة الرئيسية

و القوسان يعبران على أن `Main` كتلة برمجية تسمى `function` و أي برنامج يجب أن يحتوي على تابع أو أكثر ، أحدهما يجب أن يكون التابع `main` و في هذين القوسين يكتب أيضاً البراميتير الخاص للمتغيرات و لكن في هذه الحالة لا شيء

} يعبر القوس الكبير الأيسر عن بداية جسم التابع `main`

و السطر الذي `Cout<<"welcome arab to C++";` يليهما

هو أمر للطباعة على الشاشة و هذا الأمر معرف و موجود مسبقاً في الملف الرأسي `iostream` `cout`

<< يجب أن تتبع بعد أمر الطباعة `cout`, فهي ترمز إلى عملية الإدخال.

" " فهذه التعليمة هي عبارة عن علامة اقتباس و كل شيء يكتب داخلها سوف يظهر على الشاشة

; يجب أن ننهي كل تعليمة في فاصله منقوطة

`Return 0` تعبر عن إحدى الطرق المستخدمة للخروج من التابع `main`

أما القيمة ٠ تعبر عن إنتهاء البرنامج بشكل صحيح

{ يعبر القوس الكبير الأيمن عن نهاية التابع `main`

و هذا جدول يشرح أوامر اللغة التي أستخدمها

شرح لأمر اللغة	أمر اللغة
من تعليمات المعالج المبدئي	# include
ملف رأسي لتعريف الأوامر المستخدمة في البرنامج	<iostream.h>
إسم الدالة الرئيسية	main
قوسان بعد الداله الرئيسية للتعبير عنها	()
يعلن عن بداية جسم التابع main	{
أمر للطباعة على الشاشة	Cout<<
علامة اقتباس لطباعة الرسالة الموجودة بينهما	" "
علامة النهاية للتعليمة	;
إعلان عن إنتهاء البرنامج و الخروج منه	Return 0
قوس يعلن عن نهاية جسم التابع main	}

جدول شرح الأوامر

**ملاحظه:**

يمكن كتابة البرنامج السابق في عدة طرق و طباعة الجملة السابقة أيضاً في عدة طرق فعلى سبيل المثال

```
# include <iostream.h>
main()
{
    cout<<"welcome arab";
    cout<<" to C++";
    return 0;
}
```

و النتيجة بالطبع هي ذاتها و يمكن في الطريقة التالية:

```
# include <iostream.h>
main()
{
    cout<<"welcome \n arab to C++ \n";
    return 0;
}
```

و سوف يطبع لدينا على الشاشة على الشكل التالي:

```

C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\DESKTOP\Debug\12.exe
welcome arab
to C++
Press any key to continue_

```

ألا توافق معي بأن النص المطبوع أصبح أكثر تنسيقاً و سهولة للقراء إن لغة C++ تمنح لك المرونة بالتعبير عن التعليمات بطرق متنوعة مما يؤدي إلى طباعة الخرج على الشكل و التنسيق الذي تريده أنت و يوافق برنامجك بصورة عملية لقد إستعملنا في المثال السابق التعليمة `\n` و هذه التعليمة هي من أحد سلاسل الهروب التالية:

فالحرف `\` backslash يسمى بحرف الهروب Escape character وهو يدل على أن هناك حرف يجب إظهاره و هذا الحرف يجب أن يكون من إحدى أحرف سلاسل الهروب فالحرف `\n` يدل على بداية سطر جديد new line و هذا الحرف يسبب في نقل المؤشر إلى بداية سطر جديد على الشاشة

### ملاحظة:

لاحظ أخي الكريم بأن حرف الهروب يكتب بين علامتي الاقتباس " " و أنه لا يطبع على الشاشة لأنه من تعليمات اللغة. و هذه بعض الأحرف الأكثر إستعمالاً:



معنى الحرف	أحرف سلسلة الهروب
سطر جديد. ووضع المؤشر في بداية سطر جديد	\n
تنسيق الخرج على شكل جدول منسق.	\t
وضع المؤشر على بداية السطر الحالي. أما لا يطبع كل شيء كتب قبله في فيجوال C++	\r
alert إصدار صوت النظام	\a
يطبع على الشاشة حرف الشق المعكوس \	\\
يطبع على الشاشة علامة الأقتباس "	\"

بعض أحرف سلاسل الهروب

أمثلة على أحرف الهروب:  
١ - الحرف \n

```
# include <iostream.h>
main()
{
    cout<<"\nalsalam alekom\n";
    cout<<"welcome arab\n to C++\n";
    cout<<"\n with \n";
    cout<<"namer alnserat\n\n\n";
    return 0;
}
```

```
C:\Documents and Settings\Administrator\Desktop\Debug\12.exe
alsalam alekom
welcome arab
to C++

with
namer alnserat

Press any key to continue
```

## ٢- الحرف \t:

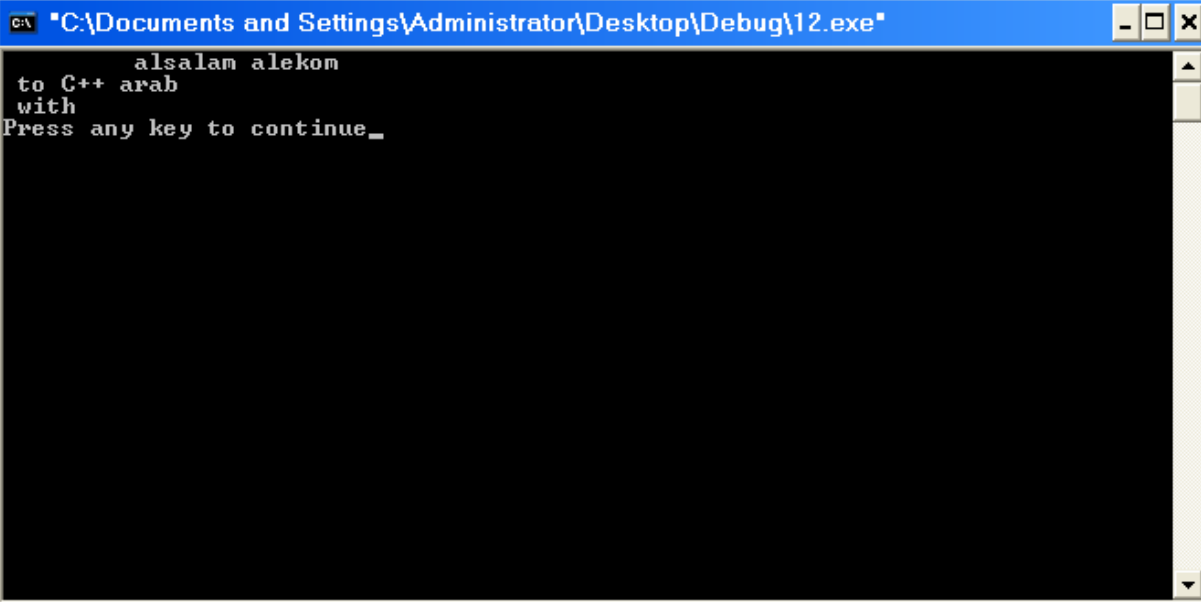
```
# include <iostream.h>
main()
{
    cout<<"\t alsalam alekom \n";
    cout<<"welcome arab\t to C++ \n";
    cout<<"\t with \n";
    cout<<"\t namer alnserat \n\n\n";
    return 0;
}
```

```
C:\Documents and Settings\Administrator\Desktop\Debug\12.exe
alsalam alekom
welcome arab\t to C++
\t with
\t namer alnserat
Press any key to continue
```

## الحرف

٣- \r: إذا كتب هذا الحرف في بورلاند سي بلس بلس فإنه سوف يعيد المؤشر إلى بداية السطر الحالي دون الانتقال إلى سطر جديد أما في الإصدار فيجوال سي بلس بلس فإن تأثيره هو عدم طباعة ما قبله ضمن علامتي الاقتباس

```
# include <iostream.h>
main()
{
    cout<<"\t alsalam alekom \n";
    cout<<"welcome arab\r to C++ \n";
    cout<<" with \n";
    cout<<"\t namer alserat \r";
    return 0;
}
```



#### ٤- الحرف \a:

و وظيفة هذا الحرف هو إصدار صوتي ، صوت النظام. يمكنك أن تجربه في نفسك.

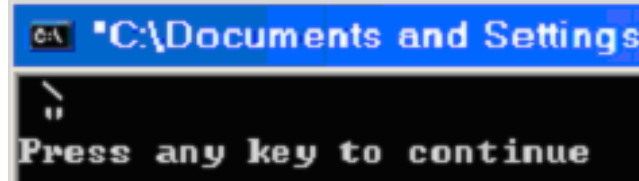
#### ٥- الحرف \\:

هذا الحرف يساعدك على طباعة حرف الشق المعكوس | على الشاشة.

#### ٦- الحرف \"

هذا الحرف يطبع لك على الشاشة علامة الاقتباس " . و هذا مثال يجمع بين الحرف الخامس و السادس.

```
# include <iostream.h>
main()
{
    cout<<" \\" ;
    cout<<"\n";
    cout<<" \" " ;
    cout<<endl;
    return 0;
}
```



**ملاحظته :**

لقد إستعملت في المثال السابق التعليمة `Cout<<endl;` وهذه التعليمة إذا لاحظت معي أخي الكريم بأن لها نفس تأثير حرف الهروب `\n` فهي تنقل المؤشر إلى سطر جديد و لا كن لاحظ معي بأن `endl` لا تكتب بين علامتي الأقتباس " " .

**مثال ثاني للتوضيح:**

سوف نكتب برنامجنا الثاني و الذي يجب أن ينفذ عملية حسابية بسيطة و هي عملية جمع عددين صحيحين بحيث يطلب منا البرنامج بإدخال العدد الأول ثم يطلب منا في إدخال العدد الثاني و بعد ذلك يجب أن يظهر على الشاشة نتيجة الجمع

```
# include <iostream.h>
main()
{
    // هذا هو البرنامج الثاني
    // برنامج جمع عددين صحيحين
    int number1, number2, sum; // الاعلان عن المتغيرات

    cout<<" enter the number1: \n "; // خرج على الشاشة طلب إدخال الرقم الأول
    cin>>number1; // مجرى الدخل من أجل الحصول على القيمة المدخلة من قبل المستخدم

    cout<<"enter the number2: \n "; // طلب إدخال الرقم الثاني
    cin>>number2; // الحصول على القيمة المدخلة للعدد الثاني

    sum = number1 + number2 ; // إعلان المترجم عن العملية الحسابية

    cout<<" the sum is "<< sum <<endl; // طباعة الناتج
    cout<<endl;
    return 0;
}
```

**شرح كود البرنامج:**

السطر الأول و الثاني من الكود قد تم شرحهما مسبقاً في المثال الأول لكنك تلاحظ شيء جديد و هو الإشاره `||` أو الشرح في اللغة العربية بعها و أيضاً لاحظت عدم ظهور هذه الكتابة في البرنامج بعد التنفيذ وذلك لأن المترجم يقوم بتجاهل أي شيء كتب بعد إشارة التعليق `||`.

في الحقيقة إن الذي تراه أمامك من كتابة في اللغة العربية ما هو إلا شرح مبسط للأسطر البرنامج أو تعليق يكتبه المبرمج خلال كتابة كود البرنامج حتى تساعده على قراءة و فهم

البرنامج. أو أي شخص آخر يريد أن يقرأ كود البرنامج فإن ذلك سوف يسهل عليه الكثير من العناء من أجل فهم الكود.

ولكن هناك مشكله واحده في هذه التعليمة و هي ماذا سوف نعمل إذا أردت أن تكتب تعليق مكون من عدة أسطر حيث إن إشارة التعليق هذه || تعطيق الحق في كتابة تعليقك على سطر واحد فقط. ولحل هذه المشكلة يوجد في سي بلس بلس تعليمة لكتابة التعليق على أكثر من سطر و هي

```
/*
كتابة التعليق الذي تريده هنا
*/
```

أما السطر التالي

```
int number1 , number2 , sum;
```

فهو يعني عن إعلان المترجم عن المتغيرات التي ننوي إستخدامها . وهذه المتغيرات سوف يتم حجز موضع خاص لها في الذاكرة و في هذا المكان الذي حجز في الذاكرة سوف يخزن فيه قيمه ليتم استخدامها فيما بعد .

و لكن قبل أن يمكن استعمال المتغير يجب التصريح عنه . أي تحديد نوعه و إعطاؤه إسماً و التصريح على نوع المتغير هو أمر هام جداً لأنه يعرف نطاق القيم المسموحة التي يستطيع النوع تخزينها و نوع العمليات التي يمكن تنفيذها على المتغير. و قد أعلننا عن هذه المتغيرات على أنها من النمط Int و هذا يعني أن المتغيرات سوف تقبل قيم صحيحه فقط مثل ٠،٥،٩، ٢٢٨....

أما إذا أردت ان تستخدم أعداد حقيقية مثل ٢,٥، ٠,٣، ٠,٠٣، -٢١٣، ٣١٥ حينئذ يجب عليك استخدام النمط float أو النمط double للأعداد الحقيقية الكبيرة أما لحالة الأحرف فيجب استخدام النمط Char

نوع النمط	حالة استخدام النمط
Int	للأعداد الصحيحة
Float	للأعداد الحقيقية
Double	للأعداد الحقيقية الكبيرة
char	لحالة الأحرف

جدول أنواع الأنماط

أما التعليمة:

```
Cout<<"enter the number1 \n";
```

فهو أمر لطباعة هذه الرسالة على الشاشة Enter the number1 التي تطلب من المستخدم بإدخال الرقم الأول. أي وكان البرنامج يقول للمستخدم أدخل العدد الأول فأنا جاهز للإستقباله.

أما التعليمة:

```
Cin>>number1;
```

هي عبارة عن عملية الدخل من القيمة التي أدخلها المستخدم للمتغير الأول و في مثالنا هي القيمة العدد ٢ . ويمكن أن نعبر عن التعليمة السابقة كما يلي التعليمة cin تعطي القيمة المدخلة إلى المتحول Number1 و ذلك طبعاً يتم بعد الضغط على زر الإدخال لكي يتم إرسال العدد إلى الكمبيوتر . ثم ليقوم ذلك الكمبيوتر بإسناد هذا العدد إلى المتغير الأول number1 بعد

### ملاحظة:

يجب أن تنتهي كل تعليمة في الفاصلة المنقوطة ; وهذه الفاصلة المنقوطة تسمى بمؤشر نهاية التعليمة . وإن عدم وضعها بعد كل تعليمة سوف يظهر لك المترجم خطأ خلال مرحلة الترجمة.

أما التعليمتين

```
Cout<<"enter the number2: \n";
```

```
Cin>>number2 ;
```

فهما يقومان على نفس مبدأ التعليمتين السابقتين التين سبق شرحهما و لآكن في هذه الحالة هذه التعليمتين تستقبل القيمة للمتغير الثاني و ترسل القيمة للمتغير الثاني

Number2

أما بالنسبة للسطر

```
Sum = number1 + number2 ;
```

فهي في كل بساطة عبارة عن عملية حسابية بسيطة تقوم بحساب مجموع المتغيرين Number1 و المتغير Number2 ثم بعد إجراء العملية الحسابية تقوم بإرسال الناتج إلى المتغير Sum و ذلك عن طريق عملية الإسناد =

السطر

```
Cout<<" the sum is " << sum << endl ;
```

عبارة عن طبع الناتج للمتغير sum على الشاشة.

### ملاحظة:

هل لاحظت معي أخي الكريم أن cin و cout هي عبارة عن عملية حوار و خطاب بين الكمبيوتر و المستخدم لذلك سميت هذه العملية بالبرمجة الخطابية **Interactive computing**

### ملاحظة هامة:

يوجد في لغة C++ عمليتين هما تقريباً متشابهتين و يكثر وقوع في الخطأ بينهما  
 ١- العملية الأولى هي عملية المقارنة (=):  
 و التي تعني أن الطرف الأول من المعادلة يأخذ قيمة الطرف الثاني من المعادلة.

٢- العملية الثانية هي عملية المساواة (=):  
 و هذه العملية تعني أن الطرف الأول من المعادلة يساوي الطرف الثاني من المعادلة.

و يتمثل هذا الخطأ في التبديل بين العمليتين (=) و (=) أن مشكلة الخطأ بينهما تعد من المشاكل الخطيرة . و ذلك أن عملية الخطأ بينهما لا تسبب خطأ قواعدي . بل يتم ترجمة التعليمة التي ورد فيها الخطأ بشكل طبيعي و صحيح لكن البرنامج يعطي بعد تنفيذه أخطاء منطقية أثناء مرحلة التنفيذ .

### تسلسل تطبيق العمليات الحسابية في C++:

إن لغة C++ هي لغة تستخدم قواعد الأولوية في إجراء العمليات الحسابية و هذه الطريقة قد تختلف عن الطريقة الجبرية المعروفة لدينا  
 إن أي معادلة نكتبها في هذه اللغة فإن C++ تبدأ في إجراء العملية الحسابية بطريقة متسلسلة على ثلاث مراحل:

١- المرحلة الأولى :

إجراء العمليات الحسابية الموجودة ضمن الأقواس

٢- المرحلة الثانية :

إجراء العمليات الحسابية بالنسبة إلى الضرب و القسمة و باقي القسمة الصحيحة

## ٣- المرحلة الثالثة :

إجراء العمليات الحسابية بالنسبة إلى الجمع و الطرح .

كما هو معروف فإن كل العمليات الحسابية تتم من اليسار إلى اليمين.

**مثال:**

ليكن لدينا هذه المعادلة و نريد أن نجري لها العملية الحسابية في C++

$$Y=2 \times (8 \div 4) + 6 \times 3 - 4$$

الشكل الجبري للمعادلة

$$Y=2*(8/4)+6*3-4$$

C++ حسب لغة

خطوات إجراء العمليات الحسابية في C++ :

١- الخطوة الأولى :

$$(8 / 4) = 2$$

$$y = 2 * 2 + 6 * 3 - 4$$

٢- الخطوة الثانية :

$$2 * 2 = 4$$

$$y = 4 + 6 * 3 - 4$$



### 3-الخطوة الثالثة :

$$6 * 3 = \text{18}$$

$$y = 4 + 18 - 4$$

### الخطوة الرابعة :

$$4 + 18 = \text{22}$$

$$y = 22 - 4$$

### 5-الخطوة الخامسة :

$$22 - 4 = 18$$

### 6-الخطوة السادسة :

$$Y = 18$$

**شرح خطوات تنفيذ برنامجنا السابق بالنسبة إلى الذاكرة :**

لقد شرحنا في المثال السابق انه قبل ان نستخدم المتغير يجب التصريح عنه وهذه المتغيرات سوف يتم حجز موضع خاص لها في الذاكرة و في هذا المكان الذي حجز في الذاكرة سوف يخزن فيه قيمه ليتم استخدامها فيما بعد أثناء تنفيذ البرنامج .  
ففي الذاكرة يتم لكل متغير اسم name و نمط type و قيمة value

عندما صرحنا عن المتغيران Number1 و Number2 و sum فنحن في هذه الطريقة قومنا في حجز أماكن لهم في الذاكرة. و يمكن ان نعبر عن ذلك بالصورة التالية :

Number1	<input type="text"/>
Number2	<input type="text"/>
Sum	<input type="text"/>

و بعد أن قمنا بإدخال لكل متغير قيمة عددية فإن هذه القيمة سوف توضع في موضع الذاكرة المخصص لها مع العلم أن هذه المواضع تكون جنباً إلى جنب في الذاكرة .  
ويمكن أن نعبر عن ذلك بالشكل التوضيحي التالي:

Number1	<input type="text" value="2"/>
Number2	<input type="text" value="3"/>
Sum	<input type="text"/>

بعد أن قمنا بإدخال القيم إلى المتغيرات و أخذت مواقع لها في الذاكرة هذا يعني أن البرنامج قد حصل على قيمتي المتغيرين  
بعد ذلك يقوم البرنامج بعملية الجمع و يضع النتيجة في موضع sum  
بعد ذلك سوف تبدوا المواضع في الذاكرة على الشكل التالي :

Number1	2
Number2	3
Sum	5

# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

### الجزء الثاني :

### بنى التحكم

#### مقدمة:

إن لغة C++ هي ليست لغة تستخدم فقط من أجل طبع نص على الشاشة أو إجراء عملية حسابية. إن لغة C++ تعد من اللغات العالية المستوى فهي تزودك في بنى للتحكم يمكنك من إتخاذ القرارات و تنفيذ الحلقات .حتى تتمكن من إنشاء برنامج قوي بمعنى الكلمة يجب أن يكون هذا البرنامج متفاعل بين الحاسب و المستخدم .بحيث أن البرنامج يعرض على المستخدم عدة خيارات و يعطي المستخدم إختيار واحد أو عدة خيارات من الخيارات المعروضة أمامه ثم تنفيذ الذي إختاره المستخدم.

لذلك فإن لغة C++ تزودك في عدة جمل لاتخاذ القرارات و تنفيذ الحلقات.

و هذه هي جمل القرارات و الحلقات الموجودة في لغة C++:

#### ١ - جمل إتخاذ القرارات في C++ :

أ- الجملة `if`.

ب- الجملة `switch`.

#### ٢ - جمل تنفيذ الحلقات في C++:

أ- الجملة `for` .

ب- الجملة `while` .

ت- الجملة `do- while` .

بالإضافة إلى ذلك فإن ++c تزودك أيضاً في  
الجملة break للإنتهاء حلقة فوراً

و الجملة continue للعودة إلى بداية الحلقة لتنفيذ التكرار التالي.

و سوف نبدأ بعون الله عز و جل في شرح لتلك الجمل كل واحدة في التفصيل مع إراد  
الأمثلة على كل واحدة التي أرجوا من الله عز و جل أن تكون واضحة الشرح و مبسطة  
بحث أن يكون هذا الكتاب مفيداً لجميع إخواننا المسلمين في مشارق الأرض و مغاربها

أ- الجملة if (إذا):

إن الأكثر شيوعاً للإتخاذ القرارات في ++c هي عن طريق استعمال الجملة if  
و هذه الجملة تنقسم إلى ثلاث أنواع أو اتجاهات

١- أحادي الاتجاه.

٢- ثنائي الاتجاه.

٣- متعدد الاتجاهات.

أما الآن فدعنا يا أخي الكريم أن نبدأ في النوع الأول، أحادي الاتجاه  
و هذه الجملة تكتب في ++c على الشكل التالي:

(شرط تنفيذ التعليمة) If

{

جسم التعليمة if

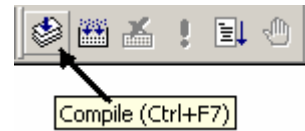
و هنا نضع التعليمات و الأوامر التي نريد من البرنامج أن ينفذها إذا تحقق الشرط

}

أما الآن دعنا نضرب مثال على هذه التعليمات ثم نقوم بشرح كيف تنفيذ المترجم لها

### مثال:

لنفرض أن جامعة الهندسة أعلنت في بداية السنة الدراسية عن بدء قبول الطلاب الراغبين في الإنتساب إليها و وضعت عدة شروط للانتساب و كان إحدى هذه الشروط بأن يكون مجموع علامات الطالب لا يقل عن ٧٠ في المئة  
دعنا نحاول كتابة هذا البند للقبول على شكل برنامج على C++. و بعد كتابة كود البرنامج



نضغط على

للتحقق من عدم وجود أخطاء في كود البرنامج، فإذا أظهر لنا المترجم هذه الرسالة التي تعلن عن عدم وجود أخطاء، و التي تكون في أسفل برنامج visual C++

```
Compiling...
Skipping... (no relevant changes detected)
if.cpp
if.obj - 0 error(s), 0 warning(s)
```

ثم نضغط على



لتنفيذ البرنامج بعد التحقق من صحة الكود المكتوب

و هذا هو كود البرنامج بعد كتابته و عملية التنفيذ ، سوف يظهر على الشكل التالي:

```
# include <iostream.h>
main()
{
    //برنامج مثال للبنية
    //الإعلان عن المتغيرات
    int grade;

    cout<<" enter the grade: ";
    cin>>grade;

    if (grade>= 70)
    {
        cout<<"welcome to University \n";
    }

    cout<<endl;
    return 0;
}
```

```
C:\Documents and Settings\Administrator\Desktop\Debug\12.exe
enter the grade: 70
welcome to University
Press any key to continue_
```

### برنامج للجملة If

### شرح للأسطر البرنامج:

سوف أبدأ في شرح الأسطر الجديدة الموجودة في هذا البرنامج فقط و لن أكرر شرح لبعض الأوامر و التعليمات التي تم شرحها مسبقاً للإبتعاد عن التكرار. فبعد الإعلان عن المتغير grade و الذي يرمز في هذا المثال إلى درجة علامة الطالب و كتابة أوامر الدخل و الخرج بدأنا في كتابة شرط دخول كلية الهندسة عن طريق الجملة if

```
If(grade > = 70)
{
    cout<< "welcome to university \n "
}
```

إن لغة C++ هي لغة تنفيذ التعليمات المكتوبة ضمن البرنامج على شكل حلقة متسلسلة أي تبدأ في تنفيذ البرنامج من أول سطر. الأول ثم الثاني و الثالث..... الخ

و بناء على ذلك فإن المترجم عندما يصل إلى الجملة if فإنه يقوم من التحقق من شرط تنفيذ جملة إتخاذ القرار (إذا) if الموجودة داخل القوسين الصغيرين الذين يأتين بعد if فوراً.

