

السلام عليكم ورحمة الله وبركاته.....
الحمد لله والصلاة والسلام على افضل الأنبياء والمرسلين قدوتنا وحبیبنا وشفیعنا ان شاء الله محمد بن عبد الله
وعلى اله وصحبه اجمعين.....

نبدا ان شاء الله تعالى سلسله دروس لغة من اهم وأروع لغات البرمجه الا وهي لغة ++C, في هذه الدورة ان شاء الله سنبدأ في شرح هذه الدوره من الصفر حتى يتسنى للجميع ان شاء الله فهم هذه اللغه ببسر وسهوله وسنركز في شرحنا على الأمثله العمليه لأن خير طريقة لتعلم هذه اللغه في وجهة هو تعلمها من خلال الأمثله العمليه ومحاولة تطبيقها على البرنامج وتجربتها. وتستطيع استخدام برنامج Borland ++C أو برنامج [Visual C++ 6.0](#), وسنتطرف بأذن الله الى شرح كيفية استخدام هذين البرنامجين ليتسنى لك العمل عليهم وسنتوسع في شرح برنامج [Visual C++ 6.0](#) بما انه الأفضل والأكثر انتشاراً. والذي يجيد لغة ال C او لديه خلفيه عنها سيسهل عليه جدا تعلم لغة ++C فهما متشابهتان تقريبا. وفي الحقيقة هما من اهم اللغات وبالنسبة لي أعتبرهما المدخل لتعلم لغات البرمجة فعندما تتقن تعلم هاتين اللغتين او واحده منهما تكون قد اكتسبت أساسا قويا في البرمجة ويسهل عليك الانطلاق بعدها في عالم البرمجه. وهما اول لغتين تعلمتهما من لغات البرمجه, وسهل علي بفضل الله تعلم لغات برمجة اخرى.

الدرس الأول:

بسم الله نبدا أول دروسنا على بركة الله وكما اتفقنا سنركز على فهم هذه اللغه عن طريق الأمثله العمليه...
وسنبدا بأول برنامج تراه في عالم ال C وال ++C .

```
1. #include <iostream>
2. int main() // function main begins program execution
3. {
4.     std::cout << "Welcome to C++!\n";
5.     return 0; // indicate that program ended successfully
6. }
```

شرح البرنامج:

السطر الأول:

وهذا السطر اساسي جدا في أغلب البرامج ويجب أن تعود نفسك على كتابته عند البدء في كتابة البرنامج , ويعني ان برنامجك يستخدم مكتبة `iostream`. وفي لغة ++C العديد من المكتبات المستخدمه سننظر اليها في مرحلة متقدمه ان شاء الله نستخدم كل مكتبة على حسب احتياجنا لها في البرنامج ومكتبة ال `iostream` مسؤلة عن عمليتان الإدخال والأخراج في ال ++C . وهي اختصار ل `input output stream` ويتملان هتين العمليتين في `cin` للإدخال (أي أدخل المعلومه او تخزين القيمه في متغير بأسم نوع تقوم بتحديدته للبرنامج), وال `cout` للأخراج (أي اخراج المعلومه أو ما نريد كتابته على الشاشة). خلاصة هذا السطر انه يقوم بتوجيه المترجم ليقوم بجعل البرنامج الذي تقوم بكتابته يحتوي على مكتبة `iostream`.

السطر الثاني:

ملاحظه مهمه: ستلاحظ أحتواء السطر الثاني على **// function main begins program execution** هذه العبارة ليس لها علاقة بالبرنامج ,تستخدم كهامش للتبويه اي ان اي شيء بعد // يستخدم كتوضيح يكتبه المبرمج في برنامج ولا ينظر له المترجم عند تنفيذ أوامر البرنامج او عند تنفيذ أسطر البرنامج فجميع الأسطر او العبارات بعد // وقد لونها باللون الحمر ليسهل عليك تمييزها لا يضع لها المبرمج اعتبار ولا يراها أصلا عند تنفيذ اسطر البرنامج,هي فقط يكتبها المبرمج لتوضيح معني السطر او كتابة ملاحظه تذكيري له وليس بالضرورة كتابة هذه الملاحظات فيمكنك الاستغناء عن العبارات الي باللون الحمر في البرنامج التي تأتي بعد // . فمثلا في السطر الثاني ما بعد ال // وهو عبارة **function main begins program execution** مجرد ملاحظة لتعطي قارئ البرنامج تنويه عن وظيفة **int main()** .

هذا السطر يمثل الدالة الرئيسي للبرنامج وجميع البرامج تحتوي على دالة ال **main** , فجميع العمليات التي سيقوم بها البرنامج الذي ستقوم بكتابته يجب ان يكون داخل هذه الدالة التي يجب ان تبدأ بقوس **{** كما في السطر الثالث وتنتهيها بقوس **}** كما في السطر السادس ومحتوى برنامجك يكون بين هذين القوسين.