فإذا تحقق الشرط عندئذ المترجم يعطي النتيجة true أي صح و ذلك يعني نفذ الأوامر و التعليمات الموجودة داخل جسم الجملة if

أما إذا كان الشرط غير محقق فهذا يعني أن المترجم يعطي النتيجة false أي خطأ و ذلك يعني لا تنفيذ أي شيء داخل جسم البنية if . و بمعنى آخر كأن المترجم يقوم بتجاهل كل الأوامر و التعليمات المكتوبة داخل جسم الجملة if .

و الإنتقال إلى التعليمة التي تليه في كود البرنامج

أما بالنسبة إلى برنامجنا

فعندما طلب منا بإدخال درجة علامة الطالب .و أدخلنا الدرجة ٧٠ ففي هذه الخطوة قام المترجم في التحقق من الشرط و بعد التحقق و جد أن درجة علامة الطالب المدخلة تتوافق مع الشرط أي true عندئذ قام في تنفيذ التعليمة الموجودة داخل جسم if و قام في طبع الرسالة الموجودة بين علامتي الإقتباس " welcome to university " .

و لكن ماذا سوف يحصل إذا قمنا في إدخال قيمة أقل من ٧٠

حينئذ فإن الشرط لا يكون محقق أي false و لن يقوم البرنامج في تنفيذ أي شيء.



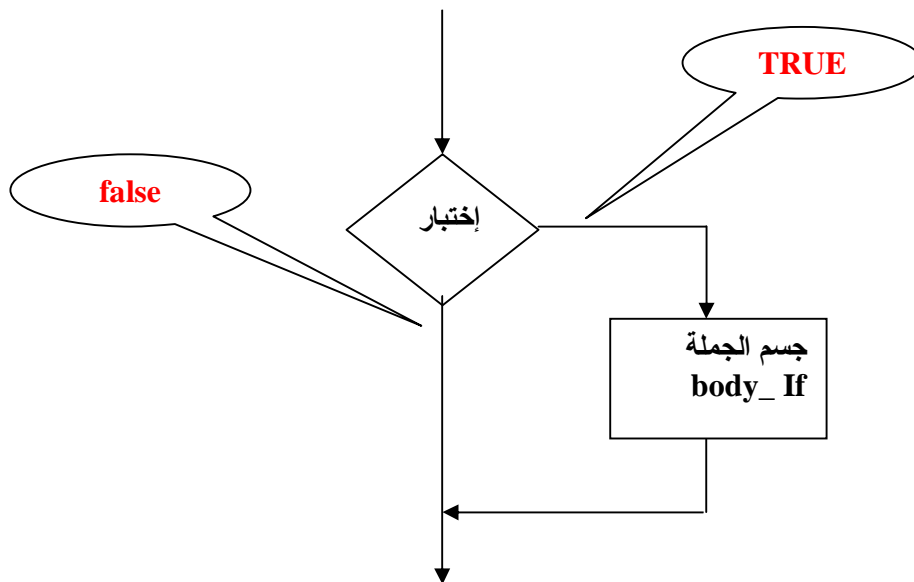
**ملاحظة:**

إن من الأخطاء التي يقع فيها معظم إخواننا المبرمجون هي وضع الفاصلة المنقوطة بعد قوسين الجملة `if` و ذلك لأنه كما تعودنا فإنه في `C++` نضع فاصلة منقوطة عند نهاية كل تعليمة.  
مثال خطأ:

```
If( grade > = 70 );
```

**رسم توضيحي لتنفيذ اختبار الجملة `if` أحادية الاتجاه**

يستخدم مبرمجون لغة `C++` قبل البدء في كتابة كود برامجهم المخططات التوضيحية للتعبير عن مجرى تنفيذ البرنامج و ذلك لكي يكون كود البرنامج أقرب إلى تصورنا العقلي و المنطقي، ليتم بعدها عملية التحويل إلى ما يكافئها في لغة `C++`.



شكل توضيحي لتنفيذ اختبار أحادية الاتجاه

٢-ثنائي الاتجاه: و هذا هو القسم الثاني من الجملة if و هو يكون على الشكل التالي if/else .

**If**(شرط تنفيذ التعليمة)

{

جسم التعليمة if

و هنا نضع التعليمات و الأوامر التي نريد من البرنامج أن ينفذها إذا تحقق الشرط

}

**Else**

{

و هنا نكتب التعليمة التي يجب على البرنامج أن ينفذها إذا لم يتحقق الشرط

}

هل تذكرون برنامجنا السابق في الجملة if أحادية الاتجاه .

عندما أردنا إدخال القيمة التي هي أقل من ٧٠ ماذا أخرج لنا البرنامج . في الحقيقة لم يخرج شيء ، مع العلم أن الشرط كان في دخول جامعة الهندسة أن تكون علامة الطالب ٧٠ أو أكثر .

و الذي أريده منك أخي الكريم أن تفهم شيء مهماً و هو أنه في الجملة if أحادية الاتجاه

**إذا كان الشرط محققاً فإنه سوف يتم تنفيذ التعليمة التي كتبناها في جسم if**

**و لكن إذا لم يكن الشرط محققاً فإنه لن يتم تنفيذ أي شيء إطلاقاً.**

و في الحقيقة في بعض الأحيان قد تكون هذه بعض المشاكل التي تواجهنا في كتابة

البرنامج ، بحيث نريد من البرنامج أن ينفذ شيء معين في حال عدم تحقق الشرط

ففي حال المثال السابق عندما قام أحد الطالب في إدخال علامته و هي ٦٥ و لم يقم

البرنامج في تنفيذ أي شيء و الطالب واقف ينتظر الرد

لذلك يجب علينا التفكير في كتابة شيء بحيث أن البرنامج يقوم في تنفيذ تعليمة أخرى في حال عدم تحقق الشرط للجملة if  
حينئذ يجب علينا استخدام الجملة if\else و التي تعطينا مجال لكتابة تعليمة أخرى في حال عدم تحقق الشرط

### مثال:

أما الآن دعنا أخي الكريم نقوم في تعديل البرنامج السابق بحيث إذا قام الطالب في إدخال قيمة أصغر من ٧٠ نكتب تعليمة بحيث تخرج على الشاشة رسالة تعلن الطالب بأن علامته أقل من ٧٠ و لا يقبل في كلية الهندسة. ثم نقوم في تنفيذ البرنامج كما تعلمنا في المثال السابق. و بعد التنفيذ يطلب منا البرنامج بإدخال العلامة فنقوم في إدخال علامة أقل من ٧٠ حينئذ سوف يخرج لنا على الشاشة الرسالة التالية كما هو موضح على الشكل التالي:

```

main()
{
    //if/else لهذا مثال للبنية
    الإعلان عن المتغيرات
    cout<<" enter the grade: \n ";
    cin>>grade;
    if(grade>=70)
    {
        cout<<"welcome to University \n";
    }
    else
    {
        cout<<" excuse? your grade less than 70";
    }
    cout<<endl;
    return 0;
}

```

```

"D:\BOOK_C++\Debug\if.exe"
enter the grade:
65
excuse? your grade less than 70
press any key to continue

```

### مثال للبنية if/else

### شرح للأسطر البرنامج:

كما قلت سابقاً سوف لن أقوم في شرح للأوامر التي سبقا شرحها سابقاً، و سوف أقصر في شرح التعليمات الجديدة

ففي هذا البرنامج لم تقم سوى في إضافة التعليمة else . و وظيفتها هي **فقط** في حال عدم تحقق شرط الجملة if المكتوب بين القوسين الصغيرين الموجودين بعد if مباشرةً أي مسرى التنفيذ سيكون على الشكل التالي :

بعد إدخال العلامة يقوم البرنامج في التحقق من شرط الجملة if إذا تحقق الشرط فإنه سوف يتم تنفيذ التعليمات الموجودة داخل جسم الجملة if و إذا لم يتحقق الشرط فإن البرنامج ينتقل إلى الخطوة التالية و هي تنفيذ التعليمة الموجودة داخل جسم البنية else

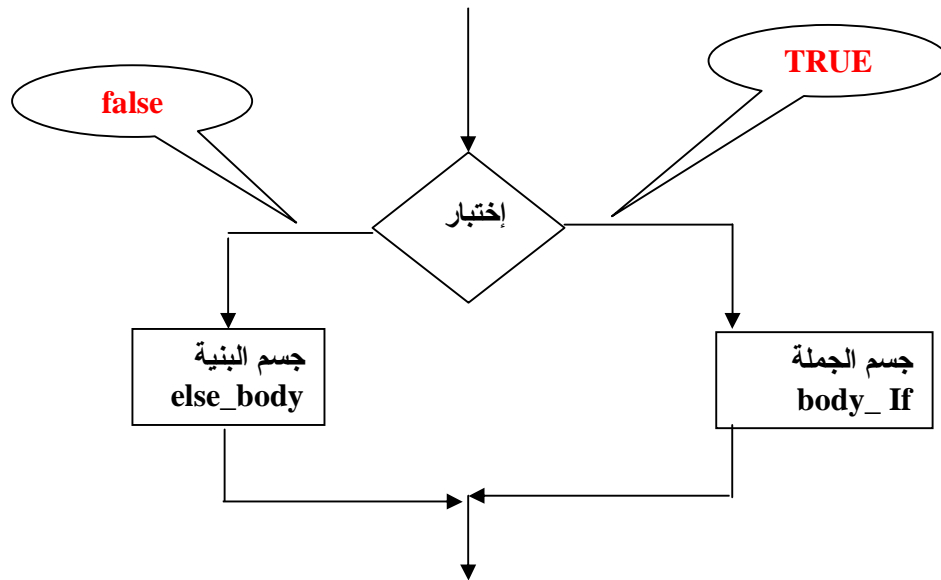
أما بالنسبة لبرنامجنا فإن التنفيذ سيكون على الشكل التالي:

إذا أخي الكريم كما ترى عندما أدخل الطالب الدرجة ٦٥ قام البرنامج في التحقق من شرط الجملة `if` و وجد بأن العلامة المدخلة غير متوافقة مع الشرط حينئذ قام البرنامج في تنفيذ التعليمة الموجودة ضمن البنية `else` عندئذ قام البرنامج بإخراج رسالة على الشاشة تخبره بأن علامة الطالب أقل من ٧٠ و لا يجوز له في الإنتساب إلى كلية الهندسة.

### ملاحظه:

أخي الكريم يجب أن تنتبه إلى الأقواس الكبيرة `{ }` و التي تحصر تعليمات البنية أو الجملة. و بالتالي يمكن بسهولة في قراءة جسم الجملة أو إضافة أوامر أخرى لجسم الجملة إذا ما أردت الإضافة أو التطوير لبرنامجك فيما بعد مع العلم أنه يمكن كتابة جسم الجملة دون وضع الأقواس الكبيرة `{ }`. و لكنني أردد بالتأكيد على وضعها لتجنب الوقوع في الأخطاء.

## رسم توضيحي لتنفيذ اختبار الجملة if ثنائية الاتجاه:



شكل توضيحي لتنفيذ اختبار ثنائية الاتجاه

## متعدد الاتجاهات:

في الحقيقة إن القسم الثالث من الجملة If ما هي إلا الجملة if في الحالة الثنائية و لكنها تختلف بأن هذه الطريقة تأتي بشكل متتالي لأخذ قرار بأسلوب متعدد الاتجاهات. أي في حالة متعدد الاتجاهات نكتب أكثر من if/else بطريقة متتالية و متداخلة تتناسب حسب البرنامج الذي نريد كتابته .

و هو يكون يشبه الشكل التالي:

```

If(شرط تنفيذ التعليمة)
{
    if التعليمة
    و هنا نضع التعليمات و الأوامر التي نريد من البرنامج أن ينفذها إذا تحقق الشرط
}
Else
{
    If(شرط تنفيذ التعليمة الموجودة ضمن else)
    {
        if التعليمة
        و هنا نضع التعليمات و الأوامر التي نريد من البرنامج أن ينفذها إذا تحقق الشرط
    }
}
Else
{
    و هنا نكتب التعليمة التي يجب على البرنامج أن ينفذها إذا لم تتحقق الشروط
    السابقة الذكر
}

```

نعم أخي الكريم أعلم أن الأمر بدأ يصعب عليك و قد تختلط عليك هذه التعليمات المتداخلة ، و لكن تذكر أنه من أراد العلى تحمل مشقة صعود الجبال و كما قيل سابقاً  
 إن الضغط و الحرارة تصنع الألماس  
 و لكن لا تقلق فإن هذه المرحلة هي من المراحل السهلة و سوف نأخذ مثال و إن شاء الله  
 فإن الأمور سوف تتضح بعد هذا المثال.

و سوف نأخذ مثالنا السابق و سوف نحاول تطويره ليصبح برنامج حقيقي يعبر عن مستوى فهمنا للغة C++ .

### مثال:

سوف نعود أيضاً إلى مثالنا السابق

كما نتذكر أخي الكريم في المثال السابق و الذي كان عبارة عن: أن جامعة الهندسة أعلنت في بداية السنة الدراسية عن بدء قبول الطلاب الراغبين في الإنتساب إليها و وضعت عدة شروط للانتساب و كان إحدى هذه الشروط بأن يكون مجموع علامات الطالب لا يقل عن ٧٠ في المئة.

و كان أيضاً إحدى هذه الشروط التي و وضعت الجامعة كمساعده للطلاب بحيث أنه إذا كان مجموع الطالب أقل من ٧٠ في المئة يجوز له في الانتساب إلى كلية الهندسة و ذلك بشرط إذا كان مجموع علامات الطالب في مادة الرياضيات أكثر أو يساوي ٨٠ في المئة و مجموعة العام لا يقل عن ٥٠ في المئة.

أي إذا كان مجموع الطالب العام أكثر أو يساوي ٥٠ في المئة و مجموعة في مادة الرياضيات أكثر أو يساوي ٨٠ في المئة حينئذ يجوز له الانتساب إلى كلية الهندسة، أما الآن دعنا أخي الكريم أن نعبر عن ذلك عن طريق لغتنا C++:

سوف نكتب كود البرنامج ثم نشرح أسطر البرنامج إن شاء الله



```
include <iostream.h>
main()
```

Execute Progr

```
int grade;
float grmat;
cout<<" enter the grade: \n ";

cin>>grade;

if(grade>=70)
{
    cout<<"welcome to University \n";
}
else
{
    cout<<"excuse! your grade less than 70 \n";
    cout<<"NO how your grade mathemateic \n";
    cout<<" enter the grade math:\n";
    cin>> grmat;
    if( grmat>=80 && grade>=50)
    {
        cout<<"welcome to University:\n";
    }
    else
    {
        cout<<"excuse? your grade less than 70 \n";
        cout<<"and too your grade mathemateic less than 80.\n"
    }
}
return 0;
```

برنامج للجملة if المتعددة الاتجاهات



Compile (Ctrl+F7)

و بعد كتابة كود البرنامج نضغط على

للتحقق من عدم وجود أخطاء في كود البرنامج، فإذا أظهر لنا المترجم هذه الرسالة التي  
تعلن عن عدم وجود أخطاء، و التي تكون في أسفل برنامج visual C++

```
Compiling...
Skipping... (no relevant changes detected)
if.cpp
if.obj - 0 error(s), 0 warning(s)
```

ثم نضغط على

لتنفيذ البرنامج بعد التحقق من صحة الكود المكتوب



و هذا هو البرنامج بعد عملية التنفيذ ، سوف يظهر على الشكل التالي:

```
"D:\book_C++\Debug\if.exe"
enter the grade:
55
excuse! your grade less than 70
NO how your grade mathemateic
enter the grade math:
82
welcome to University:
Press any key to continue_
```

يظهر لك البرنامج بعد مرحلة التنفيذ شاشة سوداء تخبرك عن درجة الطالب و أنا قمت في إدخال الدرجة ٥٥ و التي هي أقل من الدرجة المطلوبة لدخول الجامعة و التي قام البرنامج بإخباري في ذلك و لكن أيضاً قام البرنامج في إعطاء فرصة أخرى و طلب مني في إدخال علامتي في مادة الرياضيات و أنا أدخلت العلامة ٨٢ حينئذ طبع البرنامج رسالة على الشاشة بإخباري بأنه قد تم قبولي في كلية الهندسة

## شرح للأسطر البرنامج:

بعد الشروع في كتابة كود البرنامج و كتابة `#include<iostream.h>` الملف الرأسي الذي يحوي على العمليات الخاصة للدخول و الخروج و الذي بدأنا في تعريف المتغيرات التي سوف نستخدمها في برنامجنا

```
Int grade;
```

```
Float grmat;
```

أما المتغير الأول `grade` و الذي يعبر عن درجة علامة الطالب العامة و التي هي كما هو معروف من النوع الأعداد الطبيعية

أما المتغير الثاني `grmat` و الذي يرمز إلى درجة علامة الطالب في مادة الرياضيات و أنا هنا وضعتها من النوع `float` و قد وضعت احتمال بأن تكون علامة الطالب في مادة الرياضيات فيها أعداد عشرية مثل ٨١,٣  
أما السطران

```
Cout<<"enter the grade: \n";
```

```
Cin>> grade;
```

فهما كما تعرف أخي الكريم ما هما إلا عملية الدخول و الخروج و التي تطبع على الشاشة رسالة تطلب من الطالب بإدخال علامته و من ثم إرسال القيمة المدخلة إلى الذاكرة عن طريق التعليم `cin`

ثم بعد ذلك شرعنا في كتابة شرط و جسم الجملة `if`

```
If(grade >= 70)
```

```
{
```

```
cout<<"welcome to University ";
```

```
}
```

و نحن في هذه الطريقة قمنا بإخبار البرنامج عن طريق شرط الجملة `if` أنه إذا كانت العلامة المدخلة من قبل الطالب تساوي أو أكبر من ٧٠ نفذ التعليم الموجودة داخل جسم الجملة `if` أي قم في طبع رسالة على الشاشة ترحيب للطلاب لقبوله في الجامعة

## أما التعليمة التالية:

```

Else
{
    cout<<"excuse! your grade less than 70" \n;
    cout<<"NO how your grade mathemateic" \n;
    cout<<" enter the grade math: \n;
    cin>> grmat;
    if(grmat >= 80 && grade >= 50)
    {
        cout<<" welcome to University ";
    }
}

```

فنحن في هذه الطريقة قمنا في إخبار البرنامج أنه إذا قام الطالب في إدخال علامة أقل من ٧٠ ، و البرنامج سوف يعرف ذلك بعد التحقق من شرط الجملة if فإذا كانت النتيجة FALSE ذلك يعني لا تنفذ شيء ضمن بنية جسم الجملة if و انتقل إلى التعليمة else لتنفيذ الأوامر و التعليمات الموجودة ضمن جسم التعليمة else و نحن فقد كتابنا ضمن البنية else كالتالي:

أولاً قمنا في طبع رسالة على الشاشة تخبر الطالب أن مجموع درجاته أقل من ٧٠ و من ثم قمنا في طرح سؤال على الطالب و هو كم درجة علامتك في مادة الرياضيات و ثم طلبنا منه بإدخال درجت علامته في مادة الرياضيات أما التعليمة cin>>grmat; فتقوم بإرسال درجة مادة الرياضيات إلى الذاكرة ثم شرعنا بعد ذلك في كتابة جملة if الشرطية ضمن التعليمة else

```
If(grmat >= 80 && grade >= 50)
```

من أجل التحقق أن درجة الطالب الدخلة في مادة الرياضيات هي تساوي أو أكبر من ٨٠

و درجة علامة الطالب التي أدخلناها مسبقاً و التي ما زالت محفوظة في الذاكرة على

شكل متغير grade أنها هي أيضاً تساوي أو أكبر من ٥٠ في المئة

فإذا تحقق الشرطان معاً نفذ التعليمات الموجودة ضمن جسم الجملة if أي إذا كان نتيجة الاختبار TRUE اطبع على الشاشة رسالة ترحيب للطالب أما إذا كانت النتيجة FALSE نفذ السطر التالي

```
Else
{
    cout<<"excuse? your grade less than 70 \n";
    cout<<"and too your grade mathematic less than 80.\n";
}
```

أي اطبع على الشاشة تخبر الطالب بأنه لا يستطيع الانتساب إلى كلية الهندسة و ذلك لأن مجموع علامته العامة أقل من ٧٠ و لأنه أيضاً درجة علامته في مادة الرياضيات أقل من ٨٠ ، وسوف تظهر النتيجة على الشكل التالي:

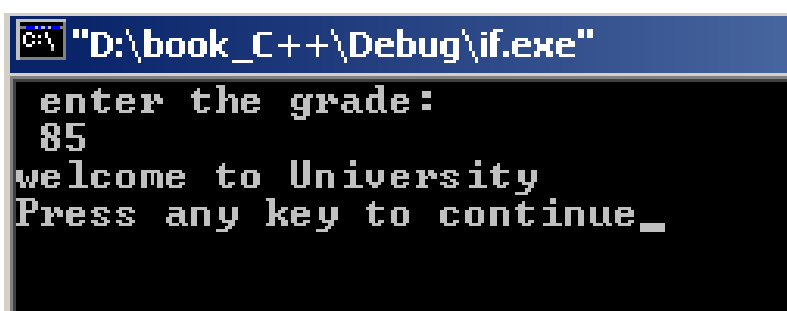
```
"D:\book_C++\Debug\if.exe"
enter the grade:
60
excuse! your grade less than 70
NO how your grade mathematic
enter the grade math:
70
excuse? your grade less than 70
and too your grade mathematic less than 80.
Press any key to continue_
```

و هذه التعليمة أيضاً لا تنفذ إلا فقط في حال عدم تحقق الشروط السابقة الذكر أما السطر الأخير return 0; و ذلك يعني الخروج من البرنامج .

و هذا هو كل برنامجنا هو ليس صعباً و كل ما عليك هو التركيز و فهم كيفية كتابة التعليمات و كيفية ترتيبها بشكل مرتب و مفهوم و بشكل يناسب البرنامج الذي تريد كتابته.

### ملاحظه:

ماذا لو كان درجة علامة أحد الطلاب ٨٥ أي المجموع العام و قام في إدخال هذه الدرجة فماذا سوف يحصل لبرنامجنا، دعنا نجرب و نرى النتيجة

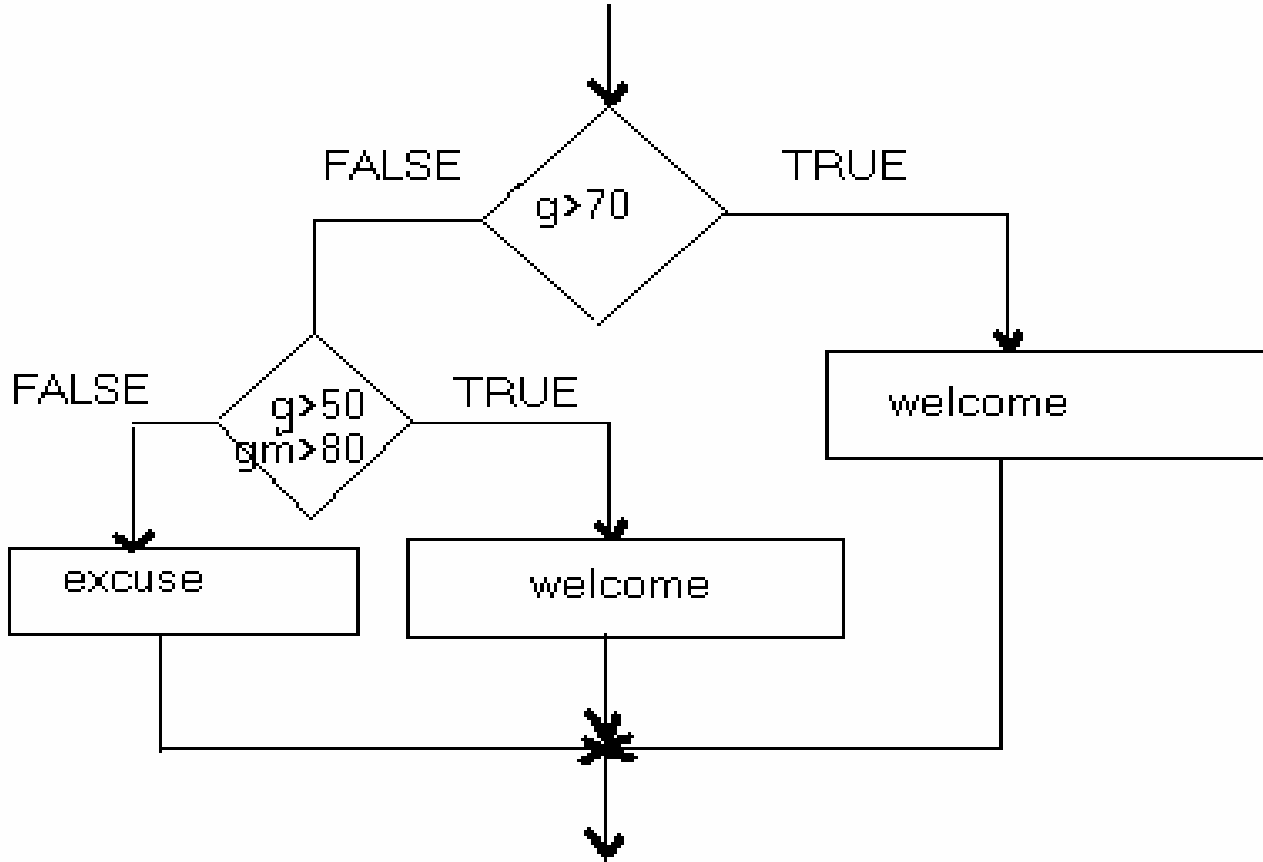


```
"D:\book_C++\Debug\if.exe"
enter the grade:
85
welcome to University
Press any key to continue_
```

كما تلاحظ أخي الكريم عندما أدخلنا درجة أكبر من ٧٠ قام البرنامج فوراً في طبع رسالة ترحيب على الشاشة و تجاهل كل التعليمات المكتوبة في كود البرنامج لماذا ؟

حاول الإجابة على هذا السؤال بنفسك!!!!

رسم توضيحي لتنفيذ اختبار الجملة if متعددة الاتجاه:  
و ذلك لمثالنا فقط.



و كما قلت أخي الكريم بأن هذا الرسم التوضيحي هو فقط لمثالنا لأنه يصعب أن أعطيك مخطط للجملة if المتعددة الاتجاهات و ذلك بسبب تشعب هذه الطريقة و وجود ألف طريقة و طريقة لكتابة if المتعددة الاتجاهات و ذلك طبعاً كما قلت بما يناسب طبيعة و متطلبات البرنامج الذي تود كتابته.

## تعديل المثال السابق:

لاحظ معي أخي الكريم في مثالنا السابق

إذا أردنا أن ندخل علامة درجة الطالب العامة أقل من ٥٠ فماذا تلاحظ ؟

```

C:\Documents and Settings\Administrator\...
enter the grade
45
exuse! your grade less than 70
No how your gared mathemateic:
enter the grade math:
  
```

عندما

قمنا في

إدخال

العلامة

العامة

للطالب و

التي هي أقل من ٥٠ (كما نتذكر فإن الدرجة ٥٠ أو أكثر هي الشرط لمتابعة شروط  
القبول في الجامعة)

قام البرنامج بعد ذلك بطلب إدخال علامة الرياضيات ؟

مع العلم أن العلامة المدخلة أقل من ٥٠ و هذا يعني أنه لا يحق للطالب الانتساب إلى  
الجامعة مهما كانت علاماته الأخرى . حتى لو كانت درجة علامة في مادة الرياضيات  
١٠٠ من مئة .

إذا يجب أن نجري تعديل على البرنامج بحيث إذا قام الطالب درجة علامة أقل من ٥٠  
فورا يقوم البرنامج بإعلام الطالب بأنه لا يحق له الانتساب إلى الجامعة .  
و عدم تنفيذ أي مرحلة أخرى و الخروج من البرنامج من دون المتابعة.  
و بعد التعديل يصبح البرنامج على الشكل التالي:



```
# include <iostream.h>
main()
{
    int grade;
    float grmath;

    cout<<" enter the grade: \n ";

    cin>>grade;

    if(grade>=70)
    {
        cout<<"welcome to University \n";

    }
    else
    {
        if(grade<50)
        {
            cout<<"Excuse! your drade less than 50!\n";
            return 0;
        }
        cout<<"excuse your grade less than 70!:\n";
        cout<<"But how your grade int the mathmatic!:\n";

        cout<<"enter the grade mathmatic:\n";
        cin>>grmath;

        if(grmath>=80 && grade>=50)
        {
            cout<<"welcome to University\n";
        }
        else
        {
            cout<<"Excuse! your grade less than 70! \n";
            cout<<"and ecxuse your grade mathmatic less than 80\n";
        }
    }

    return 0;
}
```

و بعد التحقق من عدم وجود أخطاء ، شغل البرنامج و أدخل العلامة ٤٥ أو أي علامة أقل من ٥٠ سوف يخرج لك على الشاشة على الشكل التالي :

```

C:\Documents and Settings\Administrator\
enter the grade:
45
Excuse! your drade less than 50!
Press any key to continue

```

و كما تلاحظ بعد إدخال العلامة ٤٥ قام البرنامج فوراً في إبلاغ الطالب بعد قبوله في الجامعة لأن مجموعة العام أقل من ٥٠ ، و من ثم الخروج من البرنامج دون المتابعة .

### شرح أسطر التعديل:

في الحقيقة عندما قمنا في تعديل البرنامج حتى أصبح على الشكل الذي تراه ، لم نقم في إختراع صاروخ ، بل كل ما في الأمر هو أن أضفنا ثلاثة أسطر لا غير ضمن جسم الجملة else الأولى و هي :

```

if(grade<50)
{
cout<<"Excuse! your drade less than 50!\n";
return 0;
}

```

قمنا في كتابة الجملة if للتحقق من العلامة العامة المدخلة بحيث إذا كانت أقل من ٥٠

يجب الخروج من البرنامج فوراً و ذلك عن طريق return 0;

و return 0; تعني هنا الخروج من البرنامج بشكل صحيح إذا تحقق الشرط .

و هذا كل شيء فقط لا غير!!!!

(اللهم إنا نعوذ بك من علم لا ينفع)

**مثال ثاني للتوضيح:**

أما الآن دعنا أخي الكريم نضرب مثال ثاني على الجملة if المتعددة الاتجاهات حتى يتضح لك مفهوم هذه التعليمة

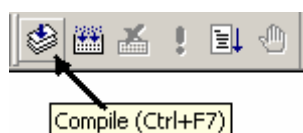
أما عن فكرة البرنامج الذي سوف نقوم في كتابته فهي عبارة عن عملية تحويل بسيطة من وحدة القياس سم إلى الإنش و عملية العكس أي من الإنش إلى سم و كما هو معروف أن كل واحد إنش تساوي ٢,٥٤ سم و سوف أكتب كود البرنامج ثم أقوم في شرح للأسطر البرنامج

```
# include <iostream.h>
int main ()
{
    float fac=2.54; //إعلان عن متغيرات
    float x, in, cm;
    char ch=0;

    cout<<"enter the length:\n";
    cin>>x; //مجرئ الدخل للقيمة المدخلة
    cin>>ch; //مجرئ الدخل للحرف المدخل
    if(ch == 'i')
    {
        //عملية التحويل إلى سم
        in=x;
        cm=x*fac;
    }
    else
        if(ch == 'c')
        {
            //عملية التحويل إلى إنش
            cm=x;
            in=x/fac;
        }
    else
        in=cm=0;
    cout<<in<<"in"<<cm<<"cm\n"<<endl;

    return 0;
}
```

## مرحلة التنفيذ:



بعد كتابة كود البرنامج نضغط على

للتحقق من عدم وجود أخطاء في كود البرنامج، أو يمكن بالنقر على Ctrl+F7 فإذا أظهر لنا المترجم هذه الرسالة التي تعلن عن عدم وجود أخطاء، و التي تكون في

أسفل برنامج visual C++

```
Compiling...
Skipping... (no relevant changes detected)
if.cpp
if.obj - 0 error(s), 0 warning(s)
```

نضغط على



لتنفيذ البرنامج بعد التحقق من صحة الكود المكتوب

و بعد كتابته و عملية التنفيذ سوف يطلب منا البرنامج بإدخال قيمة عددية ثم تحديد نوعها هل هيا سم أو إنش و ذلك عن طريق وضع الحرف i بعد القيمة المدخلة و الذي يرمز إلى وحدة القياس إنش، أو وضع الحرف c و الذي يرمز إلى وحدة القياس سم وسيظهر لديك على الشاشة على الشكل التالي:

```
"D:\book_C++\Debug\if.exe"
enter the length:
5i
5in=12.7cm

Press any key to continue_
```

أو إذا قمت في إدخال C بعد القيمة المدخلة فإنه سوف يظهر لديك على الشاشة الشكل التالي:

و كما أنه يمكننا استخدام أرقام عشرية

كالتالي:

```

C:\ "D:\book_C++\Debug\if.exe"
enter the length:
5.032i
5.032in=12.7813cm

Press any key to continue_

```

أو يمكننا تحويل عدد عشري من سم إلى الإنش :

```

C:\ "D:\book_C++\Debug\if.exe"
enter the length:
6.34c
2.49606in=6.34cm

Press any key to continue

```

## شرح للأسطر البرنامج:

بعد كتابة السطر `<iostream.h> #include` الملف الرأسي و الذي يحوي على معلومات ضرورية من أجل الدخول و الخروج بدأنا في تعريف المتغيرات لدينا و هي على الشكل التالي

```
Float fac =2.54
```

و المتغير `fac` الذي يرمز إلى وحدة القياس إنش و هو قيمة ثابتة .  
 لاحظ معي أخي الكريم بأني قلت بأن هذا المتغير هو أيضاً قيمة ثابتة لماذا ؟  
 لأنه عندما قمت بإعطاء المتغير اسم و نوع `float` أعطيته أيضاً قيمة ابتدائية و هي القيمة ٢,٥٤ و هذا الأمر هام جداً و أرجوا أن تنتبه إليه وسوف أشرح لاحقاً موضوع القيم الابتدائية و أهميته.  
 أما السطر التالي:

```
Float x , in, cm;
```

و هي أيضاً تصريح عن متغيرات  
 فالمتغير `x` يرمز إلى القيمة العددية التي سوف يدخلها المستخدم  
 أما المتغير `in` فهو عبارة عن المتغير الذي سوف نحفظ فيه ناتج التحويل إلى الإنش ثم عرض هذه القيمة على الشاشة بعد عملية التحويل  
 أما المتغير `cm` فهو عبارة عن المتغير الذي سوف نحفظ فيه ناتج التحويل إلى سم ثم عرض هذه القيمة على الشاشة بعد عملية التحويل  
 أما بالنسبة إلى متغيرنا الأخير

```
Char ch =0 ;
```

فهو يرمز إلى الحرف المرافق للقيمة العددية التي سوف يدخلها المستخدم للتعبير عن كيفية عملية التحويل من أي شيء إلى أي شيء

أي هل من سم إلى الإنش أم من الإنش إلى سم، وهذا المتغير له قيمة ابتدائية كما تلاحظ  
أما السطر التالي:

```
Cout<<"enter the length:\n";
```

رسالة عرض على الشاشة تطلب من المستخدم إدخال قيمة عددية  
السطر التالي يعبر عن

```
Cin>>x;
```

مجرى الدخل و وظيفته إرسال القيمة العددية التي قام المستخدم في إدخالها إلى الذاكرة  
أما عن مجرى الدخل الثاني

```
Cin>>ch;
```

فهو عبارة عن الحرف المدخل من قبل المستخدم و المرافق للعدد للتعبير عن نوع التحويل  
الذي سوف يقوم به البرنامج من و إلى، و هذا الحرف سوف نحدده في شرط الجملة if  
أما عن السطر التالي

```
If( ch == ' i ' )  
{
```

فهذه التعليمة بنية الاختيار if هي التي تعنى في عملية التحويل من الإنش إلى سم  
فقد و وضعنا ضمن شرط الجملة أنه إذا كان الحرف المدخل \_ المرافق للعدد \_ هو الحرف  
i أي TRUE حينئذ نفذ التعليمات الموجودة ضمن جسم الجملة if  
و لاحظ معي أخي الكريم بأننا استخدمنا هنا علامة المساواة = =  
و ليس عملية المقارنة =

أي أن الطرف الأول من المعادلة يجب أن يساوي الطرف الثاني، و الطرف الثاني من هو  
عبارة عن الحرف i و لكن هذا الحرف في الحقيقة ليس قيمة ثابتة ، و لكي نستطيع كتابته  
على شكل ثابت يجب وضعه بين فترتين على الشكل التالي 'i'

**ملاحظة:** لكي نعبر عن حرف على أنه قيمة ثابتة يجب وضعه ضمن فترتين ''

أما القوس { فهو عبارة عن بداية جسم الجملة if  
أما السطر التالي

```
In = x ;
```

في هذا السطر أسندنا القيمة العددية التي قمنا في ادخلها إلى المتغير in لكي نستخدم هذه القيمة المسندة في عملية التحويل إلى سم من خلال السطر التالي

```
Cm = x * fac ;  
}
```

كما قلت فإن هذا السطر عبارة عن عملية حسابية بسيطة للتحويل من الإنش إلى سم  
و ذلك كما ترى فإن هذه المعادلة تعني أن

كل واحد سم يساوي القيمة العددية المدخلة ضرب وحدة القياس إنش  
أما القوس الكبير } فهو يعبر عن نهاية جسم الجملة if

السطر التالي

Else

كما قلنا في شرط الجملة if إذا كان الحرف المدخل المرافق للعدد هو الحرف i نفذ  
التعليمات الموجودة ضمن جسم الجملة if أما إذا كان الحرف المدخل ليس الحرف i أي  
FALSE حينئذ نفذ التعليمات الموجودة ضمن جسم التعليمة else  
و ضمن جسم التعليمة else و ضعنا جملة if التالية

```
If( ch == ' cm ' )  
{
```

هذه جملة if الشرطية و كتبنا ضمن القوسين الذين يحويان شرط الجملة if أنه إذا كان  
الحرف المدخل المرافق للعدد المدخل من قبل المستخدم هو الحرف cm (TRUE) نفذ  
التعليمات الموجودة ضمن جسم الجملة if



أما السطر

$$Cm = x ;$$

في هذا السطر أسندنا القيمة العددية التي قمنا في ادخلها إلى المتغير cm لكي نستخدم هذه القيمة المسندة في عملية التحويل إلى إنش من خلال السطر التالي

$$In = x / fac ;$$

و هذا السطر عبارة عن عملية حسابية بسيطة للتحويل من سم إلى الإنش و ذلك كما ترى فإن هذه المعادلة تعني أن

كل واحد إنش يساوي القيمة العددية المدخلة تقسيم وحدة القياس إنش و نهاية إذا كانت القيم المدخلة غير صحيحة كأن يقوم المستخدم بإدخال قيمة عددية دون إدخال الحرف المرافق أي FALSE نفذ التعليمة التالية else

$$In = cm = 0 ;$$

أي أخرج على الشاشة تخبر المستخدم بأن  $in = cm = 0$

أما بالنسبة إلى آخر سطر

$$Cout \ll in \ll " in = " \ll cm \ll " cm / in " \ll endl ;$$

ما هو إلا عبارة عن إظهار نتيجة العمليات الرياضية التي قمنا بها ، عملية التحويل، على الشاشة.

و هذا هو ببساطة كل شيء عن البرنامج و الذي أرجوا من الله عز و جل أن أكون قد استطعت أن أوصل لك هذه المعلومات من القلب إلى القلب، و إذا لم تستطع فهم ذلك فهذا

يعني أن العيب فيا أنا ، وأرجوا أن تدعوا الله سبحانه و تعالى بأن يفتح لي من أبواب رحمته و علمه .

## ب - الجملة **switch**:

في الحقيقة إن الجملة **switch** ليس جملة قرار جديدة و غريبة عليك إذا كنت قد قرأت و تعلمت كيفية استخدام الجملة **if** .

قد تقول كيف أنها غير غريبة علي و أنا ما زلت مبتدأ و لم أقرأ أبداً عن **switch** و للتو فقط انتهيت من تعلم **if** الشرطية،

أقول لك نعم أخي المتعلم إن جملة القرار **switch** ما هي إلا عبارة عن جملة القرار **if** المتعددة الاتجاهات ، و سوف تلاحظ ذلك في نفسك بعد تعلم كيفية استخدام **switch** ، و لكن **switch** تختلف عن **if** أن الأولى أكثر تنظيماً و تنسيقاً .

إن استخدام جملة اتخاذ القرار **if** أمر مهم في كل برنامج ، و يمكن القول أنه لا يخلو برنامج من الجملة **if** ، ولكن ألا تلاحظ معي أنه إذا أردنا كتابة برنامج يحتوي على كثير من الاختيارات و استخدمنا **if** المتعددة الاتجاهات بأن الأمر بدأ يصعب علينا و قد يوقعنا في كثير من الأخطاء ، هذا بالإضافة إلى إضاعة الكثير من الوقت .

لذلك أوجدوا جملة الاختيار **switch** من أجل تجنب الوقوع في الأخطاء و إضاعة الوقت في ترتيب شروط و اختيارات متداخلة و متشابكة .

## تعريف الجملة **switch** :

الجملة **switch** تمكنك من اختيار فرع واحد فقط من عدة فروع، و كل فرع يكتب ضمن تسمية تسمى **case** ، و من ثم تنفيذ هذا الفرع الذي قام المستخدم في اختياره، (و هي تساعدك على عملية إنشاء القوائم)

و إلا سوف يتم تنفيذ الفرع **default** أوتوماتيكياً بشكل ذاتي ، إذا كان اختيار المستخدم لا يوافق أي تسمية من تسميات **case** .

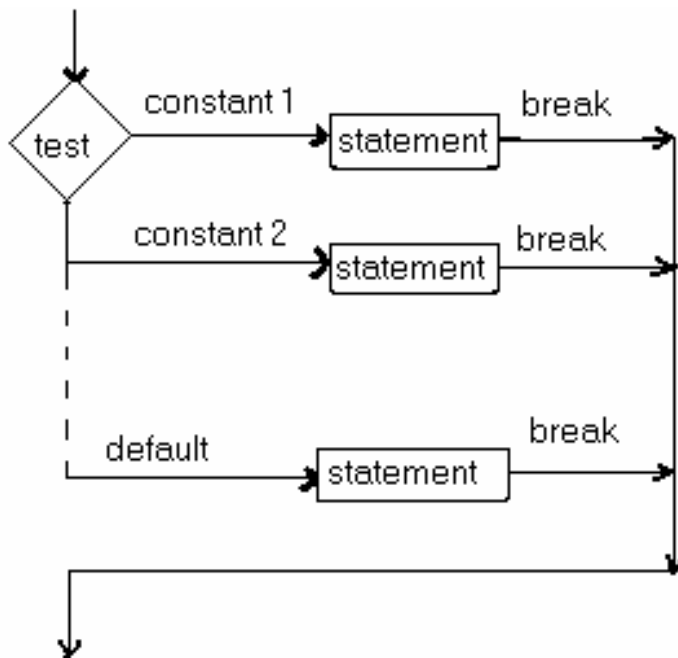
و هذا هو التعبير القواعدي في الشكل العام للجملة switch حسب لغة C++

```

switch (expression)
{
  case constant1:
    block of instructions 1
    break;
  case constant2:
    block of instructions 2
    break;
  .
  .
  .
  default:
    default block of instructions
}

```

مخطط توضيحي للتعبير عن مجرى تنفيذ البرنامج للجملة switch:



## شرح كيفية استخدام الجملة switch:

إن استخدام الجملة switch ليس في الأمر المعقد و لا حتى بالصعب ، بل على العكس تماماً ، إن تعلم كيفية استخدام هذه الجملة سوف يعطيك القدرة على تنظيم و ترتيب برامجك بشكل أكثر تنسيقاً.

و حتى لا أطيل عليك في الشرح النظري، تعال لنأخذ مثال و من ثم نقوم في شرح كيفية عمل هذه البنية .

### مثال:

تعال لنأخذ المثال السابق الذي قمنا من خلاله من تحويل من الإنش إلى سم و بالعكس ، وقد قمنا في كتابة هذا البرنامج عن طريق if المتعددة الاتجاهات و حتى أثبت لك أن switch لا تختلف كثيراً عن if المتعددة الاتجاهات ، سوف نكتب البرنامج السابق نفسه لكن عن طريق switch .

## مثال عن الجملة switch

```
# include <iostream.h>
int main()
{
    float fac=2.54;
    float x,in,cm;
    char ch=0;
    cout<<"enter the length:\n";
    cin>>x;
    cin>>ch;
    switch(ch)
    {
        case 'i':
            in=x;
            cm=in*fac;
            break;

        case 'c':
            cm=x;
            in=cm/fac;
            break;

        default:
            in=cm=0;
            break;
    }
    cout<<in<<" in = "<<cm<<" cm "<<endl;
return 0;
}
```

**تنفيذ البرنامج:**

بعد كتابته و عملية التنفيذ سوف يطلب منا البرنامج بإدخال قيمة عددية ثم تحديد نوعها هل هيا سم أو إنش و ذلك عن طريق وضع الحرف i بعد القيمة المدخلة و الذي يرمز إلى وحدة القياس إنش، أو وضع الحرف c و الذي يرمز إلى وحدة القياس سم وسيظهر لديك على الشاشة على الشكل التالي:

```

C:\ "D:\book_C++\Debug\if.exe"
enter the length:
5i
5in=12.7cm

Press any key to continue_

```

و من ثم قم بالتحويل من سم إلى الإنش:

```

C:\ "D:\book_C++\Debug\if.exe"
enter the length:
3c
1.1811in=3cm

Press any key to continue

```

## شرح للأسطر البرنامج:

و كما تلاحظ معي بعد كتابة كود البرنامج و تنفيذه ، الشبه الكبير بين switch و if و أعتقد أنه ليس من الضروري شرح البرنامج من البداية ، وذلك لأننا قد قمنا في شرح نفس البداية في المثال السابق و سوف أكتفي في شرح البنية switch فقط إن شاء الله .

```
Switch ( ch )
```

```
{
case 'i':
    in=x;
    cm=in*fac;
break;

case 'c':
    cm=x;
    in=cm/fac;
break;

default:
    in=cm=0;
break;
}
```

بعد أن قام المستخدم بإدخال القيمة العددية x و نوع عملية التحويل من و إلى عن طريق إدخال نوع المتغير ch ، يأتي عمل الجملة switch و ذلك عن طريق مقارنة الحرف المدخل أوتوماتيكياً (مقارنة المتغير الموجود داخل القوسين اللذان يليان switch ) مع كل فرع من الفروع Case و عند حصول التطابق فإنه سوف يبدأ تنفيذ التعليمات الموجودة داخل هذا الفرع case فقط لا غير ، و يتم تجاهل جميع الفروع الأخرى المكتوبة ضمن الجملة switch ،

و عند الإنتهاء من تنفيذ جميع التعليمات الموجودة داخل القسم الذي قمنا في اختياره ، يأتي عمل التعليمة break و التي تعني الخروج من هذا الفرع case بشكل صحيح . و إذا لم يتوافق الحرف المدخل من قبل المستخدم أي تسمية من تسميات case حينئذ سوف يتم تنفيذ التعليمات الموجودة داخل القسم default .

و لنرجع إلى مثالنا لنشرح في التفصيل كيفية عمل برنامجنا في الحالة الأولى بعد أن قام المستخدم في إدخال القيمة العددية المراد تحويلها ه ، ومن ثم قام في إدخال الحرف i للتعريف بأن التحويل المراد هو من الإنش إلى سم قام switch بعرض الحرف المدخل على جميع فروعه حتى تتم عملية التطابق، و الحرف المدخل يطابق الفرع 'i' case حينئذ قام البرنامج في تطبيق التعليمات الموجودة داخل الفرع ، أي قام في عملية التحويل من الإنش إلى سم، و بعد الإنتهاء من عملية التحويل يأتي عمل التعليمة break للخروج من البرنامج بشكل صحيح .

و لو قام المستخدم في إدخال الحرف C بدلاً من الحرف I ، فإنه سيتم نفس مراحل العملية السابقة بالضبط و لكن سوف يتم التحويل من سم إلى الإنش .

و لاكن لو قام المستخدم في إدخال حرف لا يطابق أي تسمية من تسميات case ، فإنه سوف يتم تنفيذ التعليمات الموجودة داخل default و تخرج على الشاشة القيمة 0 كما مكتوب في برنامجنا.

و هذا هو كل برنامجنا و لا أعتقد بوجود شيء معقد.



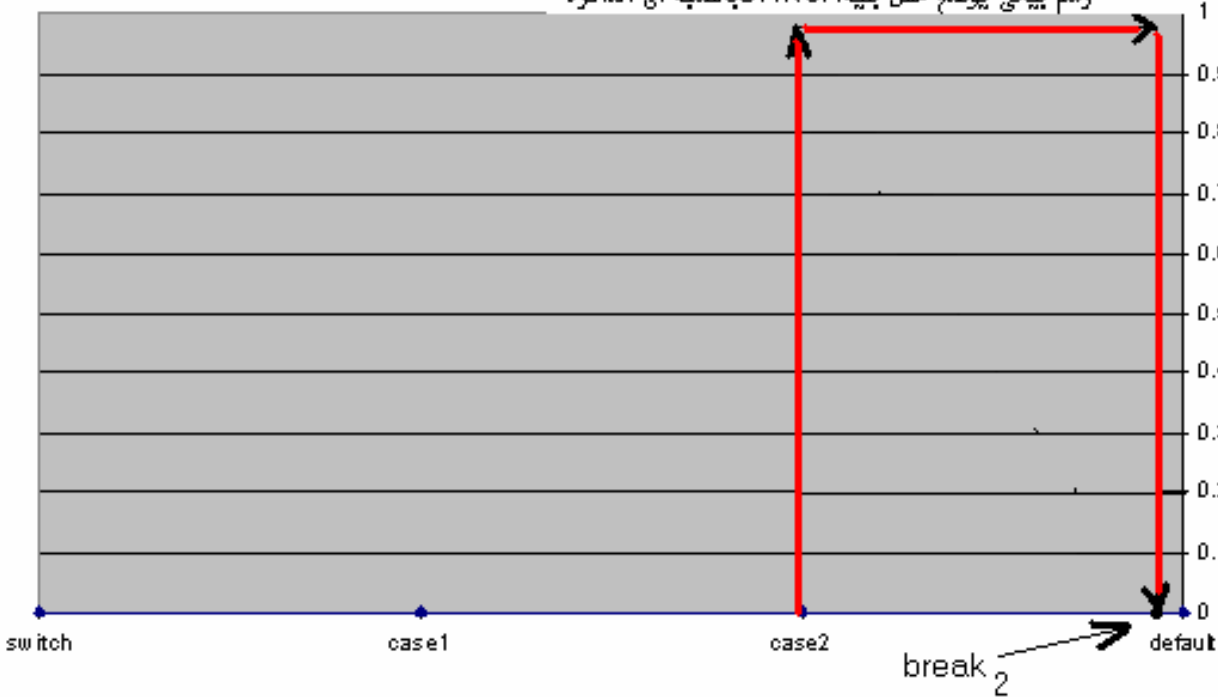
**توضيح :**

و حتى أوصل لك كيفية عمل switch سوف أعطيك مثال من الحياة حتى يقرب لك مفهوم عمل هذه الجملة

فرضاً لو كان عندك عدد كبير من العمال ، و أنت بحاجة إلى جميع هؤلاء العمال ولاكن ليس في نفس الوقت و في أن واحد ، فأنت مثلاً ليس بحاجة إلى عمال البناء و عمال النظافة في أن واحد ، فحتى يكون عمالك أكثر تنظيماً و أوفر مادياً أفضل طريقة هي تقسيم العمال إلى مجموعات كل واحد حسب المهنة و عمله ، فمثلاً تقسم عمالك إلى مجموعة عمال للبناء و مجموعة عمال للنظافة و ..... الخ و عند الحاجة إلى عمل معين ، فما عليك هو إلا استدعاء المجموعة المناسبة للعمل الذي تريده ، فهذا سوف يجعل عمالك أكثر تنظيماً و أكثر توفيراً للوقت و المال،

و كذلك بالنسبة إلى switch فأنت تقسم برنامجك إلى مجموعات ثم تقوم في استدعاء المجموعة التي تحددها أنت و المناسبة لك ، مما يجعل برامجك أقل استخداماً للذاكرة من غيرها ، لأنك تستدعي جزء واحد من البرنامج الكل ، و هذا الجزء المستدعي هو فقط سوى يشغل مكان في الذاكرة فقط ، و ليس البرنامج كله و هذا مخطط بياني يوضح كيفية عمل switch بالنسبة إلى الذاكرة ، و هذا المخطط هو فقط للمثال السابق فقط.

رسم بياني يوضح عمل بنية switch بالنسبة الى الذاكرة

**ملاحظة:**

الجملة switch تختلف عن if و حتى عن غيرها من الجمل الأخرى في أن switch لا تستخدم الأقواس الكبيرة من أجل الإحاطة في كل فرع case من فروعها على العكس من غيرها.

**مثال آلة حاسبة :**

في هذا المثال سوف نعرض كيفية إنشاء آلة حاسبة عن طريق switch حتى يترسخ لديك مفهوم هذه البنية، و في هذا المثال سوف نعرض العمليات الحسابية الأربعة (الجمع ، الطرح ، الضرب ، القسمة) في البداية سنكتب كود البرنامج و من ثم ننفذ البرنامج و من بعد ذلك سنقوم في شرح الجديد في هذا الكود .

```

#include <iostream.h>
int main()
{
float number1,number2,result;
char Operator;
cout<<" *** this is program calculeatar in C++ *** \n";
cout<<"enter the first number:\n";
cin>>number1;
cout<<"enter the Operator:\n";
cin>>Operator;
cout<<"enter the tow number:\n";
cin>>number2;

switch(Operator)
{
case ' + ' :
    result= number1+number2;
    break ;
case ' - ' :
    result= number1-number2;
    break ;
case ' * ' :
    result= number1*number2;
    break ;
case ' / ' :
    result= number1/number2;
    break ;
default :
    cout<<" bad Operator? \n";
}
cout<<"\n"<<number1<<" " <<Operator<<"
"<<number2<<" = "<<result<<endl;

return 0;
}

```

و بعد كتابة كود البرنامج نقوم في التحقق من عدم وجود أخطاء  
 ثم ننفذ البرنامج ويظهر لديك شاشة سوداء تطلب منك إدخال الرقم الأول و أنا هنا قمت في  
 إدخال الرقم ٢ ، وبعد إدخال الرقم الأول يطلب منك إدخال نوع العملية الحسابية (جمع ،  
 طرح ، ضرب ، قسمة) ، و أنا أدخلت الجمع  
 و من ثم يطلب منك إدخال الرقم الثاني ، و أنا أدخلت الرقم ٣ ، و من ثم يجب أن يخرج  
 على الشاشة العدد ٥ ناتج عملية الجمع.

```

C:\Documents and Settings\Administrator\ملف\ملف\C\Debu
*** this is program calculeatar in C++ ***
enter the first number:
2
enter the Operator:
+
enter the tow number:
3

2 + 3 = 5
Press any key to continue_
  
```

و هذه عملية الطرح :

```

C:\Documents and Settings\Administrator\ملف\ملف\C\Debu
*** this is program calculeatar in C++ ***
enter the first number:
10
enter the Operator:
-
enter the tow number:
3

10 - 3 = 7
Press any key to continue
  
```



## شرح للأسطر البرنامج :

في الحقيقة لا أرى في البرنامج الشيء الكثير للشرح ، فإن معظم الأوامر و الأسطر في رأي أنها ليت جديدة عليكم ، و لكنني أحب أن أنبه إلى أن المتغير الذي يدل على العملية الحسابية Operator الحرف الأول بدأته في حرف كبير و ذلك لأن هذا الاسم إذا بدأ في حرف صغير فإنه سيدل على إحدى الكلمات المحجوزة بذلك بدأته في حرف كبير .  
أما بقية الأسطر فهي ليست بذلك التعقيد و الصعوبة ،

**وظيفية منزلية :** أضف بعض العمليات الحسابية الأخرى مثل مربع العدد ، و الجذر ، و حاول أن تعمل العمليات الحسابية على ثلاثة أعداد .

## نصيحة إلى الأخوة المبتدئين في عالم البرمجة :

إذا أردت أخي المبتدئ أن تصبح من المبدعين في البرمجة و ليس في هذه اللغة فقط فعليك بالخيال الواسع و الأفكار الجديدة دوماً و أبداً ، و أن تفكر لكل شيء يصادفك في حياتك أن تحوله إلى شكل برنامج ، و على سبيل المثال الراتب اليومي أو الأسبوعي الذي تستلمه من والدك هو فكرة جيدة لكتابة برنامج بحيث هذا البرنامج يقوم في حساب المال الذي تأخذه من والدك و ما هو الحد المسموح به للمصروف اليومي و إذا صرفت مبلغ معين من المال كل يوم كم سيبقى معك في نهاية الشهر ..... الخ  
و هذا ما يسمى في عالم البرمجة بالخوارزميات و في الحقيقة هناك مبرمجون مختصون فقط في هندسة الخوارزميات الذين يقومون بعملية التحويل من بعض جوانب الحياة البسيطة التي نعيشها إلى كود برنامج و لكن على شكل نظري أي خوارزمي ، و كل فريق برمجي لا بد أن يحتوي على الأقل على مهندس واحد للخوارزميات ، و بعد أن توضع الخوارزمية المناسبة للبرنامج ، يقوم بعد ذلك بقية الفريق بكتابة كود البرنامج لذلك فإن التفكير المنطقي و الخيالي في أن واحد في البرمجة هما طريقك للتفوق .

## مثال طلب دخول لجامعة الهندسة :

و لنعود مرة أخرى إلى برنامجنا و الذي هو عبارة عن تعبئة طلب لدخول جامعة الهندسة ، و في هذه المرة سوف تواجهنا مشكلة ألا و هيا أن في هذه الجامعة لا يوجد هندسة كمبيوتر و فقط ! بل يوجد العديد من الكليات ضمن هذه الجامعة و سوف نأخذ في مثالنا أنه يوجد لدينا في هذه الجامعة كلياتان فرضاً :

١ - كلية للمعلوماتية (هندسة الكمبيوتر) .

٢ - كلية لهندسة الميكانيك .

٣ - الخروج من القائمة .

و أنا قلت فرضاً و لكنه في الحقيقة يوجد العديد من الكليات داخل الجامعة الواحدة و لكني إخترت هنا كلياتان فقط وذلك من أجل التقليل من عدد أسطر البرنامج و من أجل فهم الكود بسهولة .

و سوف نحتاج في كتابة الكود المناسب للبرنامج إلى قائمة توفر لنا الاختيار إلى أي كلية أريد الانتساب ، و من ثم أدخل إلى هذه الكلية من أجل تعبئة الطلب إن كان يحق لي الانتساب إلى هذه الكلية حسب شروط الانتساب .

و بالطبع نحن بحاجة إلى استخدام الجملة switch التي توفر لنا صنع القوائم ، و سوف نضع كل كلية على شكل case ضمن البنية switch ، و داخل الفرع case نكتب شروط الانتساب إلى هذه الكلية ، لن أطيل عليك في الشرح هذا هو كود البرنامج بعد كتابته و سوف أقوم في شرح للأسطر البرنامج الجديدة بعد مرحلة التنفيذ .

```

#include <iostream.h>
main()
{
    int n;
    cout<<"\n\n";
    cout<<"\t\t ***** \n";
    cout<<"\t\t *** Welcome to University *** \n";
    cout<<"\t\t ***** \n\n";
    cout<<" \t 1- to enter Faculty Of Information Technology Computer .\n";
    cout<<" \t 2- to enter Faculty Machine .\n";
    cout<<" \t 3- to EXIT .\n";
    start_menu:
    cin>>n;
switch (n)
{
    case 1:
        int grade;
        float grmath;
            cout<<"\t\t *****\n";
            cout<<"\t\t *** Faculty Computer ***\n";
            cout<<"\t\t *****\n";
        cout<<" enter the grade: \n ";
        cin>>grade;

        if(grade>=65)
        {
            cout<<"welcome to University \n";
        }
        else
        {
            if(grade<50)
            {
                cout<<"Excuse! your grade less than 50!\n";
                return 0;
            }
            cout<<"excuse your grade less than 70!:\n";
            cout<<"But how your grade int the mathematic!:\n";
            cout<<"enter the grade mathmatic:\n";
            cin>>grmath;

            if(grmath>=70 && grade>=50)
            {
                cout<<"welcome to University\n";
            }

            else
            {
                cout<<"Excuse! your grade less than 70! \n";
                cout<<"and excuse your grade mathmatic less than 80\n";
            }
        }
        break;
}
}

```



```

case 2:
    cout<<"\t\t\t *****\n";
    cout<<"\t\t\t *** Faculty machin ***\n";
    cout<<"\t\t\t *****\n";
    int grade_machine;
    float grphysic_machine;

    cout<<" enter the grade: \n ";
    cin>>grade_machine;

    if(grade_machine>=65)
    {
        cout<<"welcome to University \n";
    }
    else
    {
        if(grade_machine<50)
        {
            cout<<"Ecxuse! your grade less than 50!\n";
            return 0;
        }
        cout<<"excuse your grade less than 65!:\n";
        cout<<"But how your grade int the phisic!:\n";

        cout<<"enter the grade phisic:\n";
        cin>>grphysic_machine;

        if(grphysic_machine>=70 && grade_machine>=50)
        {
            cout<<"welcome to University\n";
        }

        else
        {
            cout<<"Ecxuse! your grade less than 65! \n";
            cout<<"and ecxuse your grade grphysic less than 70\n";
        }
    }
    break;

case 3:
    return 0;

    break;

default:
    cout<<"erorr? enter agin\n\n";
    goto start_menu;
    break;
}

return 0;
}

```

## ما هو الجديد في هذا البرنامج :

قبل أن أبدأ في شرح للأسطر البرنامج سوف أعطيك النقاط التي أضفتها إلى هذا البرنامج

- ١- كلية المعلوماتية : و هي لا تختلف أبداً عن المثال السابق و لم أضف أي شيء سوى أنني و ضعتها داخل فرع case1 من الجملة switch .
- ٢- كلية هندسة الميكانيك : و هي على نفس نمط كلية المعلوماتية إلا أن شروط الانتساب تختلف قليلاً في علامات القبول (درجة علامة الطالب) شروط الانتساب إلى كلية هندسة الميكانيك :
  - أ- درجة علامة الطالب العامة لا تقل عن ٦٥ % .
  - ب- درجة علامة الطالب العامة لا تقل عن ٥٠ % و درجة علامة الفيزياء لا تقل عن ٧٠ % .
- ٣- قمت في إضافة الجملة goto .

## شرح للأسطر البرنامج :

```
int n ;
```

في هذا السطر قمنا في إعلان البرنامج عن المتغير n و الذي يرمز إلى حالة الإدخال من عناصر القائمة الموجودة لدينا و هو من النوع الأعداد الطبيعية int ، لأنه في هذه الحالة قمنا في تسمية الفروع case بأعداد طبيعية ١ و ٢ و ٣ .

أما الأسطر التالية :

```
cout<<" \t 1- to enter Faculty Of Information Technology
Computer .\n";
cout<<" \t 2- to enter Faculty Machine .\n";
cout<<" \t 3- to EXIT .\n";
```

فهي تعرض للطالب على الشاشة الاختيارات الموجودة في البرنامج .  
و السطر التالي :

```
start_menu:
```

و هذا السطر عبارة عن عنوان للجملة goto و سوف أشرح هذه الجملة فيما بعد .

و السطر التالي :

```
cin>>n;
```

فهو يستقبل من الطالب القيمة المختارة من القائمة و يرسلها إلى الجملة switch حتى يتم تنفيذ الفرع الموافق للاختيار الطالب .

أما في هذا السطر :

switch (n)

فإنه يتم اختبار القيمة المدخلة ، فإذا كانت القيمة المدخلة توافق إحدى تسميات case فإنه سوف يتم تنفيذ هذا الفرع و إلا سيتم تنفيذ default .

و بالنسبة إلى الفرع الأول :

case 1:

فقد وضعت بداخله البرنامج السابق شروط الانتساب إلى كلية الهندسة و لم أضف أي شيء جديد

و أما الفرع الثاني :

case 2:

فقد وضعت فيه شروط الانتساب إلى كلية هندسة الميكانيك ، و في الحقيقة لقد كتبت شروط الانتساب على نفس نمط شروط الانتساب إلى كلية المعلوماتية و لكن الفرق في هنا هو فقط في درجات علامة القبول للكلية و هي على الشكل المعروض في الصفحة ٤٨ .

أما الفرع الثالث :

case 3:

return 0;

فهو للخروج من البرنامج و ذلك عن طريق return 0; و هي تعني خروج من البرنامج بشكل صحيح .

أما السطر التالي :

default:

cout<<"erorr? enter agin\n\n";

فهو يعرض على الشاشة رسالة في حالة إدخال خاطئ تخبر الطالب بالخطأ و تعلنه بإدخال قيمة جديدة .

أما السطر الأخير :

```
goto start_menu;
```

فهو عبارة عن الجملة goto و هذه الجملة تدعى بأمر القفز و هي تساعدنا على العودة إلى نقطة معينة من البرنامج ومن ثم يبدأ البرنامج في التنفيذ من جديد من تلك النقطة و هذه النقطة أنت الذي تحدد اسمها و مكانها في البرنامج ، و بالنسبة إلى برنامجنا هنا فإن النقطة هي start\_menu و يجب أن تتبع في نقطة منقوطة (:).

### ملاحظة :

لا ينصح استخدام الجملة goto في أكواد البرامج الكبيرة و للأكثر من مرة ، و استخدامها في كود البرنامج بكثرة يؤدي إلى بطئ البرنامج أثناء استخدامها من قبل المستخدم و إلى ظهور بعض الأخطاء أحياناً ، و يحرص معظم المبرمجون على عدم استخدام هذه الجملة و ذلك من أجل أن تكون برامجهم ذاتية أي تعتمد على نفسها في كل شيء من دون تدخل المبرمج لكي يوجها و يقول للبرنامج إذهب إلى هنا أو إلى هناك .

## جمل تنفيذ الحلقات في C++:

### أ- الجملة for :

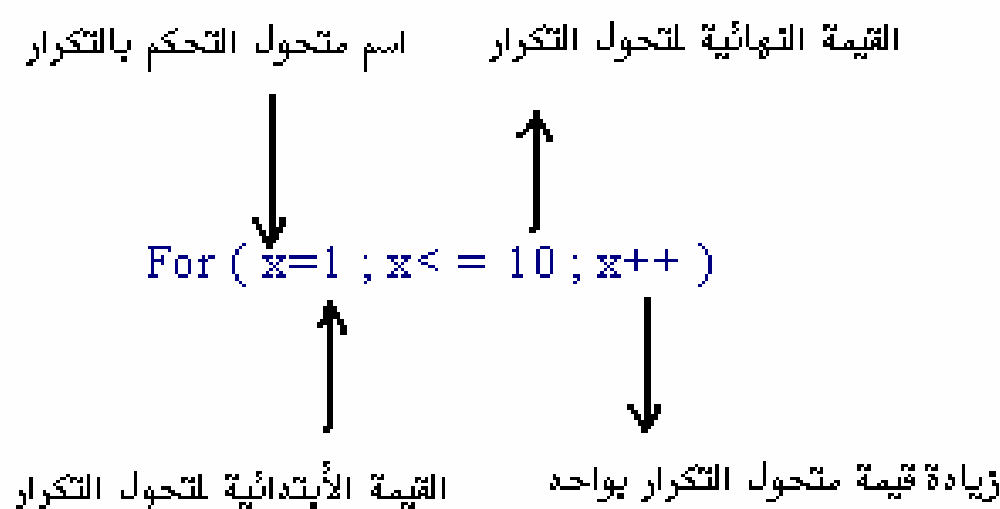
إن جملة التكرار for تضيف القوة إلى برامجك و ذلك بسبب بنيتها المتينة و المرنة و التي تعطيك المزيد من التحكم في صياغة برامجك و بالإضافة إلى توفير الوقت الكثير الذي توفرة عليك هذه الجملة .

و الشكل العام لجملة التكرار for يكون على الشكل التالي :

**For ( expression1 ; expression2 ; expression3 )  
Statment**

حيث يعبر expression1 عن قيمة متحول التحكم بالتكرار .  
و expression2 تعبر عن شرط الاستمرار بالتكرار .  
أما expression3 تقوم بزيادة قيمة متحول التحكم بالتكرار أو بالتناقص .

ولنأخذ مثال لتوضيح الأجزاء المولفة للبنية for التكرارية :



و حتى لا أطيل عليكم في الكلام النظري لنأخذ مثال فوراً لكي تستوعب مفهوم كيفية عمل هذه الجملة .

### مثال: جدول الضرب

إخراج على الشاشة جدول الضرب من ١ إلى ١٠

في الحقيقة لو أردت كتابة هذا المثال البسيط من دون for فإن كتابة كود البرنامج سيأخذ منك وقت و جهد طويل بالإضافة إلى زيادة أسطر البرنامج والذي قد يؤدي بك إلى الوقوع في الكثير من الأخطاء ، أما الآن فلنكتب هذا البرنامج من دون الجملة for و من ثم سوف نعيد كتابة نفس البرنامج و لكن من خلال for و من ثم نلاحظ الفرق بينهما.

و هكذا سوف يكون كود البرنامج بدون جملة التكرار for :

```
# include <iostream.h>
int main()
{
int x1=1,y1=1,z1;
z1=x1*y1;
cout<<x1<<"*"<<y1<<" = "<<z1<<endl;
int x2=1,y2=2,z2;
z2=x2*y2;
cout<<x2<<"*"<<y2<<" = "<<z2<<endl;
.....أكمل أنت بقية هذا الكود.....
.....
int x10,y10,z10;
z10=x10*y10;
cout<<x10<<"*"<<y10<<" = "<<z10<<endl;

return 0;
}
```

و هذا كود البرنامج نفسه ولكن من خلال جملة التكرار for .

```
# include <iostream.h>
int main()
{
int x,y,z;
for(x=1;x<=10;x++)
{
for(y=1;y<=10;y++)
{
z=x*y;

cout<<x<<"*"<<y<<" = "<<z<<endl;
}
}
return 0;
}
```

و طبعاً أنت تلاحظ الفرق الكبير بين الطريقتين من حيث التنظيم و اختصار عدد أسطر البرنامج ، و نحن في هذا البرنامج قمنا في كتابة برنامج يطبع على الشاشة جدول الضرب من ١ إلى ١٠ فقط ، تخيل مثلاً لو طلب منا برنامج يخرج على الشاشة جدول الضرب من ١ إلى ١٠٠٠ ، فكم سيتطلب منك من الوقت و العبء الكبيران من أجل كتابة كود هذا البرنامج ، و لكن مع جملة التكرار for فأنت بحاجة فقط إلى نفس أسطر البرنامج السابق ، فقط قم بتغيير القيم التالية :

- ١- قيمة متحول التحكم الابتدائي .
- ٢- قيمة شرط استمرار التكرار .



## شرح للأسطر البرنامج :

في الحقيقة لا أعتقد أنه يوجد أي صعوبة في أسطر هذا البرنامج و الجديد هنا هو جملة التكرار for فقط لا غير،

بعد أن أعلننا عن المتغيرات الثلاثة x, y, z و أنواعها (int)، بدأنا في كتابة الجملة for

```
For ( x = 1 ; x <= 10 ; x++ )
```

في هذه البنية قمنا أولاً في إعطاء متحول التحكم x بالقيمة الابتدائية العدد ١ و الفاصلة المنقوطة بعده تعلن عن نهاية قيمة متحول التحكم بالتكرار الابتدائي ثم أعلننا بعد ذلك عن شرط الاستمرار بالتكرار و الذي هو  $x \leq 10$  و الفاصلة المنقوطة بعده لها نفس وظيفة الفاصلة المنقوطة الأولى و من ثم أعلننا عن كيفية زيادة قيمة متحول التحكم بالتكرار الابتدائي و هنا قمنا في زيادة متحول التكرار في واحد كل مرة أما بالنسبة إلى البنية for الثانية

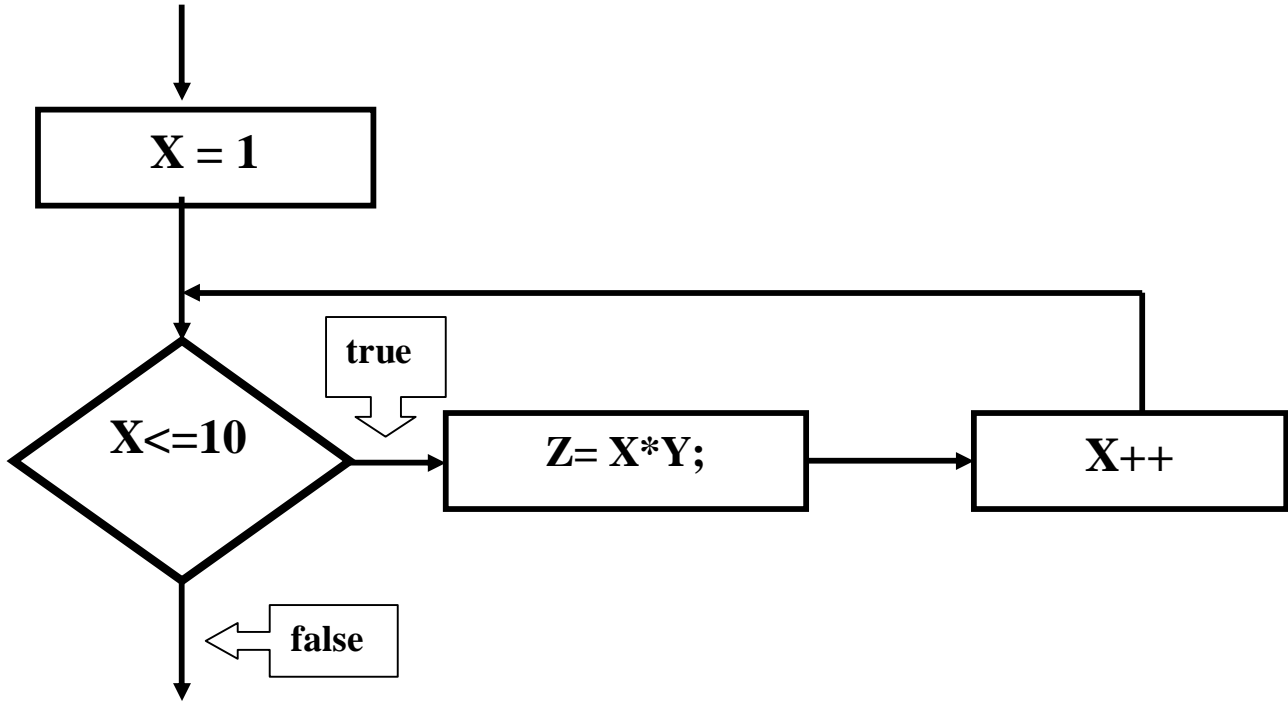
```
For ( y = 1 ; y <= 10 ; y++ )
```

فهي تماماً لها نفس مبدأ عمل و تنظيم for الأولى التي تم شرحها قبل قليل و لا يوجد أي اختلاف سوى في اسم المتغير y فقط لا غير .

و الأقواس الكبيرة { } بعد الجملة for الأولى و الثانية هي لتحصر ما يجب تنفيذه ما زال شرط الاستمرار بالتكرار محقق ، و هنا هو إخراج على الشاشة قيمة z و التي هي كما موضحة في البرنامج عملها هو إيجاد ناتج ضرب العدد الأول من الجملة for الأولى مضروب في العدد الأول من الجملة for الثانية وهكذا إلى العدد ١٠

```
z = x * y ;  
cout << x << " * " << y << " = " << z << endl ;
```

و حتى يتم لك مفهوم عمل الجملة for التكرارية سوف أقوم في شرح مبدأ عملها من خلال مخطط توضيحي للبنية for



مخطط توضيحي لبنية عمل for

تمعن جيداً في المخطط الصندوقي للجملة for و لاحظ كيف يتم تنفيذ هذه الجملة يتم تنفيذ الجملة for على الشكل التالي :

عندما يبدأ المترجم في تنفيذ الجملة for يبدأ في متحول التحكم الابتدائي حيث أن متحول التحكم الابتدائي يأخذ قيمته الابتدائية و التي تساوي هنا العدد  $x = 1$  ، بعد أن يتم إسناد العدد 1 إلى متحول التحكم الابتدائي ، يتم التحقق من صحة شرط الاستمرار بالتكرار

و الذي هو هنا أصغر أو يساوي العدد  $x \leq 10$  ، و العدد 1 يدخل ضمن المجال أصغر أو يساوي من العدد 10 أي أن الشرط محققاً true ،

و بعد التحقق من صحة شرط الاستمرار يتم تنفيذ الأوامر و التعليمات الموجودة داخل الأقواس الكبيرة للجملة for مرة واحدة فقط ، و في برنامجنا فإنه يتم تنفيذ ضرب العدد ١ من الجملة for الأولى و العدد ١ من الجملة for الثانية وإخراج الناتج على الشاشة .

و بعد أن يتم الانتهاء من تنفيذ التعليمات الموجودة ضمن for ، بعدها يتم زيادة قيمة المتحول الابتدائي x بواحد و ذلك من خلال التعليمة x++ ، أي أن قيمة x تصبح تساوي العدد ٢ ، و بعد الزيادة بواحد تبدأ الجملة for من جديد في عملية التحقق من صحة شرط الاستمرار بالتكرار بالنسبة إلى القيمة الجديدة للمتحول x .

و العدد ٢ يدخل ضمن المجال أصغر أو يساوي ١٠ ، أي أن شرط الاستمرار محقق true ..... وهكذا يتم زيادة قيمة المتحول بالتكرار x كل مرة بواحد و من ثم التحقق من صحة الاستمرار بالتكرار و من ثم التنفيذ حتى تصبح قيمة المتحول تساوي القيمة ١١ مما يسبب في عدم صحة شرط الاستمرار بالتكرار false مما يؤدي إلى إنهاء عمل for

ليتابع بعدها البرنامج في تنفيذ التعليمات التالية في البرنامج ، و في برنامجنا يقوم البرنامج في تنفيذ التعليمة ; return 0 أي الخروج من البرنامج .

و أرجوا من الله عز و جل أن أكون قد أوصلت لك مفهوم عمل الجملة for إن كنت قد أحسنت في الشرح فهذا من فضل الله و لله الحمد، و إن لم أوفق فهذا من ظلمي لنفسي، نسأل الله الهداية و الرحمة، و لا تتسوني من دعوة أخ في الله .

## تنبيه :

من الأخطاء التي يقع بها بعض المبرمجين هي وضع فاصلة منقوطة بعد أقواس الجملة for

و في الحقيقة إن وضع الفاصلة المنقوطة بعد أقواس for لا يظهر أي خطأ أثناء مرحلة ترجمة الكومبليتر لكود البرنامج ( أثناء مرحلة الترجمة ) ، ولكن الخطأ يكمن في أن جسم الجملة for يصبح مؤلفاً من تعليمة فارغة ، أي و كأن الجملة for غير موجودة ، جرب في نفسك و ضع بعد التعليمة for فاصلة منقوطة في المثال السابق و لاحظ ماذا سوف يخرج لك البرنامج على الشاشة .

## ملاحظة :

إن كل تعبير من التعبيرات الثلاثة في جملة التكرار for هي تعبير اختيارية و ليست إجبارية ،

أي وضع اسم التعبير الأول فقط من دون قيمة ابتدائية هو اختياري ، و لكن لو وضعت في المثال السابق اسم المتحول بالتكرار بدون قيمة ابتدائية سينتج لدينا حلقة غير منتهية وذلك لأن المترجم يفترض قيمة التعبير true . و لو قمنا في كتابة التعبير الثاني من دون وضع القيمة النهائية لمتحول التكرار ، سوف ينتج لدينا نفس النتيجة ، وكذلك الأمر بالنسبة إلى التعبير الثالث .

## ملاحظة :

يُحِبُّ وضع عملية الإعلان عن المتغيرات ضمن الجملة for لسهولة قرأت البرنامج فيما بعد ، و مما يفضله بعض المبرمجين هو تعريف المتغيرات المستخدمة في جملة التكرار for من النوع int .

مثال:

```
for (int x=0;x<=10;x++)
```

## عمليات الإسناد في C++

تدعم لغة C++ عمليات الإسناد معظم العمليات وليس كلها ، دعنا في البداية نتعرف على عملية الإسناد .

عملية الإسناد هي في الحقيقة عملية اختصار للعمليات الحسابية البسيطة المعروفة و نقصد في اختصار العمليات الحسابية هو الجمع بين عمليتين حسابيتين في عملية حسابية واحدة و ذلك من أجل كتابة برامجنا بقوى و مرونة أكبر .

فعلى سبيل المثال يمكن اختصار العملية الحسابية التالية

$$x = x + 5 ;$$

من خلال عملية الإسناد على الشكل التالي:

$$x += 5 ;$$

و بالطبع سوف تكون النتيجة واحدة لا محالة ، و لكن كيف سوف تكون النتيجة واحدة مع الإختلاف الواضح في الطريقتين ،

يقوم مترجم لغة C++ في قرأت عملية الإسناد += على الشكل التالي:

جمع قيمة المتغير x الموجود على يمين المعادلة مع العدد الموجود بعد قيمة الإسناد ثم إسناد القيمة الناتجة إلى قيمة المتغير x .

مثال للتوضيح:

```
# include<iostream.h>
int main()
{
    int x=1;
    cout<<"x = "<<x<<endl;
    x+=3;
    cout<<"(x+=3) = "<<x<<endl;
    x+=3;
    cout<<"(x+=3) = "<<x<<endl;

    return 0;
}
```

و النتيجة كالتالي:

```

C:\ "C:\DOCUMENTS AND SETTINGS\
x = 1
(x+=3) = 4
(x+=3) = 7
Press any key to continue_

```

و هذه بعض العمليات الحسابية التي يمكن إسنادها و كتابتها على شكل عملية إسنادية مع الطريقة الرياضية التي تكتب فيها في الشكل الطبيعي

عملية الإسناد في C++	الشكل الرياضي
$x+=5;$	$x=x+5$
$x-=7;$	$x=x-7$
$x*=x;$	$x=x*2$
$x/=3;$	$x=x/3$
$x\%=4;$	$x=x\%4$

\*\*\*\*\*

اللهم لا تنسني من رحمتك

\*\*\*\*\*

## عمليات الزيادة و الإنقاص في C++ :

تدعم لغة C++ عمليات الزيادة و الإنقاص بشتى أشكالها و الشكل الأساسي للزيادة أو الإنقاص هو الزيادة بواحد (++) و عملية الإنقاص بواحد (-- ) ، و هما عمليتان متعاكستان و لكن لهما نفس مبدأ العمل و الفرق هو أن الأولى تبدأ بتنفيذ البرنامج من الأصغر إلى الأكبر أما عملية الإنقاص بواحد فإنها تبدأ في تنفيذ البرنامج من الأكبر إلى الأصغر .

**مثال لعملية الزيادة بواحد :**

إخراج على الشاشة من العدد 0 إلى العدد 10 بشكل تصاعدي.

يخرج لدينا على التالي:

```
# include<iostream.h>
int main()
{
    for(int x=0; x<=10;x++)
        cout<<x<<" ";
return 0;
}
```

وبعد عملية التنفيذ سوف الشاشة على الشكل

C:\> "C:\Documents and Settings\Administrator\..."  
0 1 2 3 4 5 6 7 8 9 10

**مثال على عملية الإنقاص بواحد :**

إخراج على الشاشة من العدد 0 إلى العدد 10 على شكل تنازلي.

```
# include<iostream.h>
int main()
{
    for(int x=10; x>=0;x--)
        cout<<x<<" ";
return 0;
}
```



وبعد عملية التنفيذ سوف يخرج لدينا على الشاشة على الشكل التالي:

```
C:\Documents and Settings\Administrator\
10 9 8 7 6 5 4 3 2 1 0
```

لاحظ الفرق بين جملة التكرار في عملية الزيادة و عملية الإنقاص بواحد من خلال جملة التكرار `for` هو أنهما عمليتان متعاكستان من حيث كود البرنامج و الخرج على الشاشة .

و يوجد نوعين لعملية الزيادة بواحد و كذلك الأمر بالنسبة لعملية الإنقاص بواحد و هما:

١ - **عملية الزيادة بواحد (++) :**

أ- عملية الزيادة بواحد الأمامية .

ب- عملية الزيادة بواحد الخلفية .

٢ - **عملية الإنقاص بواحد (--):**

أ- عملية الإنقاص بواحد الأمامية .

ب- عملية الإنقاص بواحد الخلفية .

الفرق بين عملية الزيادة (أو الإنقاص) بواحد الأمامية و الخلفية هو متى يتم استخدام قيمة المتحول المراد زيادته (أو نقصانه) بعد أم قبل الزيادة على المتحول. ففي عملية الزيادة (النقصان) بواحد الأمامية يتم أولاً زيادة قيمة المتحول ثم بعد الزيادة يتم استخدام القيمة الجديدة.

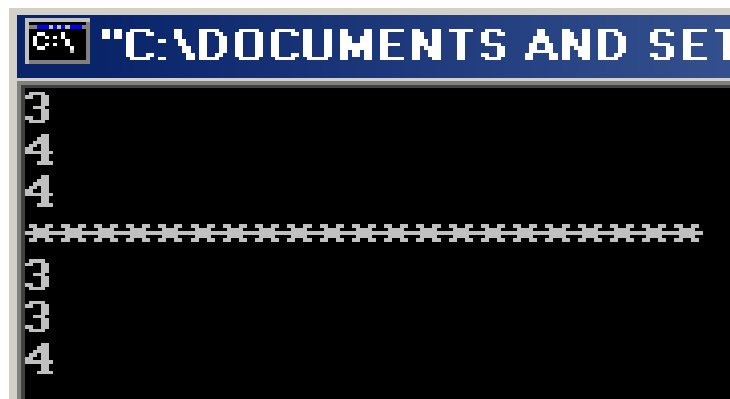
أما في عملية الزيادة (النقصان) بواحد الخلفية يتم أولاً استخدام القيمة الأساسية المعطاة للمتحول ثم بعد ذلك تتم عملية الزيادة بواحد للمتحول.

و هذا مثال يوضح مفهوم استخدام كل واحدة من العمليات السابقة

```
include<iostream.h>
t main()

int x=3;
cout<<x<<endl; // إخراج على الشاشة القيمة الأصلية
cout<<++x<<endl; // عملية الزيادة بواحد للأمام
cout<<x<<endl; // قيمة المتحول بعد عملية الزيادة بواحد للأمام
cout<<"*****"<<endl;
x=3;
cout<<x<<endl; // إخراج على الشاشة القيمة الأصلية
cout<<x++<<endl; // عملية الزيادة بواحد للخلف
cout<<x<<endl<<endl; // قيمة المتحول بعد عملية الزيادة بواحد للخلف
return 0;
```

و الخرج على الشاشة سيكون على الشكل التالي:



```
"C:\DOCUMENTS AND SET"
3
4
4
*****
3
3
4
```

## تنبيه:

يؤدي استخدام عملية الزيادة أو النقصان في أكثر من واحد ضمن جملة التكرار `for` إلى نشوء حلقة غير منتهية مع العلم أن المترجم لا يخرج أي خطأ.  
مثال خطأ:

```
for(int x=0 ; x<=10 ; x+2)
```

## أمثلة حول جملة التكرار `for` :

١- أكتب برنامج حيث يخرج على الشاشة الأعداد الزوجية فقط من ٠ إلى ٦٠ بشكل تصاعدي وذلك من خلال الجملة `for` التكرارية.

الحل:

```
# include<iostream.h>
int main()
{
    for(int x=0;x<=60;x+=2)
        cout<<"\tx="<<x;
    cout<<endl;
    return 0;
}
```

"C:\Documents and Settings\Administrator\مكتبة\program\_C\Debug\program\_C.exe"

x=0	x=2	x=4	x=6	x=8	x=10	x=12	x=14	x=16
x=18	x=20	x=22	x=24	x=26	x=28	x=30	x=32	x=34
x=38	x=40	x=42	x=44	x=46	x=48	x=50	x=52	x=54
x=58	x=60							

Press any key to continue

٢- أكتب برنامج حيث يخرج على الشاشة الأعداد الفردية فقط من ١ إلى ٩٩ بشكل تنازلي وذلك من خلال الجملة `for` التكرارية.

الحل:

```
# include<iostream.h>
int main()
{
    for(int x=99;x>=1;x-=2)
        cout<<"\tx="<<x;
    cout<<endl;
    return 0;
}
```

"C:\Documents and Settings\Administrator\... \namer\_C\Debug\namer\_C.exe"

```
x=99    x=97    x=95    x=93    x=91    x=89    x=87    x=85    x=
x=81    x=79    x=77    x=75    x=73    x=71    x=69    x=67    x=65    x=
x=61    x=59    x=57    x=55    x=53    x=51    x=49    x=47    x=45    x=
x=41    x=39    x=37    x=35    x=33    x=31    x=29    x=27    x=25    x=
x=21    x=19    x=17    x=15    x=13    x=11    x=9     x=7     x=5     x=
Press any key to continue_
```

٣- أكتب برنامج حيث يخرج على الشاشة الأعداد من ٠ إلى ٥٥ بخطوة قدرها ٥ و بشكل تصاعدي.

الحل:

```
# include<iostream.h>
int main()
{
    for(int x=0;x<=55;x+=5)
        cout<<"\tx="<<x;
    cout<<endl;
    return 0;
}
```

"C:\Documents and Settings\Administrator\... \namer\_C\Debug\namer\_C.exe"

```
x=0     x=5     x=10    x=15    x=20    x=25    x=30    x=35    x=
x=40    x=45    x=50    x=55
Press any key to continue_
```

- ٤ - تاجر أعلن عن فتح محله التجاري الجديد و لكي يجذب عدد زبائن أكثر قرر عمل دعاية لمحلة التجاري و ذلك من خلال إعطاء لكل زائر يدخل المحل و يكون رقمه الثلاثون ٣٠ إعطاؤه هدية معينة.
- أكتب برنامج لذلك التاجر بحيث عندما يدخل الزائر رقم ٣٠ يخرج على الشاشة أنه الزائر الرابح الذي سوف يمنح الهدية ، و لنفرض أن عدد الزوار في اليوم مئة زائر فرضاً.
- الحل:

```

#include<iostream.h>
int main()
{
    for(int x=0;x<=100;x++)
    {
        if(x%30==0)
        {
            cout<<endl;
            cout<<"\t\t\t*** this is win ="<<x<<" ***\n"<<endl;
        }
        cout<<"\tx="<<x;
    }
    cout<<endl;
    return 0;
}

```



**الحل:**

قبل أن نبدأ في كتابة كود البرنامج المطلوب منا يجب أن نحاول كتابة خوارزمية هذا البرنامج لكي يسهل علينا مفهوم المطلوب منا .

عمل أيام الأسبوع ما عدا الخميس و الجمعة يستلم الموظف راتب قدره n1

عمل يوم الخميس يستلم الموظف راتب قدرة n2

يصرف الموظف في يوم الجمعة مبلغ قدرة n3

يوجد لدى الموظف في البنك مبلغ معين قدرة n4

عدد الأيام الكلي لعمل الموظف n5

و من ثم نكتب أسماء المتغيرات لكل متغير سوف نستخدمه في البرنامج:

١- راتب اليوم: rateb\_day

٢- راتب يوم الخميس: rateb\_alkamis

٣- المال المتوفرة لدى الموظف في البنك : money\_bank

٤- مصروف يوم الجمعة : saruf\_aljum3a

٥- عدد الأيام الكلي لعمل الموظف : days

٦- مجموع المال الكلي : full\_money

٧- متغير لكي يحدد أي يوم من أيام الأسبوع : s

و بعد ذلك نقوم في كتابة كود البرنامج إذا كانت فكرة البرنامج قد اتضحت لك ، و إن لم تكن فكرة البرنامج غير واضحة فأرجوا قراءة المثال مرة أخرى حتى تتضح لك فكرة البرنامج لكي تستوعب كود البرنامج.

## كود البرنامج:

```

#include<iostream.h>
int main()
{
int rateb_day,rateb_alkamis,money_bank,saruf_aljum3a,full_money,days;

cout<<"enter the rateb day : ";
cin>>rateb_day;
cout<<"enter the rateb alkamis: ";
cin>>rateb_alkamis;
cout<<"enter masruf aljum3a: ";
cin>>saruf_aljum3a;
cout<<"enter the money in the Bank:";
cin>>money_bank;
cout<<"enter how days: ";
cin>>days;

full_money=money_bank;
for(int s=1;s<=days;s++)
{
if(s%6==0)
{
full_money+=rateb_alkamis;
}

else
{
if(s%7==0)
{
full_money-=saruf_aljum3a;
}
else
{
full_money+=rateb_day;
}
}
}
cout<<"\t\t *****\t\t\n";
cout<<"\t\t *****\t\t\n";
cout<<"\t\t *****\t\t\n\n";
cout<<"this is your money : "<<full_money <<" *** since days: "<<days<<endl;
cout<<endl;
return 0;
}

```



## مرحلة تنفيذ البرنامج :

بعد كتابة كود البرنامج و التأكد من خلو البرنامج من أي خطأ نضغط على `ctrl + f5` سوف يخرج لدينا شاشة سوداء تطلب منا على الشكل التالي:

١ - المرتب اليومي للموظف

enter the rateb day :

و أنا هنا قمت فرضاً قمت في إدخال مرتب يومي قدرة ١٠

٢ - مرتب يوم الخميس إذا كان الموظف قد عمل ذلك اليوم

enter the rateb alkamis:

قمت في إدخال القيمة ١٥

٣ - إدخال مصروف يوم الجمعة

enter masruf aljum3a:

أدخلت ٢٥

٤ - إدخال المبلغ الموجود في البنك لدى الموظف

enter the money in the Bank:

أدخلت مبلغ قدرة ٥٠٠

٥ - و نهاية عدد الأيام التي قام الموظف في عملها

enter how days:

و هنا أدخلت فقط سبعة أيام فقط ( أسبوع ) و ذلك للتوضيح مع العلم أنك تستطيع عدد

الأيام التي تريد شهر أو أكثر أو أقل من ذلك.

و نهاية سوف يخرج لك البرنامج مبلغ المال الكلي المتوفر لديك خلال عدد الأيام التي

قمت في إدخالها

و ذلك سيكون على الشكل التالي:

```

C:\ "C:\Documents and Settings\Administrator\... \نامر_C\Deb
enter the rateb day : 10
enter the rateb alkamis: 15
enter masruf aljum3a: 25
enter the money in the Bank:500
enter how days: 7
*****
*****
*****

this is your money : 540 *** since days: 7
Press any key to continue_

```

للتأكد:  $10 * 5 = 50$  مرتب خمسة أيام من السبت إلى الأربعاء  
و 15 مرتب يوم الخميس = 65 + المال المتوفر في البنك 500 = 565  
ناقص مصروف يوم الجمعة 25 = 540 مجموع المال الكلي المتوفر لدى الموظف بعد  
أسبوع و الإجابة صحيحة كما في البرنامج

أو خلال عام ٣٦٠ يوم سيكون على الشكل التالي:

```

C:\ "C:\Documents and Settings\Administrator\... \نامر_C\Deb
enter the rateb day : 20
enter the rateb alkamis: 30
enter masruf aljum3a: 50
enter the money in the Bank:1000
enter how days: 360
*****
*****
*****

this is your money : 5790 *** since days: 360
Press any key to continue

```

## شرح للأسطر البرنامج :

في الحقيقة عندما تقرأ ما المطلوب منك للوهلة الأولى تعتقد أن كود هذا البرنامج سيكون في عشرات الأسطر حتى ينفذ المطلوب، ولكن في الحقيقة مع فهمنا و استيعابنا لكيفية استخدام أدوات و جمل التحكم في C++ يوفر عليك الكثير من الوقت و العناء.

بعد أن قمنا في تعريف المتغيرات المراد استخدامها و التي قمتُ في توضيحها في الصفحة السابقة،

في البداية قمتُ في إسناد قيمة المال الموجودة في البنك إلى مجموع المال الكلي من خلال التعليمة التالية :

```
full_money = money_bank;
```

## تحذير:

و أرجوا الانتباه إلى كلمة إسناد (=) و التي نرمز لها بإشارة يساوي واحدة فقط، و لو قمنا في وضع إشارتين (=) فهذا يعني أن قيمة المال الموجود في البنك تساوي نفس قيمة المال الكلي و هذا يعتبر خطأ برمجي فادح .

و بعد ذلك قمت في عملية التعرف على طبيعة عدد الأيام التي قام المستخدم في إدخالها

وذلك من خلال جملة التكرار for و جملة التحكم if .

فمن خلال التعليمة التالية:

```
for(int s=1;s<=days;s++)
```

قمت في صنع حلقة كعداد للأيام العمل، تبدأ من يوم رقم واحد ( حيث أنه من الطبيعي لا يوجد أقل من يوم واحد دوام ) و نهاية الحلقة هنا الذي يحددها هو المستخدم من خلال

إدخاله لعدد الأيام من خلال المتغير days ، و هذه الحلقة تتزايد بمقدار واحد كل مرة

s++

فمن خلال هذا العداد سوف يتم حساب مجموع الأيام الكلي التي قام الموظف في إدخالها ، ولكن كيف سوف يتعرف البرنامج أن هذا اليوم هو الخميس أو الجمعة أو أي يوم آخر،

في الحقيقة هنا يأتي دور المبرمج في الإبداع و التفكير في حل مثل هذه المشاكل التي تواجهه .

في الحقيقة الأسبوع عبارة عن سبعة أيام {1,2,3,4,5,6,7} و اليوم الأول هو السبت أي رقم ١ ، الأحد رقم ٢ ، الاثنين ٣ ، الثلاثاء ٤ ، الأربعاء ٥ ، الخميس ٦ ، الجمعة ٧

وعلى هذا الأساس سوف نتابع حل هذه المسألة

و لكي يتعرف البرنامج على أي يوم من أيام الأسبوع هذا فيمكننا ذلك من خلال if

```
if(s%6==0)
{
full_money+=rateb_alkamis;
}
```

فمن خلال جملة الاختيار if هذه سوف يتم تعرف البرنامج على يوم الخميس فقط و ذلك على الشكل التالي :

يتم اختبار عدد الأيام ( المدخلة من قبل الموظف ) من خلال if حيث أنه كل يوم يكون باقي قسمته يساوي ٦ ( و ٦ يعني أن هذا اليوم هو يوم الخميس) يتم إضافة راتب يوم الخميس إلى مجموع المال الكلي

و إذا لم يكن باقي قسمة العدد ٦ يساوي الصفر ننتقل إلى المرحلة التالية:

```
else
{
    if(s%7==0)
    {
        full_money-=saruf_aljum3a;
    }
}
```

يتم في جملة الاختبار if هذه التحقق من أن هذا اليوم هو يوم الجمعة .  
حيث أنه إذا كان باقي قسمة اليوم على ٧ يساوي الصفر فهذا يعني أن هذا اليوم هو يوم  
الجمعة ( يوم الجمعة رقمه ٧ ) .

و بعد التحقق من أن هذا اليوم هو يوم الجمعة يتم إنقاص مصرف يوم الجمعة من  
مجموع المال الكلي .  
أما في السطر التالي :

```
else
{
    full_money+=rateb_day;
}
```

فإنه إن لم يكن هذا اليوم هو يوم الخميس و لا يوم الجمعة فهذا يعني أنه أي يوم من  
السبت إلى الأربعاء، و سيتم إضافة المرتب اليومي للموظف إلى مجموع المال الكلي.

أما في هذا السطر التالي فيتم عملية الخرج على الشاشة للمجموع المال الكلي المتبقي  
لدى الموظف بعد عدد الأيام التي قام الموظف في إدخالها

```
cout<<"this is your money : "<<full_money
<<" *** since days: "<<days<<endl;
```

**ملاحظة :**

لقد وضعت هنا المتغيرات من نوع int أي من نوع الأعداد الصحيحة ، فلو أدخلنا عدد عشري على سبيل المثال فسوف يخرج لنا خطأ أثناء مرحلة تنفيذ البرنامج كالتالي:

```

"C:\Documents and Settings\Administrator\... \namer_C\Debug\namer_C.exe"
enter the rateb day : 10.50
enter the rateb alkamis: enter masruf aljum3a: enter the money in the Bank:enter
how days:
*****
*****
*****
this is your money : -858993460 *** since days: -858993460
Press any key to continue_

```

لذلك يرجى الانتباه فإذا أردت إدخال عدد غير طبيعي، حينئذ يجب عليك أولاً القيام في تغيير نوع المتغيرات كالنوع float أو double .

## الجملة while :

إن كنت قد قرأت و تعلمت جملة التكرار for فإن جملة التكرار while ستكون بإذن الله أمر من البديهي بالنسبة لك.

و ذلك لأن while ما هي إلا نسخة من جملة التكرار for مع وجود بعض الاختلافات البسيطة في الشكل العام و الوظيفة.

و في الحقيقة هنا يكمن قوة هذه اللغة C++ حيث أنها لا تحصرك في جملة تكرار واحدة و يجب عليك التقيدُ بها و كتابة أكواد برامجك من خلالها، بل تعطيك الحرية و المجال الواسع في اختيار التعليمات المناسبة و جمل التحكم المناسبة لكتابة برامجك، و سنرى من خلال هذه الجملة و التعليمة الجديدة while مجال و حرية أوسع في البرمجة أكثر بكثير من اعتمادنا فقط على جملة التكرار for وحدها فقط.

مع العلم أنه يمكن كتابة بعض البرامج من دون استخدام while ، ولكن من خلال هذه التعليمة يمكنك كتابة برامج أكثر ليونة و مرونة و قابلة للتطوير من غيرها.

### ملاحظة:

جملة التكرار for تعتبر أكثر شيوعاً من جملة التكرار while من حيث الاستخدام و ذلك بسبب قوة و ليونة و تنظيم for عن while. مع العلم أن المبرمجون لا يستطيعوا التخلي عن while، و ذلك أن لكل واحدة من هاتين الجملتين لها موقعها الخاص في كود البرنامج، و الموقع الخاص لكل واحدة هو أنت بنفسك الذي سوف تتعرف عليه و ذلك من خلال التجربة و الخبرة في كتابة برامجك.

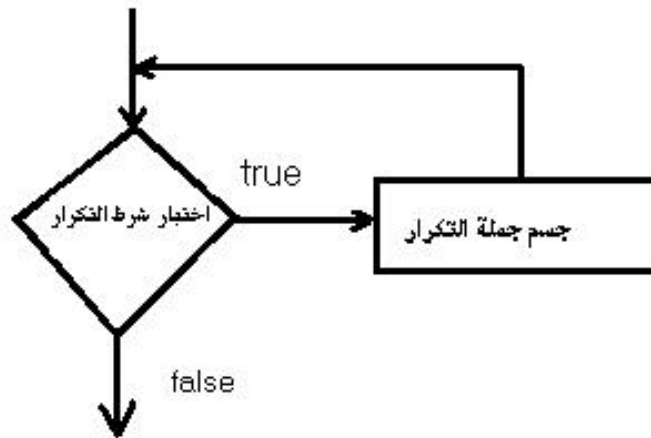
و الشكل العام لجملة التكرار while يكون على الشكل التالي :

## **while (expression) statement**

و طبعاً كما ترى من خلال الشكل العام أن while تتألف من الشكل التالي:  
أولاً من الكلمة الأساسية while

- و يأتي بعد الكلمة الأساسية قوسين إلزامية، و يحصران هذين القوسين شرط الجملة من أجل عملية التكرار
- إذا كان شرط التكرار صحيح (true) يتم تنفيذ جسم جملة التكرار while

و هذا المخطط التوضيحي للجملة while التكرارية:



المخطط التوضيحي لبنية التكرار while



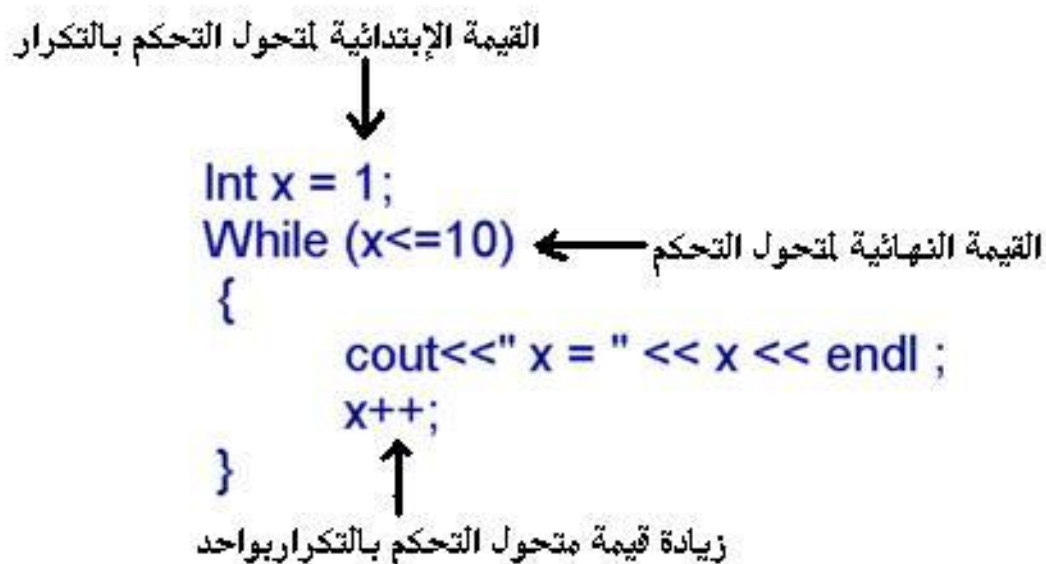
## مبدأ عمل الجملة **while**:

من خلال المخطط التوضيحي لبنية الجملة **while** فإن مبدأ عمل هذه الجملة يكون على الشكل التالي :

عندما تبدأ مرحلة الترجمة (compile) للجملة **while** يبدأ أولاً المترجم في التحقق من شرط الجملة **while** الموجود داخل القوسين الواقعان بعد **while** فوراً، و النتيجة إما أن تكون صح (true) أو خطأ (false) إذا كان شرط الجملة محقق **true** يتم تنفيذ التعليمات الموجودة داخل جسم الجمل ، **while**

و بعد تنفيذ ما بداخل جسم الجملة **while** مرة واحدة، يعود المترجم مرة أخرى للتحقق من صحة شرط الجملة **while** ..... و هكذا سيبقى المترجم في التحقق من شرط الجملة **while** و تنفيذ التعليمات الموجودة ضمنه ما زالت النتيجة **true**، و عندما تصبح النتيجة **false** يخرج المترجم من الجملة **while** و يتوقف عن تنفيذ التعليمات الموجودة ضمنه، لينتقل إلى أسطر البرنامج التالية إن وجدت.

ولنأخذ مثال لتوضيح الأجزاء المؤلفة للبنية for التكرارية :



و لكي ترى في نفسك التشابه الكبير بين for و while أعمل مقارنة بين الأجزاء

المؤلفة لكل من الجملتين، و لاحظ التالي:

- ١ - مكان توضع متحول التحكم بالتكرار و قيمته.
- ٢ - مكان توضع القيمة النهائية لمتحول التحكم بالتكرار.
- ٣ - مكان توضع زيادة قيمة متحول التحكم بالتكرار.

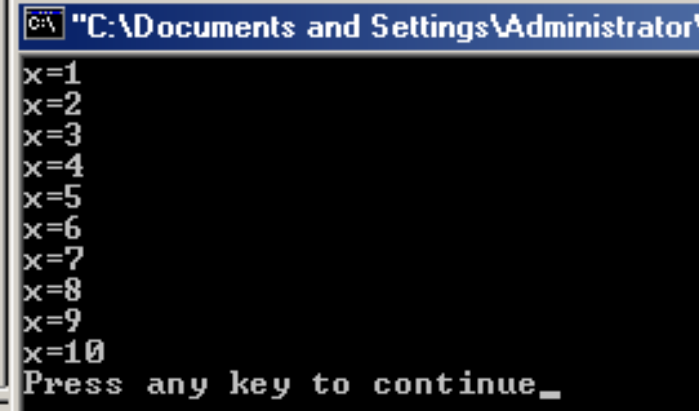
أما الآن لنأخذ مثال عملي ليتكون لدينا مفهوم أوضح لهذه الجملة.

## مثال ١:

إخراج على الشاشة الأعداد من ١ إلى ١٠ بشكل تصاعدي.

الحل:

```
# include <iostream.h>
int main()
{
    int x=1;
    while(x<=10)
    {
        cout<<"x="<<x<<endl;
        x++;
    }
    return 0;
}
```



## شرح للأسطر البرنامج :

لا أعتقد بوجود شيء ذا صعوبة للشرح، و إن صعب عليك فهم شيء فتمعن جيداً في أسطر البرنامج و بإذن الله عز و جل لن تجد أي صعوبة.

## مثال ٢:

أكتب برنامج يطلب من المستخدم إدخال حرف معين، ويكرر البرنامج الطلب ما لم يدخل المستخدم الحرف n.

الحل:

```
#include <iostream.h>

int main()
{
    char h;
    cout << "enter the char :\n";
    cin >> h;
    while(h != 'n')
    {
        cout << "This is not char. Enter another one...\n";
        cin >> h;
    }
    return 0;
}
```

```
C:\Documents and Settings\Administrator\... \namer_c\Debug\namer_c.exe
enter the char :
a
This is not char. Enter another one...
d
This is not char. Enter another one...
g
This is not char. Enter another one...
n
Press any key to continue
```

## ملاحظة هامة:

لاحظ وجود الحرف n داخل قوسين، لماذا؟ (ذكرتُ هذه الملاحظة سابقاً) و لو نسيت وضع الحرف داخل قوسين سوف يخرج لك المترجم الخطأ التالي:  
**error C2065: 'n' : undeclared identifier**

## مثال ٣:

أكتب برنامج لحساب المعدل الحسابي (مجموع العلامات مقسوم على عددهن) لعلامات طالب فرضاً أن هذا الطالب قام في تقديم خمسة امتحانات في مواد معينة لا على التعيين

## الحل:

من أجل كتابة هذا البرنامج اخترت للمتغيرات التي سوف استخدمها الأسماء التالية:  
 المتغير grade و الذي يعبر عن درجة علامة الطالب المدخلة .  
 و المتغير total\_grade لحساب مجموع علامات الكالب المدخلة.  
 و المتغير counter\_grade يستخدم كعداد لحساب عدد مرات العلامات المدخلة.  
 و المتغير sum لحساب المعدل الوسطي للعلامات المدخلة.

```
#include <iostream.h>

int main()
{
    cout<<"\t\t*****\n";
    cout<<"\t\t*** program total garde ***\n";
    cout<<"\t\t*****\n";
    double grade=0,sum=0;
    double total_grade=0,counter_grade=1;

    while(counter_grade<=5)
    {
        cout<<"enter the grade: \n";
        cin>>grade;
        total_grade+=grade;
        counter_grade+=1;
    }
    sum=total_grade/5;
    cout<<"sum your total grade is = "<<sum<<endl;

    return 0;
}
```

```
C:\Documents and Settings\Administrator\...
*****
*** program total garde ***
*****
enter the grade:
65.50
enter the grade:
75
enter the grade:
80
enter the grade:
60
enter the grade:
92
sum your total grade is = 74.5
Press any key to continue_
```

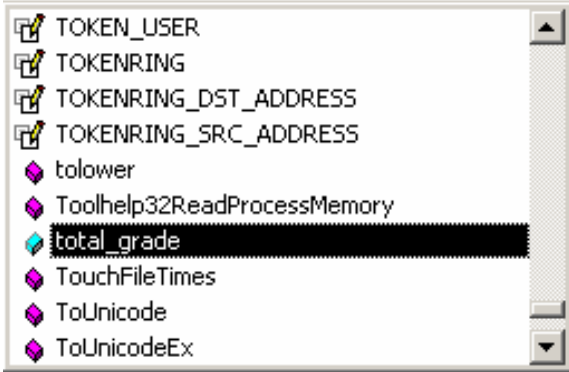
## ملاحظات جيدة حول كود البرنامج:

١- لا تنزعج من طول أسماء المتغيرات التي قمت في اختيارها، لأنه في اعتقادي أن كل واحدة من أسماء المتغيرات يجب أن تعبر عن ما يقصد التعبير من خلالها، و إن كنت تتضايق أو يصعب عليك من إعادة كتابة اسم المتغير في كل مرة ، فإن في برنامج Visual C++ يوجد الاختصار التالي `ctrl+alt+t` و الذي يقوم في تسهيل المهمة و يخفف العناء عنك في إعادة أسماء المتغيرات في كل مرة.

و ذلك يتم على الشكل التالي:

بعد كتابة أول أو ثاني حرف من اسم المتغير أضغط على `ctrl+alt+t` وسوف تخرج لك قائمة مُنسدلة لتعطيك الخيار في انتقاء اسم المتغير المراد اختياره و ذلك كما هو موضح على الشكل التالي:

```
cout<<"enter the grade: \n";
cin>>grade;
to
```



لذلك لا تبخل على برامجك في توضيح أسماء المتغيرات، حتى و إن تطلب ذلك منك كتابة اسم المتغير في ثلاث كلمات إن كنت تكتب برامجك في محرر Visual C++

٢- و لاحظ أنه أيضاً قمنا في إعطاء قيم أولية للمتغيرات و هي القيمة صفر (ما عدا القيمة counter\_grade و التي سأقوم في شرح ذلك لاحقاً)

و قمنا في إعطاء المتغيرات قيم أولية للسبب التالي:

في الحقيقة عندما أعطينا للمتغيرات أسماء و عرفنا المتغيرات من أي نوع و ذلك يعني أنه حجزنا لهذه المتغيرات أماكن في الذاكرة . في هذه الحالة يقوم المعالج في إسناد قيمة مخزنة في الذاكرة سابقاً إلى المكان الذي حجز للمتغير في الذاكرة، وهذا يعني أن المتغير سيأخذ هذه القيمة أسندت له.

و سوف أقوم في شرح الكلام النظري السابق على شكل عملي و مرئي و واضح و ذلك من خلال بيئة عمل Visual C++ و ذلك على الشكل التالي:

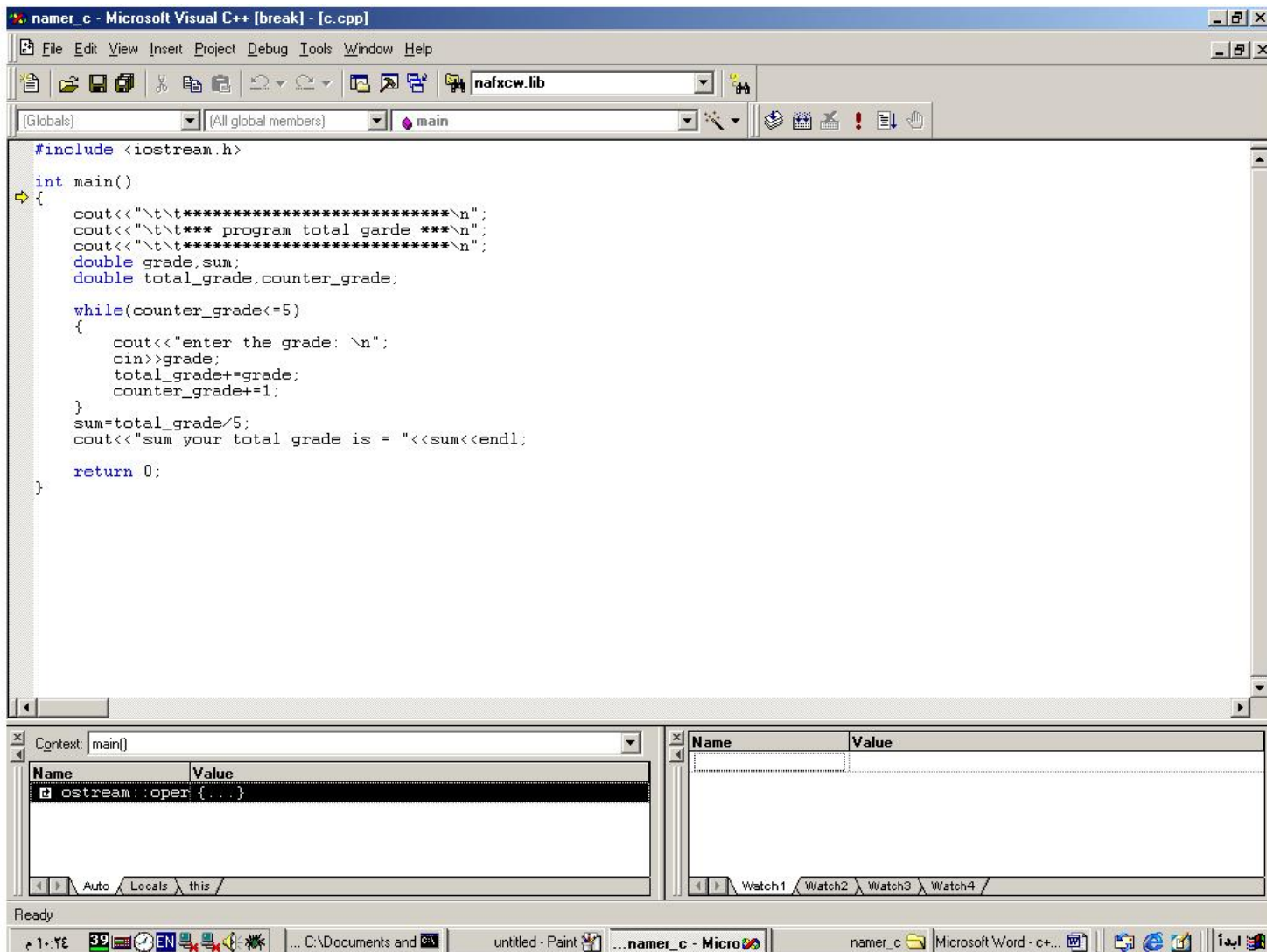
لكي ترى القيم التي تسند إلى المتغيرات في حالة عدم إعطائها قيم أولية إتبع الخطوات التالية:

١- لا تعطي للمتغيرات قيم أولية، أي قم في تعريفها من النوع double فقط و لا تضيف أي شيء غير ذلك.

٢- بعد التحقق من عدم وجود أخطاء و ذلك بعد الضغط على ctrl+f7

٣- قم في الضغط فقط على الزر f10

ستظهر لك في البداية شاشة سوداء ثم تختفي بسرعة، وسيبقى على الشاشة أمامك بيئة Visual C++ على الشكل التالي:



في هذا الشكل يرجى الانتباه إلى النقاط التالية:

- ١- إلى السهم الأصفر الصغير الموجود في أعلى الصفحة لليسار و هذا السهم يسمى بالكاشف لأنه يكشف لنا عن الدخل والخرج و المتغيرات الموجودة ضمن البرنامج و ذلك من خلال على الزر f10 كل مرة (مرة و احدة)
- ٢- و يرجى الانتباه إلى النافذة السفلى لليسار، و هذه النافذة تعرض لنا تعليمات الدخل و الخرج و أسماء المتغيرات في كود البرنامج، و ذلك بعد الضغط على الزر f10 كل مرة.
- ٣- و أخيراً النافذة السفلى لليمين، و في هذه النافذة تكتب أسماء المتغيرات التي تريد معرفة قيمها، فعلى سبيل المثال قم في كتابة اسم المتغير grade على الشكل التالي



Name	Value
grade	CXX0069: Error: variable needs stack fram

◀ ▶ Watch1 Watch2 Watch3 Watch4

و كما ترى فإن هذه النافذة تتألف من حقلين :

١ - name للإدخال اسم المتغير (و أنا هنا قمت في إدخال المتغير grade)

٢ - value والتي يقوم الكمبيوتر من خلالها في عرض قيمة المتغير الذي قمت في إدخاله، في هذه المرحلة يعرض لنا العنوان CXX0069 بالنظام الست عشري و الذي يعبر عن وجود خطأ.

يرجى الانتباه إلى أن الخطأ هنا ليس لأنه لم نعطي قيم ابتدائية للمتغير، ولكن لأن القوس الأصفر ( أي المؤشر ) ما زال خارج نطاق الوظيفة main .

## كيفية التعرف على قيم المتغيرات من خلال Visual C++ :

- ١- قم في كتابة أسماء المتغيرات في النافذة السفلى لليمين في الحقل name ( التي تريد بالتعرف على قيمها) و ذلك على الشكل التالي:

Name	Value
grade	CXX0069: Error: variable needs stack fram
sum	CXX0069: Error: variable needs stack fram
total_grade	CXX0069: Error: variable needs stack fram
counter_grade	CXX0069: Error: variable needs stack fram

- ٢- قم في الضغط على الزر f10 ضغطة واحد كل مرة مع الانتباه إلى التغير الذي سيطرأ على قيم المتغيرات، وذلك سيكون على الشكل التالي:

Name	Value
grade	-9.2559631349318e+061
sum	-9.2559631349318e+061
total_grade	-9.2559631349318e+061
counter_grade	-9.2559631349318e+061

- ٣- تابع في الضغط على الزر f10 حتى ينطلق صوت من الجهاز يعلن عن إنتهاء استخدام قيم المتغيرات المدخلة.

ما الذي تلاحظه بعد هذا كله:

نلاحظ أن قيم المتغيرات لم تتغير لا قبل و لا بعد استخدامها و هذا يعني أن البرنامج استخدم قيم للمتغيرات ضمن المجال المعلن في حقل القيمة value، مما يعني أنه على الرغم من صحة كود برنامجنا و على الرغم من أن المترجم لم يخرج لنا أي خطأ على الإطلاق إلا أن النتيجة بعد إدخال علامات الطالب ليست صحيحة.

أما الآن لنقم في نفس العملية السابقة و لكن مع إعطاء قيم ابتدائية للمتغيرات، و سنلاحظ الفرق بين الحالتين و لكن أولاً يجب عليك العودة إلى محرر الأكواد وذلك بالضغط على f7 و من ثم على الزر enter

و بعد إعطاء قيم للمتغيرات و العودة لرؤية قيم المتغيرات المجوزة في الذاكرة بالضغط مرة أخرى على F10 سنلاحظ كالتالي:

١- عرض المجال لقيم المتغيرات الموجودة ضمنها

Name	Value
grade	-9.2559631349318e+061
sum	-9.2559631349318e+061
total_grade	-9.2559631349318e+061
counter_grade	-9.2559631349318e+061

ففي هذه المرحلة من البرنامج أعطيت للمتغيرات قيم غير معروفة من الذاكرة.

٣- نتابع بالضغط على F10 حتى يصل المؤشر إلى المتغير sum, grade عندئذ سيظهر لنا الشكل التالي:

Name	Value
grade	0.000000000000000
sum	0.000000000000000
total_grade	-9.2559631349318e+061
counter_grade	-9.2559631349318e+061

نلاحظ أنه تم تصفير المتغيران و أصبحت القيم الابتدائية لهما يساوي الصفر. و لاحظ الفرق مع المرحلة السابقة حين كانت المتغيرات بدون قيم أولية.

٤ - نتابع بالضبط على F10 حتى يصل المؤشر إلى المتغير , `total_grade` , `counter_grade` عندئذ سيظهر لنا الشكل التالي:

Name	Value
grade	0.0000000000000000
sum	0.0000000000000000
total_grade	0.0000000000000000
counter_grade	1.0000000000000000

نلاحظ أنه تم تصفير المتغيران و أصبحت القيمة الابتدائية للمتغير `total_grade` يساوي الصفر، و قيمة المتغير `counter_grade` تساوي الواحد.

**لماذا قمنا في إعطاء المتغير قيمة ابتدائية تساوي الواحد و ليس الصفر؟**

في الحقيقة لو قمنا في إعطاء المتغير `counter_grade` قيمة ابتدائية ٠ بدلاً من ١ فسينتج لدينا خطأ منطقي على الشكل التالي:

في كود البرنامج قمنا في إنشاء حلقة `while(counter_grade<=5)`

و عمل هذه الحلقة هو كعداد يحسب كم مرة يدخل الطالب درجة علامة في الامتحان، و في نص المثال طلب منا فرضاً أن هذا الطالب قام في تقديم خمسة امتحانات فقط لا غير.

فلو كان للمتغير `counter_grade` قيمة الابتدائية تساوي ٠ فإن `while` ستبدأ بالعد من

الرقم ٠ إلى الرقم ٥ (كما هو معروض في شرط التكرار) ،

و الأرقام من ٠ إلى ٥ عددها ٦ أي أن البرنامج سيطلب منا إدخال ست علامات للطالب و ليس خمسة كما طلب منا في نص المثال.

لذلك يجب عليك الحذر في كتابة شرط التكرار للجملة `while`، فإذا لم يعرض لك المترجم بوجود أخطاء هذا لا يعني أن برامجك خالية من الأخطاء تماماً.

## الخلاصة:

يجب إعطاء قيم ابتدائية للمتغيرات ضمن البرنامج من أجل الحصول على نتيجة صحيحة مئة بالمئة و من دون شكوك في النتائج، و إلا سيوف يُعطى قيم أخرى للمتغيرات (هذه القيم تكون في الموضع الذي حجز للمتغير في الذاكرة ) مما يؤدي إلى الحصول على نتائج خاطئ على الرغم من صحة كود البرنامج بالنسبة للمترجم.

**مثال ٤:**

أكتب برنامج لحساب المعدل الحسابي (مجموع العلامات مقسوم على عددهن) لعلامات طالب، مع العلم أن الطالب نفسه هو الذي سوف يحدد عدد الأمتحانات المراد إدخالها لحساب معدلها، أي أن البرنامج يعطيه حرية الاختيار.

**الحل:**

في الحقيقة هذا المثال ما هو إلا عبارة عن تعديل للمثال السابق، وسوف نجري بعض التغييرات البسيطة على المثال السابق لحل هذا المثال، وذلك سيكون على الشكل التالي:

```
#include <iostream.h>

int main()
{
    cout<<"\t\t*****\n";
    cout<<"\t\t*** program total garde ***\n";
    cout<<"\t\t*****\n";

    int number=0;
    double grade=0,sum=0;
    double total_grade=0,counter_grade=1;
    cout<<"enter how examine:\n";
    cin>>number;

    while(counter_grade<=number)
    {
        cout<<"enter the grade: \n";
        cin>>grade;
        total_grade+=grade;
        counter_grade+=1;
    }
    sum=total_grade/number;
    cout<<"sum your total grade is = "<<sum<<endl;

    return 0;
}
```

و بعد تنفيذ البرنامج سوف تخرج لنا شاشة تخبرنا عن عدد الامتحانات التي قام الطالب في تقديمها، و من ثم تطلب منا درجة علامة كل امتحان، و بعد ذلك سيخرج لنا البرنامج مجموع متوسط العلامات المدخلة، و ذلك سيكون على الشكل التالي:

```

C:\ "C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\
*****
*** program total grade ***
*****
enter how examine:
3
enter the garde:
77.60
enter the garde:
88
enter the garde:
85.50
sum your total grade is = 83.7
Press any key to continue_

```

في هذا البرنامج لم نوظف أي شيء جديد غير إضافة المتغير number و وظيفته على الشكل التالي:

يأخذ من الطالب عدد الامتحانات التي قام في إدخالها الطالب، و من ثم بعد ذلك يمرر المتغير number القيمة المدخلة إلى شرط إستمرار جملة التكرار while و هنا ينتهي عمل المتغير number داخل while ، أما عمله بعد الجملة while فهو عبارة عن تمرير قيمته إلى المعادلة التالية

$$\text{Sum} = \text{total\_grade} / \text{number}$$

لتقسيم مجموع علامات الطالب على عدد الامتحانات.

و هذا هو كل برنامجنا.

## مثال ٥:

هذا المثال ما هو إلا عبارة عن تطوير بسيط للمثال الأساسي في هذا الكتاب و الذي تدور فكرته حول تقديم طلب انتساب للجامعة.

ففي هذه المرحلة سنطور برنامجنا على الشكل التالي:

- ١- تحديد عدد الطلاب المنتسبين إلى كل كلية من كليات الجامعة، حيث أنه من المعلوم أن كل كلية تستطيع استيعاب عدد محدد من الطلاب.  
(و سنحدد نحن هنا فرضاً ٣ طلاب لكل كلية و ذلك للاختصار فقط)
- ٢- إخراج رسالة على الشاشة تخبرنا باكتمال عدد الطلاب المنتسبين إلى الكلية و من ثم الخروج من البرنامج.



## كود البرنامج:

```

#include <iostream.h>
main()
{
    int n;
    int number_student=1; // while شرط استمرار جملة التكرار

    cout<<"\n\n";
    cout<<"\t\t\t ***** \n";
    cout<<"\t\t\t *** Welcome to University *** \n";
    cout<<"\t\t\t ***** \n\n\n";
    cout<<" \t 1- to enter Faculty Of Information Technology
Computer .\n";
    cout<<" \t 2- to enter Faculty Machine .\n";
    cout<<" \t 3- to EXIT .\n";
    start_menu:
    cin>>n;
switch (n)
{
    case 1:

        cout<<"\t\t\t *****\n";
        cout<<"\t\t\t *** Faculty Computer ***\n";
        cout<<"\t\t\t *****\n";

        while(number_student<=3)
        {
            number_student++;

            int grade;
            float grmath;

            cout<<" enter the grade: \n ";
            cin>>grade;

```

```

if(grade>=65)
{
    cout<<"welcome to University "<<">>"<<"\t student
    number:"<<number_student-1<<endl<<endl;
}
else
{
    if(grade<50)
    {
        cout<<"Ecxuse! your grade less than 50!\n";
        return 0;
    }
    cout<<"excuse your grade less than 70!:\n";
    cout<<"But how your grade int the mathmatic!:\n";
    cout<<"enter the grade mathmatic:\n";
    cin>>grmath;

if(grmath>=70 && grade>=50)
{
    cout<<"welcome to University
    "<<">>"<<"\tstudent
    number:"<<number_student-1<<endl<<endl;
}

    else
    {
        cout<<"Ecxuse! your grade less than 70!
\n";

        cout<<"and ecxuse your grade mathmatic
less than 80\n";
    }
}
}

```

```

    cout<<"\t\t\t^^^^^^^^^^^^^^^^^^^^\n";
    cout<<"\t\t\t calss is full\n\n";
    cout<<"\t\t\t^^^^^^^^^^^^^^^^^^^^\n";
    break;

```

case 2:

```

    cout<<"\t\t\t *****\n";
    cout<<"\t\t\t *** Faculty machin ***\n";
    cout<<"\t\t\t *****\n";

    while(number_student<=3)

    {
        number_student++;
        int grade_machine;
        float grphysic_machine;

        cout<<" enter the grade: \n ";
        cin>>grade_machine;

        if(grade_machine>=65)
        {
            cout<<"welcome to University "<<">>"<<"\tstudent
            number:"<<number_student-1<<endl<<endl;
        }
        else
        {
            if(grade_machine<50)
            {
                cout<<"Ecxuse! your grade less than 50!\n";
                return 0;
            }
            cout<<"excuse your grade less than 65!:\n";
            cout<<"But how your grade int the phisic!:\n";

            cout<<"enter the grade phisic:\n";
            cin>>grphysic_machine;

```

```

if(grphysic_machine>=70 && grade_machine>=50)
    {
        cout<<"welcome to University
        "<<">>"<<"\tstudent
        number:"<<number_student-1<<endl<<endl;
    }

else
    {
        cout<<"Excuse! your grade less than 65! \n";
        cout<<"and excuse your grade grphysic less
than 70\n";
    }
}
}
}
cout<<"\t\t\t^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n";
cout<<"\t\t\t calss is full\n\n";
cout<<"\t\t\t^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n";
    break;

case 3:
    return 0;

    break;

default:
    cout<<"erorr? enter agin\n\n";
    goto start_menu;
    break;
}

return 0;
}

```

## مرحلة التنفيذ:

بعد تنفيذ البرنامج قمت في البداية في اختيار رقم واحد و الذي هو عبارة عن كلية الكمبيوتر. و ذلك على الشكل التالي:

```

C:\ "C:\Documents and Settings\Administrator\ملف عمل\program_c\Debug\program_c.exe"

*****
*** Welcome to University ***
*****

1- to enter Faculty Of Information Technology Computer .
2- to enter Faculty Machine .
3- to EXIT .
1

*****
*** Faculty Computer ***
*****

enter the grade:
85
welcome to University >> student number:1

enter the grade:
56
excuse your grade less than 70!:
But how your grade int the mathematic!:
enter the grade mathematic:
90
welcome to University >> student number:2

enter the grade:
90
welcome to University >> student number:3

^^^^^^^^^^^^^^^^^^^^
calss is full
^^^^^^^^^^^^^^^^^^^^

Press any key to continue_

```

أو باختيارك لرقم ٢ كلية الميكانيك. و ذلك على الشكل التالي:

```

"C:\Documents and Settings\Administrator\ملف العمل\program\Debug\namer_c.exe"

*****
*** Welcome to University ***
*****

1- to enter Faculty Of Information Technology Computer .
2- to enter Faculty Machine .
3- to EXIT .

2

*****
*** Faculty machin ***
*****

enter the grade:
90
welcome to University >>      student number:1

enter the grade:
55
excuse your grade less than 65!:
But how your grade int the physic!:
enter the grade physic:
85
welcome to University >>      student number:2

enter the grade:
95
welcome to University >>      student number:3

~~~~~
      calss is full
~~~~~

Press any key to continue

```

## شرح أسطر البرنامج:

سأقوم في شرح الأسطر ذات اللون الأحمر فقط، و ذلك لأن بقية أسطر البرنامج معروفة و قد شرحت من قبل.

```
int number_student=1;
```

قمنا في إنشاء متغير number\_student من أجل شرط جملة التكرار while و هو هنا يقوم بوظيفة عداد لحساب عدد الطلاب المنتسبين.

### **سؤال:**

لماذا قمنا في إعطاء المتغير number\_student قيمة ابتدائية واحد و ليس صفر؟

أجب عن هذا السؤال من خلال مراجعتك للمثال السابق.

أما الأسطر التاليه:

```
while(number_student<=3)  
{  
    number_student++;
```

فهي كما ترون عبارة عن جملة التكرار while لتقوم في عملية إعادة و تكرار طلب الانتساب ما دام الشرط الموجود داخل القوسين الذين يلين while محقق .

أما number\_student فهو عبارة عن شرط تنفيذ تكرار الجملة while و الذي هو هنا إيقاف تنفيذ عمل while عندما يكتمل عدد الطلاب إلى ٣ طلاب

أما بالنسبة إلى number\_student++ فهي عبارة عن إضافة واحد كل مرة يتم تنفيذ فيها الجملة while

أما التعليمة التالية فهي عبارة عن إخراج على الشاشة رقم الطالب المنتسب إلى الجامعة

```
cout<<"welcome to University "<<">>"<<"\tstudent  
number:"<<number_student-1<<endl<<endl;
```

أما بالنسبة إلى التعليمات ذات اللون الأحمر التي بداخل case 2 فهي نفس الشيء بالنسبة إلى case 1 و لا يوجد أي اختلاف بينهما .

و هذا كل شيء جديد في برنامجنا.



## ت- الجملة do- while:

و هي الجملة الأخيرة من جملة الحلقات loop في لغة C++، و في الحقيقة يوجد شبه بين الجملة while و الجملة do/while من حيث مبدأ العمل و البنية، مع وجود بعض الاختلافات أيضاً من حيث مبدأ العمل و تركيب الجملة، و سنرى نقط التشابه و الاختلاف بينهما و ذلك بعد التعرف على تركيب بنية do/while .

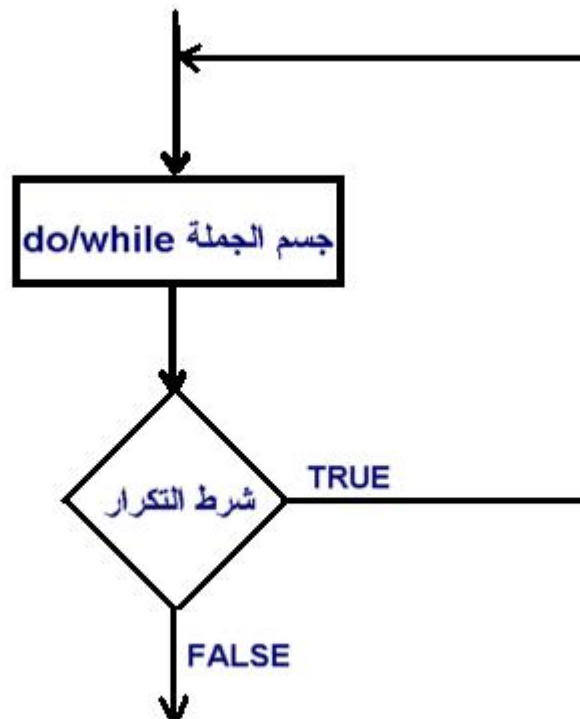
و الشكل العام لجملة التكرار do/while يكون على الشكل التالي :

```
do
{
    statement
}
while(condition);
```

و طبعاً كما ترى من خلال الشكل العام أن do/while تتألف من الشكل التالي:  
أولاً من الكلمة do و التي تعني افعل

- و يأتي بعد الكلمة do قوسين كبيرين، و يحصران هذين القوسين التعليمات و الأوامر التي يجب تنفيذها
- و بعد القوسين تأتي الكلمة while و بعدها قوسين صغيرين يحصران بداخليهما شرط التكرار.
- و بعد القوسين نضع فاصلة منقوطة للدلالة على انتهاء الجملة do/while .

## مخطط توضيحي لعمل الجملة do/while:



مخطط توضيحي لعمل الجملة do/while

و لاحظ معي أخي الكريم مكان توضع شرط التكرار و جسم الجملة في do/while و قارن ذلك في مكان توضع شرط التكرار و جسم الجملة بالنسبة إلى الجملة .while.

**مبدأ عمل do/while :**

كما قلت سابقاً أنه يوجد تشابه بين الجملة `while` و `do/while` من حيث مبدأ العمل بالنسبة إلى الجملة `while` فإن عملها يتم على الشكل التالي:  
 يتم أولاً من التحقق من شرط التكرار قبل تنفيذ أي شيء من جسم الجملة و من ثم بعد التحقق من صحة شرط التكرار يتم تنفيذ التعليمات الموجودة داخل `while` وإذا لم يتحقق شرط التكرار فلا يتم تنفيذ أي شيء من التعليمات الموجودة داخل `while`

أما بالنسبة إلى `do/while` فإن التحقق من صحة شرط التكرار يتم بعد تنفيذ جسم الجملة `do/while` أولاً .  
 و بعد التنفيذ المرة الأولى يتم التحقق من شرط التكرار فإذا كانت النتيجة `true` يتم التنفيذ مرة أخرى و هكذا حتى تصبح نتيجة التحقق من شرط التكرار `false` .

و الفرق بين `while` و `do/while` هو أنه في الأخيرة يتم تنفيذ جسم الجملة `do/while` مرة و احدة على الأقل و ذلك على عكس الجملة `while` فمن الممكن عدم تنفيذ تعليماتها حتى و لا مرة واحدة إذا لم يتحقق شرط التكرار.

و حتى يتضح لك مفهوم عمل الجملة `do/while` فلا بد من دعم الكلام النظري السابق بالأمثلة لكي تساعدك على استيعاب عمل هذه الجملة.

## مثال ١:

أخراج على الشاشة الأعداد من واحد إلى عشرة من الجملة do/while

الحل:

```
# include <iostream.h>
int main()
{
    int x=1;
    do
    {
        cout<<"this is x: "<<x<<endl;
        x++;
    }
    while(x<=10);
    return 0;
}
```

```
"C:\Documents and Settings\Administrator\المطبوعات\n...
this is x: 1
this is x: 2
this is x: 3
this is x: 4
this is x: 5
this is x: 6
this is x: 7
this is x: 8
this is x: 9
this is x: 10
Press any key to continue_
```

## ملاحظات جيدة:

١ - عدم وضع الفاصلة المنقوطة بعد القوسين الواقعين بعد `while` يسبب خطأ أثناء مرحلة الترجمة وسيخرج لك خطأ على الشكل التالي:

'error C2143: syntax error : missing ';' before 'return

٢ - إن لغة C++ تعد من اللغات المرنة و اللينة في كتابة أكواد برامجها فالتعليمة في البرنامج السابق `x++` و التي ترمز إلى الزيادة بمقدار واحد كل مرة يتم فيها التكرار، يمكننا وضعها مع شرط التكرار في القوسين الصغريين بعد `while` و ذلك على الشكل التالي:

**`while(++x<=10) ;`**

و طبعاً هذا الشكل ليس جبرياً و فرضياً و لكن المبرمجون يختلفون في طرق كتابة أكواد برامجهم و ذلك تبعاً لم يرى المبرمج هو بنفسه الطريقة المثلى و الأسهل لمكان توضع التعليمات داخل البرنامج.

و لكن يرجى الملاحظة أن التعليمة `x++` يجب أن تكتب حصراً داخل جسم الجملة `do/while` و إلا أعتبر ذلك خطأ قواعدياً .

و أرجوا الانتباه إلى مكان توضع عملية الزيادة بالنسبة إلى المتغير في الحالتين :

الحالة الأولى في البرنامج و التي كتبت على الشكل التالي `x++`

و الحالة الثانية و التي كتبت على الشكل التالي `++x`

لماذا في الحالة الثانية قمنا في وضع عملية الزيادة بواحد أمامية و ليس خلفية؟

و هل ذلك سيغير في الخرج على الشاشة ؟

إذا كانت الإجابة نعم فلماذا؟

حاول أن تجيب على هذا السؤال في نفسك من خلال قرأتك لأنواع عمليات الزيادة  
بواحد الأمامية و الخلفية !!!

## مثال ٢ :

أكتب برنامج يقوم في طبع الأعداد على الشاشة، بحيث أن المستخدم هو الذي يحدد هذه الأعداد من أي رقم إلى أي رقم.

**الحل:**

في الحقيقة هذا المثال لا يخلف عن سابقه كثيراً من حيث المبدأ العمل حيث أنه يجب على البرنامج في هذا المثال كسابقة طبع على الشاشة أعداد من خلال do/while، و نقطة الاختلاف هي أنه في هذا البرنامج المستخدم هو الذي يتحكم في بداية و نهاية حلقة التكرار.

و على ذلك فإن كود هذا البرنامج لا يختلف عن سابقة سوى أنه يجب علينا أن نضيف لكود برنامجنا مجرى دخل و خرج من أجل التحكم في بداية حلقة التكرار و نهاية. و ذلك سيكون على الشكل التالي:

```
#include <iostream.h>
int main()
{
    int x=0,y=0; // لا تنسى وضع قيمة ابتدائية للمتحولات
    cout<<"enter the start number:\n";
    cin>>x; // القيمة الابتدائية للأعداد
    cout<<"enter the end number:\n";
    cin>>y; // القيمة النهائية للأعداد
    do
    {
        cout<<"this is your number:"<<x<<endl;
        x++; // زيادة قيمة المتحول بواحد كل مرة
    }
    while(x<=y); // شرط عملية التكرار

    return 0;
}
```

و بعد عملية التنفيذ سوف تظهر لدينا شاشة سوداء على الشكل التالي:

```
C:\ "C:\Documents and Settings\Admin
enter the start number:
3
enter the end number:
14
this is your number:3
this is your number:4
this is your number:5
this is your number:6
this is your number:7
this is your number:8
this is your number:9
this is your number:10
this is your number:11
this is your number:12
this is your number:13
this is your number:14
Press any key to continue
```

### وظيفة منزلية:

لاحظ معي أخي الكريم، في المثال السابق كانت نوع الأعداد التي قمنا في إدخالها من النوع الأعداد الصحيحة .int.

قم في تغيير نوع المتغيرات حيث يصبح بمقدورنا إدخال أعداد عشرية أي ذات فواصل؟

و بعد التعديل يجب أن تكون النتيجة على الشكل التالي:

```
C:\ "C:\Documents and Settings\Admin
enter the start number:
2.03
enter the end number:
9.5
this is your number:2.03
this is your number:3.03
this is your number:4.03
this is your number:5.03
this is your number:6.03
this is your number:7.03
this is your number:8.03
this is your number:9.03
Press any key to continue
```

**مثال ٣:**

هذا المثال ما هو إلا عبارة عن المثال الخامس في الجملة `while` و الذي تدور فكرته حول تطوير بسيط للمثال الأساسي في هذا الكتاب و الذي تدور فكرته حول تقديم طلب انتساب للجامعة و الفرق هو أنه سنقوم في تنفيذ الخطوات التالية من خلال `do/while`. ففي هذه المرحلة سنطور برنامجنا على الشكل التالي (كما في المثال السابق):

٣ - تحديد عدد الطلاب المنتسبين إلى كل كلية من كليات الجامعة، حيث أنه من المعلوم أن كل كلية تستطيع استيعاب عدد محدد من الطلاب.

(و سنحدد هنا فرضاً ٣ طلاب لكل كلية و ذلك للاختصار فقط)

٤ - إخراج رسالة على الشاشة تخبرنا باكتمال عدد الطلاب المنتسبين إلى الكلية و من ثم الخروج من البرنامج.

**الحل:**

في الحقيقة لم أقم في ذلك التغير الكبير سوى أنني قمت في وضع الجملة `do/while` مكان جملة التكرار `while`، و هذا هو كود البرنامج بعد التعديل، و لاحظ الفرق بين الطريقتين التي كتب فيها البرنامج من خلال `while` و الجملة `do/while`.



```

#include <iostream.h>
main()
{
    int n;
    int number_student=1; // شرط استمرار جملة التكرار

    cout<<"\n\n";
    cout<<"\t\t\t\t\t*****\n";
    cout<<"\t\t\t\t\t*** Welcome to University ***\n";
    cout<<"\t\t\t\t\t*****\n\n\n";
    cout<<"\t 1- to enter Faculty Of Information Technology Computer .\n";
    cout<<"\t 2- to enter Faculty Machine .\n";
    cout<<"\t 3- to EXIT .\n";
    start_menu:
    cin>>n;
    switch (n)
    {
    case 1:

        cout<<"\t\t\t\t\t*****\n";
        cout<<"\t\t\t\t\t*** Faculty Computer ***\n";
        cout<<"\t\t\t\t\t*****\n";

    do
    {
        number_student++;

        int grade;
        float grmath;

        cout<<" enter the grade: \n ";
        cin>>grade;

        if(grade>=65)
        {
            cout<<"welcome to University "<<">>"<<"\t student number:"<<number_student-1<<endl<<endl;
        }
        else
        {
            if(grade<50)
            {
                cout<<"Excuse! your grade less than 50!\n";
                return 0;
            }
            cout<<"excuse your grade less than 70!\n";
            cout<<"But how your grade int the mathematic!:\n";
            cout<<"enter the grade mathmatic:\n";
            cin>>grmath;

            if(grmath>=70 && grade>=50)
            {
                cout<<"welcome to University "<<">>"<<"\tstudent number:"<<number_student-1<<endl<<endl;
            }

            else
            {
                cout<<"Excuse! your grade less than 70! \n";
                cout<<"and excuse your grade mathmatic less    than 80\n";
            }
        }
    }
    while(number_student<=3);

    cout<<"\t\t\t\t\t^*****\n";
    cout<<"\t\t\t\t\t calss is full\n\n";
    cout<<"\t\t\t\t\t^*****\n";
    break;
}

```

```

case 2:
    cout<<"\t\t\t *****\n";
    cout<<"\t\t\t *** Faculty machin ***\n";
    cout<<"\t\t\t *****\n";

do
{
    number_student++;
    int grade_machine;
    float grphysic_machine;

    cout<<" enter the grade: \n ";
    cin>>grade_machine;

    if(grade_machine>=65)
    {
        cout<<"welcome to University "<<">>"<<"\tstudent number:"<<number_student-1<<endl<<endl;
    }
    else
    {
        if(grade_machine<50)
        {
            cout<<"Excuse! your grade less than 50!\n";
            return 0;
        }
        cout<<"excuse your grade less than 65!\n";
        cout<<"But how your grade int the physic!:\n";

        cout<<"enter the grade physic:\n";
        cin>>grphysic_machine;

        if(grphysic_machine>=70 && grade_machine>=50)
        {
            cout<<"welcome to University "<<">>"<<"\tstudent number:"<<number_student-1<<endl<<endl;
        }

        else
        {
            cout<<"Excuse! your grade less than 65! \n";
            cout<<"and excuse your grade grphysic less than 70\n";

        }
    }
}
while(number_student<=3);
cout<<"\t\t\t*****\n";
cout<<"\t\t\t calss is full\n\n";
cout<<"\t\t\t*****\n";
    break;

case 3:
    return 0;

    break;
default:
    cout<<"erorr? enter agin\n\n";
    goto start_menu;
    break;
}

return 0;
}

```

**مثال ٤ :**

أكتب برنامج يطلب من المستخدم إدخال عدد معين من الأحرف الأنكليزية لا على التعيين و يجب على البرنامج أن ينفذ الخطوات التالية قبل عملية الخروج من البرنامج:

- ١ - حساب عدد الأحرف المدخلة من النوع A.
- ٢ - حساب عدد الأحرف المدخلة من النوع B.
- ٣ - حساب عدد الأحرف المدخلة من النوع C.
- ٤ - حساب المجموع الكلي للأحرف المدخلة.
- ٥ - اختيار الحرف X للخروج من البرنامج. مع الأخذ بعين النظر أن X لا تعتبر من الأحرف المدخلة، بل حرف من أجل الخروج من البرنامج فقط.

**الحل:**

أعتقد في رأي الخاص قبل أن نبدأ في حل المسألة و المباشرة في كتابة كود البرنامج يجب علينا في البداية أن نحاول فهم نص المسألة و ما المطلوب منا عمله في هذا البرنامج، فإذا قمت بتلك الخطوات البسيطة و استوعبت نص المسألة و ما المطلوب منك فأنا أبشرك بأنك قد قمت في حل ٥٠% من المسألة أي نصف المسألة قد حُلت و لم يبقى سوى النصف الآخر و الذي في تصوري بأنه سيكون أسهل من النصف الأول. و أقدم نصيحة للأخوة المبرمجون و هذه من وجهة نظري الخاصة:

قبل البدء في كتابة كود أي برنامج سواء أكان المطلوب منك هو ليس بالكثير أو بالكثير حاول في البداية قراءة نص المسألة جيداً و استيعاب ما المطلوب منك، و حبذا لو كان في يدك و رقعة و قلم تدون النقاط المطلوب عملها منك و لا مناع من استخدام الرسم في ذلك مثل رسم المخططات التوضيحية أو ما شابه ذلك. و الهدف من وراء ذلك هو تكوين فكرة و صورة عن البرنامج الذي ستقوم في عمله.

أما الآن سوف نقوم في وضع النقاط الأساسية لبرنامجنا.  
 فمن خلال قراءة نص المسألة يستوجب علينا إيجاد متغير من النوع char و ذلك لأنه  
 طلب منا إدخال أحرف و ليس أعداد أو غير ذلك.  
 و من ثم إيجاد متغيرات من النوع int لحساب عدد الأحرف المدخلة الكلي و لكل من  
 A,B,C  
 و لكي يتم تعرف البرنامج على الحرف المدخل و تحديده دون غيره، فلا بد من  
 استخدام الجملة switch من أجل التعرف على الحروف المدخلة.  
 فإذا افترضنا أن المستخدم قام في إدخال الحرف A فسيتم مقارنة A مع كل فرع case  
 من فروع switch و عند حصول التطابق عندها يتم زيادة قيمة المتغير بواحد و من  
 ثم الخروج من هذا الفرع case عن طريق break  
 و طبعاً ذلك كله لا بد وضعة ضمن جملة التكرار do/while من أجل ضمان التكرار  
 ما لم نقم في عند إدخال X .

و هذا هو كود البرنامج

```

int main()
{
    char ch;
    int ch_a=0,ch_b=0,ch_c=0;
    int ch_number=0; // متغير لحساب مجموع عدد الأحرف المدخلة

    do
    {
        cout<<"enter the chars (to EXIT enter X):\n";
        cin>>ch;
        ch_number++;
        switch(ch)
        {
            case 'a':
                ch_a++; //a النوع من المدخلة من النوع
                break;
            case 'b':
                ch_b++; //b النوع من المدخلة من النوع
                break;
            case 'c':
                ch_c++; //c النوع من المدخلة من النوع
                break;

        }

        }while(ch!='x'); // شرط عملية التكرار
    cout<<endl<<endl;
    cout<<"this is number your chars:"<<ch_number++ -1<<endl; // إخراج المجموع الكلي لعدد أحرف المدخلة

    cout<<"this is number your chars A: "<<ch_a<<endl;
    cout<<"this is number your chars B: "<<ch_b<<endl;
    cout<<"this is number your chars C: "<<ch_c<<endl;

    return 0;
}

```

و هذا البرنامج في مرحلة التنفيذ:

```

C:\Documents and Settings\Administrator\III I
enter the chars (to EXIT enter X):
a
enter the chars (to EXIT enter X):
b
enter the chars (to EXIT enter X):
c
enter the chars (to EXIT enter X):
n
enter the chars (to EXIT enter X):
a
enter the chars (to EXIT enter X):
x

this is number your chars:5
this is number your chars A: 2
this is number your chars B: 1
this is number your chars C: 1
Press any key to continue.

```

**ملاحظات و أسئلة حول كود البرنامج:**

١ - تمعن في السطر التالي جيداً و أجب على السؤال

```
cout<<"this is number your chars:"<<ch_number++ -1<<endl;
```

لماذا قمنا في طرح ١ واحد من المجموع الكلي للأحرف المدخلة؟

(راجع نص المسألة)

٢ - (ملاحظة مهمة) لنفرض أننا قمنا في وضع المتغير المسؤل عن مجموع عدد

الأحرف المدخلة ch\_number++ -1 ضمن الجملة switch ماذا سوف يحدث و هل

سوف يعتبر المترجم ذلك خطأ أم لا ؟

بعد أن قمت في وضع المتغير ; ch\_number ++ - ١ ضمن الجملة switch و ذلك

على الشكل التالي:

```
do
{
cout<<"enter the chars (to EXIT enter X):\n";
cin>>ch;
    switch(ch)
    {
        case 'a':
            ch_a++;
            break;
        case 'b':
            ch_b++;
            break;
        case 'c':
            ch_c++;
            break;
        ch_number++;
    }
}while(ch!='x');
```

لم يخرج لي المترجم أي خطأ أبداً ، بل أخرج لي الرسالة التالية:

-----Configuration: namer\_c - Win32 Debug-----

Compiling...

c.cpp

Linking...

namer\_c.exe - 0 error(s), 0 warning(s)

مما يعني أن برنامجنا من الناحية القواعدية صحيح و خالي من أي خطأ error أو تحذير warning.

أما الآن سوف نقوم في مرحلة التنفيذ بالضغط على Ctrl+F5 و نقوم في إدخال عدد من الأحرف و من ثم سنشاهد النتيجة، و ذلك سيكون على الشكل التالي:

```

C:\> "C:\Documents and Settings\Administrator\III
enter the chars (to EXIT enter X):
a
enter the chars (to EXIT enter X):
d
enter the chars (to EXIT enter X):
b
enter the chars (to EXIT enter X):
c
enter the chars (to EXIT enter X):
x

this is number your chars:-1
this is number your chars A: 1
this is number your chars B: 1
this is number your chars C: 1
Press any key to continue_

```

قمت في إدخال أربع حروف: A مرة واحده و كذلك الأمر بالنسبة إلى B و C و D و مجموع الأحرف هو ٤ أربع أحرف،

ولكن ماذا أخرج لنا البرنامج بالنسبة إلى المجموع الكلي لعدد الأحرف المدخلة؟  
الجواب هو ١ - لماذا؟

لكي نجيب على هذا السؤال يجب علينا في البداية فهم مبدأ عمل الجمل و البنى المتداخلة، ففي برنامجنا السابق قمنا في استخدام الجملة switch و التي تقع بداخل جملة التكرار do/while.  
و قمت في رسم هذه الصورة للتعبير عن موقع switch بالنسبة إلى do/while.



و لاحظ بأن الجملة switch تقع ضمن الجملة do/while أي أن الجملة switch داخلية و do/while خارجية و في هذه الحالة لكل جملة حدود لمجال تعليماتها.

و لكي أوصل لك الفكرة التي أريد منك أن تفهما سوف أسمى الجملة الخارجية do/while بالأب، و الجملة switch بالأبن



و من المعروف بأن الأب يورث إلى ابنه بعض الصفات كالطول و لون الشعر... الخ و العكس غير صحيح فالابن لا يورث إلى أبيه صفاته ( و تقبل هذا الكلام مبدأً فقط . و سوف نتعرض إلى هذا الموضوع بالتفصيل إن شاء الله في فصل الوظائف ( Function ) .

و كذلك الأمر بالنسبة إلى مثالنا، فعندما قمنا في كتابة المتغير ++ch\_number داخل switch قام البرنامج على تنفيذ دور المتغير داخل switch (الابن) فقط! و لم يتم في تنفيذه داخل جملة التكرار do/while و كما هو معروف فإن المتغير ++ch\_number لا دور له داخل switch، مما أدى إلى إعطائه القيمة صفر أي لا شيء.

و لماذا كان الخرج على الشاشة لمجموع الأعداد الكلي يساوي - 1 ؟  
ذلك بسبب شرط نص المسألة (مجموع عدد الحروف - 1 الحرف x)  
 $0 - 1 = - 1$

### الخلاصة:

أن الأوامر و التعليمات الواقعة ضمن switch (الابن) فإنها تنفذ فقط ضمن حدود switch فقط و لا يتعدى ذلك إلى التعليمات و الأوامر الموجودة ضمن do/while (الأب) و العكس غير صحيح.

و كما قلت تقبل مني هذا الكلام كبداية فقط ، و هدفي من هذا الشرح كله هو إيصال فكرة عمل جملة داخل جملة و علاقة ذلك بالمتغيرات الواقعة ضمن تلك الجمل و سيكون لنا حديث مفصل إن شاء الله عن تمرير القيم بين الجمل و الوراثة في الفصول القادمة بمشيئة المولى عز و جل.

و نهاية أسأل الله عز و جل أن يكون قد وفقني في شرح الجملة do/while فإن كنت قد أحسنت في الشرح و وصول الفكرة إليك أخي الكريم، فإن هذا من فضل ربي، و إن لم أوفق في ذلك فهذا مما ذنوبي و مما كسبت نفسي نسأل الله العافية.

## التعليمة break :

هذه التعليمة ليست بالغريبة عليك أخي القارئ إن كنت قرأت الجملة switch. فهل تذكر تركيب الجملة switch و فروعها case فقد كان في نهاية كل case نضع التعليمة braek و ذلك للإنتهاء أو إيقاف عمل ذلك الفرع case بعد الإنتهاء من تنفيذ التعليمات الموجودة داخله.

و في الحقيقة هذه هي نفس التعليمة braek التي سوف نقوم في دراستها بإذن الله، و لكن الفرق هنا هو أنه سنعمم استعمال هذه الجملة على بقية الجمل التي سبق دراستها ( for,while,do/while ) و سنوضح بإذن الله كيفية استخدامها مع تلك الجمل.

تستخدم التعليمة break من أجل الخروج من الجملة الواقعة ضمنها فوراً ، لكي يتابع البرنامج بعد ذلك تنفيذ بقية الأوامر و التعليمات الواقعة بعدها مباشرة. و يمكننا القول بشكل آخر أنها تغير مجرى تنفيذ جمل اتخاذ القرار و جمل التكرار. لنأخذ بعض الأمثلة حتى تتكون لديك صورة و مفهوم أوسع حول هذه التعليمة.

### مثال :

لنفرض أنه يوجد في إحدى المشافي عدد من المرضى لا على التعيين في غرفة واحدة و في هذه الغرفة يوجد طبيب واحد، و من الصعب على هذا الطبيب مراقبة هؤلاء المرضى جميعاً في أن واحد.

أكتب برنامج لتحذير الطبيب بوجود خطر على حياة إحدى المرضى و ذلك بسبب ارتفاع درجة حرارته إلى 39 (و هذه الدرجة تعتبر خطرة على حياة المريض مما يستدعي مساعدة عاجلة من الطبيب لتخفيض درجة حرارة المريض) بحيث يكون هذا التحذير على شكل صوت تنبيه يخرج من جهاز الكمبيوتر. مع العلم أنه يتم قياس درجة حرارة المريض بفارق نصف درجة في كل مرحلة.

## الحل:

في البداية سوف نقوم في حل المسألة على الطريقة الخوارزمية ثم بعد ذلك نقوم بتحويلها إلى شيفرة برنامج C++

من خلال نص المسألة نضع النقاط المطلوبة منا عملها:

- ١ - برنامج يقوم في قياس درجة حرارة المريض بفارق 0.5 درجة كل مرحلة
- ٢ - عند وصول درجة حرارة المريض إلى 39 يجب على البرنامج اخراج رسالة تحذير على شكل صوت يخرج من الجهاز.
- ٣ - من المعلوم أن المريض يستطيع تحمل درجة حرارة تبدأ من ٣٦ إلى ٤٠ كحد أقصى

(أنا لست طبيباً لكي أعطيك درجات الحرارة التي يستطيع المرض تحملها كحد أدنى وأقصى، و لكنني قمت في استشارة بعض الأطباء في ذلك و أخبروني كشكل عام عن حدود درجة الحرارة التي يستطيع المريض تحملها، إن كنت تريد كتابة برنامج بشكل جدي فعليك استشارة الأطباء بذلك للأخذ بالمعلومات الدقيقة منهم )

من النقاط السابقة نكتب كود البرنامج و الذي سيكون على الشكل التالي:

```
# include <iostream.h>
int main()
{
    for(float t=36;t<=40;t+=0.5)
    {
        cout<<"the Temperature is: "<<t<<endl;

        if(t == 39)
        {
            cout<<"\n\n\t warning!!!!!!!!!!!!\a\a\a \n";
            cout<<"\t Temperature T = "<<t<<endl<<endl;
            break;
        }
    }
    return 0;
}
```

و بعد التحقق من عدم وجود أخطاء في كود البرنامج، نقوم في تنفيذ البرنامج وسيكون الخرج على الشاشة على الشكل التالي:

```

"C:\Documents and Settings\Administr
the Temperature is: 36
the Temperature is: 36.5
the Temperature is: 37
the Temperature is: 37.5
the Temperature is: 38
the Temperature is: 38.5
the Temperature is: 39

warning!!!!!!!!!!
Temperature T = 39

Press any key to continue

```

و لاحظ عمل التعليمة break داخل الجملة for، و ماذا أثرت هذه التعليمة على مجرى تنفيذ تعليمات for (مع ملاحظة نهاية شرط التكرار  $t \leq 40$  و مقارنة ذلك بالخرج على الشاشة )

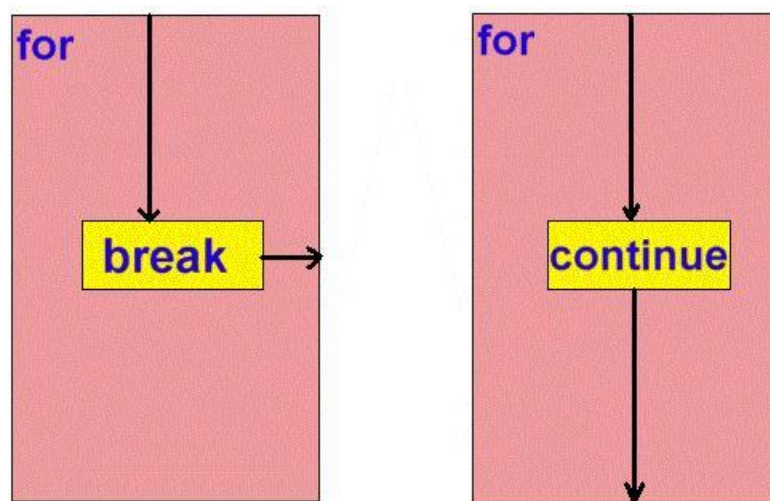
وسوف نقوم في استخدام التعليمة break في كثير من برامجنا التي سنكتبها لاحقا في المواضيع التي سنتحدث عنها في الفصول القادمة بإذن الله عز و جل.

و الخلاصة أنني أريد منك أخي الكريم أن تستوعب و تدرك عمل هذه التعليمة ضمن الجمل الأخرى ( for,switch,while,do/while ) و الارتباط بينها و بين if و لا تنسى أن خير معلم للغة C++ هو أنت، من خلال خبرتك و التي تعتمد على كثرة التمارين و الأمثلة التي تقوم في إيجادها أنت بنفسك ثم تقوم بحلها، وهذا الأمر يتطلب إلى جهد و وقت. و لا تنسى من سار على الدرب وصل.

## التعليمة `continue` :

التعليمة `continue` تشبه ألى حد كبير التعليمة `break` مع وجود بعض الاختلاف بينهما من حيث مبدأ العمل.

وجهة التشابه بين `break` و `continue` أن لهما نفس الموضع داخل الجمل ( `for,while,do/while` ) و لهما أيضاً نفس التأثير داخل هذه الجمل، إلا أن التعليمة `continue` تختلف عن `break` بأن الثانية تقوم بالخروج من الجملة فوراً ليتابع بعدها البرنامج تنفيذ الأوامر و التعليمات الواقعة بعد الجملة المتضمنة التعليمة `break` ، أما الأولى فتوقف عمل الجملة الواقعة ضمنها عند المكان الذي نحدده ثم تعود من جديد لتنفيذ بقية أوامر الجملة الواقعة ضمنها بعد النقطة التي توقفت عندها. هل تشابكت عليك الأمور لا تقلق سوف أقوم بتوضيح هذه الفكرة من خلال الأمثلة و لكن في البداية قمت في إنشاء شكل رسمي توضيحي لكل من التعليمة `break` و `continue` داخل الجملة `for` حتى يتكون لديك مفهوم أوسع حول كيفية مبدأ عملهما



و كما تلاحظ أخي الكريم تأثير التعليمة `break` داخل الجملة `for` .  
تلاحظ أن البرنامج توقف عن تنفيذ التكرار للجملة `for` عند التعليمة `break` ثم قام بالخروج من جسم الجملة `for` بعد ذلك فوراً .  
أما بالنسبة إلى التعليمة `continue` فتلاحظ أن البرنامج توقف عن تنفيذ التكرار في الجملة `for` عند التعليمة `continue` مؤقتاً ، ثم عاد من جديد بتنفيذ التكرار للجملة `for` .

و لنأخذ مثالين لكل من break و continue لنوضح الفرق بين عمل هاتين التعليميتين و المثال الأول سيكون توضيح لعمل التعليمة break.

### مثال ١:

أكتب برنامج يخرج على الشاشة الأعداد من ١ إلى ٥ بشكل تكرار تصاعدي و ذلك من خلال الجملة for، ثم بعد ذلك قم في إدخال التعليمة break في حال إذا كانت قيمة المتغير تساوي ٣، و لاحظ ذلك على تأثير مجرى الخرج على الشاشة.

### الحل:

```
# include <iostream.h>

int main()
{
    for(int x=0;x<=5;x++)
    {
        cout<<"this X without break = "<<x<<endl; //تنفيذ التكرار قبل break
        if(x==3)
        {
            break;
        }
        cout<<"this X with break = "<<x<<endl<<endl; // إخراج على الشاشة تنفيذ التكرار بعد break
    }

    return 0;
}
```

و الخرج على الشاشة سيكون على الشكل التالي:

```
"C:\Documents and Settings\Admini...
this X without break = 0
this X with break = 0

this X without break = 1
this X with break = 1

this X without break = 2
this X with break = 2

this X without break = 3
Press any key to continue.
```

**مثال ٢:**

أكتب برنامج يخرج على الشاشة الأعداد من ١ إلى ٥ بشكل تكرار تصاعدي و ذلك من خلال الجملة for، ثم بعد ذلك قم في إدخال التعليمة continue في حال إذا كانت قيمة المتغير تساوي ٣، و لاحظ ذلك على تأثير مجرى الخرج على الشاشة.

**الحل:**

```
#include <iostream.h>

int main()
{
    for(int x=0;x<=5;x++)
    {
        cout<<"this X without continue = "<<x<<endl; //تنفيذ التكرار قبل continue
        if(x==3)
        {
            continue;
        }
        cout<<"this X with continue = "<<x<<endl<<endl; // إخراج على الشاشة تنفيذ التكرار بعد continue
    }
    return 0;
}
```

و الخرج على الشاشة سيكون على الشكل التالي:

```
C:\Documents and Settings\Adminis
this X without continue = 0
this X with continue = 0

this X without continue = 1
this X with continue = 1

this X without continue = 2
this X with continue = 2

this X without continue = 3
this X without continue = 4
this X with continue = 4

this X without continue = 5
this X with continue = 5

Press any key to continue
```



و لاحظ معي أخي الكريم الخرج على الشاشة للمثال الأول ( break ) حيث بدأ البرنامج في تنفيذ حلقت التكرار للجملة for من العدد ١ و حتى و صول قيمة المتغير إلى العدد ٣

```
If(x == 3)
{
    break;
}
```

مما أدى إلى الخروج فوراً من حلقة التكرار، و لم يتم تنفيذ بقية حلقة التكرار مع العلم أن شرط عملية التكرار  $x \leq 5$ .

أما في المثال الثاني فالخرج على الشاشة بدأ من العدد ١ وصولاً إلى القيمة ٣ و لكن هذه القيمة ٣ لم تظهر على الشاشة، و ذلك بسبب التعليمة continue

```
If( x == 3)
{
    continue;
}
```

حيث قام البرنامج بتخطي هذه القيمة و من ثم تابع بعد ذلك عملية التكرار إلى أن وصل عداد التكرار إلى ٥ أي قيمة نهاية شرط التكرار. و أرجوا من الله عز و جل أن أكون قد وفقت في شرح هاتين التعليمتين و في إيصال الفكرة إليكم و في تكوين فكرة عامة و مفهومة حول كيفية استخدامهم.

يتبع في هذا الكتاب إن شاء الله:

الجزء الثالث:

الوظائف (التوابع) Function.

الجزء الرابع:

المصفوفات array .

الجزء الخامس:

المؤشرات points .

الجزء السادس:

الصفوف classes .

لا تنسوني من دعوة أخ في ظهر الغيب

مع تحيات: نمر مذيّب النصيرات

بيلاروسيا – ماجيليف

هاتف: ٠٠٣٧٥٢٩٧٤٨٦١٦٨

٠٠٩٧١٥٠٥٨٣١٧٤١

[namer315@yahoo.com](mailto:namer315@yahoo.com)

تم بعون الله في:

٢٠٠٥/٣/٢١ م