السطر الرابع:

في هذا السطر سترى ان هذا السطر يحتوي في بدايته على **std::** طبعا هذه الجملة نقوم بكتابتها قبل ال **cin** وال **cout** وتستطيع تعريفهما قبل ال **main** وهذا الأفضل لك لا تحتاج لتعريفهما عند كل استخدام لل **cin** وال **cout** وسيكون البرنامج هكذا.

```
1. #include <iostream>
2. using std::cout; // program uses cout
3. int main() // function main begins program execution
4. {
5.     cout << "Welcome to C++!\n";
6.     return 0; // indicate that program ended successfully
7. }
```

ولك ان تلاحظ الأختلاف بين البرنامجين وأنصحك باستخدام الأسلوب الثاني الأكثر اختصارا . نعود لأستكمال شرحنا ولاحظ اننا نشرح البرنامج بالترتيب الأول اي البرنامج الذي كتبناه في بداية شرحنا . في السطر الرابع قمنا باستخدام ال **cout<<** ونستخدم هذا الخدمة اتي توفرها مكتبة ال **iostream** كما اسلفنا عند طباعة معلومات او قيم على الشاشة وتمكننا هذه العملية من اخراج المعلومات على الشاشة وتكون جملة **cout** متبوعه بمعاملين << وبعدهما يجب وضع المعلومات المراد أخراجها على الشاشة بين "" أما اذا كنا نريد ان نظهر قيمة متغير نكتب أسم المتغير بدون "" لأنه عند وضع اي شيء بين ال "" يتم طباعته كما هو فمثلا عند كتابة السطر **cout<<"salam 3alekom"** يقوم البرنامج بأظهار على الشاشة **salam 3alekom** , ومعنى **\n** يعني يقوم البرنامج بالنزول الى سطر جديد بعد كتابة **!Welcome to C++** على الشاشة ونلاحظ ان برنامجنا سيقوم بكتابة **!Welcome to C++** وهذه هي العملية التي كتب من اجلها هذا البرنامج البسيط . وأخيرا نهي السطر بعلامة الفاصلة المنقوطة; ويجب انهاء كل سطر بعلامة الفاصلة المنقوطة فهي تعطي المترجم معنى بأن السطر قد انتهى وانتهت الأوامر لهذا السطر فعليه الذهاب للسطر الثاني وتنفيذ الأوامر فيه .

السطر الخامس:

هذا السطر يحتوي على `return 0;` وهذا السطر يجب كتابته عند نهاية دالة ال `main` فهو يخبرنا بان البرنامج نفذ بنجاح وتستطيع استبداله بكتابة `void main()` بدل `int main()` فعند كتابة `void main` لست بحاجة لكتابة `return 0;` قبل غلق دالة ال `main` بـ `}`.
ملاحظة: لاحظ ان `return 0;` تحتوي على صفر وليس حرف ال `o`، فمن أكثر الأخطاء شيوعا كتابة حرف ال `o` بدل الصفر.

وصلنا اخوتي وأخواتي في الله الى نهاية الدرس الأول الذي أسأل الله ان يكون واضحا ويفترض ان تكون قد فهمت من خلال هذا البرنامج بعض الأساسيات في ال `C++` وسنواصل في بقية الدروس شرح برامج أكثر تطورا وشرح مفاهيم واساسيات اعرق فهذا الدرس عبارته عن مقدمة تدخلك الى عالم `C++`.
وفي نهاية الدرس نقول هذا اجتهدنا فأنا اخطانا فمن انفسنا وان اصبنا فمن الله سبحانه وتعالى.....

°~*α@§(*§سبحانك اللهم وبحمدك أشهد ان لا اله الا انت استغفرك وأتوب إليك *§~*α@°

°~*α!|||α*!~`((كاتبة الدرس بنت النور..... عفوا يمنع اجراء اي تعديل للملف))°~*α!|||α*!~`

°~*α!|||α*!~`((نسألکم دعوة صادقة لأختکم في ظهر الغيب))°~*α!|||α*!~`

السلام عليكم ورحمة الله وبركاته.....
الحمد لله والصلاة والسلام على افضل الأنبياء والمرسلين قدوتنا وحبیبنا وشفیعنا ان شاء الله محمد بن عبد الله
وعلى اله وصحبه اجمعين.....

نواصل في هذا الموضوع الدرس الثاني في دورة لغة الـ C++ وبعد ان اخذنا فكرة ارجو ان تكون كمقدمة جيدة
ووافيه للدخول في تفاصيل هذه اللغة.

في هذا الدرس سوف نقوم بالتعرف على المتغيرات (Variables) , تسمية المتغيرات , انماط المتغيرات وانواعها
, الثوابت, العمليات الحسابية (Arithmetic Operation), عمليات المقارنة (relation operator).

١- المتغيرات (Variables) وتسميتها.

تتكون ذاكرة الحاسوب من خانات اما ان يكون في الخانه صفر او واحد , يترجم كل شيء الى لغة
الباينري (binary) وهذه اللغة عباره عن صفر ١ واحد ٠١٠١٠١٠١ والذاكره تنقسم الى bytes وكل
bytes=8bit وال bit =one or zero . فعندما تريد ان تدخل قيمة بالذاكره وتحفظها يجب عليك ان
تحجز لها مساحة في الذاكره وتسميها وتعرف نوعها ومساحتها وهذا ما يجب القيام به في عملية
الأدخال التي شرحناها في الدرس السابق.

مثال: int box;

Cin >>box;

لا حظ هنا البرنامج ليس كاملا فقط قمت بأقتطاع ما نحتاجه لفهم جزئية المتغيرات وتسميتها, نلاحظ في
البداية قمنا بتعريف نوع المتغير الذي اسمناه box ونوعه int يعني يحتوي على ارقام صحيحه
وسندرج كل الانواع وشرح لها وهنا لم نحدد قيمته لننا سندخلها عن طريق ال cin ولو اننا حددنا
قيمتها لعملنا 5 = int box فبهذه الطريقه قمنا بحجز مساحه اسمها box ونوعها int يعني تحوي
ارقام صحيحه وقيمتها خمسه هنا ثبتنا القيمه على خمسه لكن في الأعلى نحن لم نحدد القيمه وقمنا
بأدخال القيمه فالقيمه التي سيدخلها المستخدم ستوضع في المساحه المحجوزة بأسم box.

**ملاحظه: عند تسمية المتغير لا بأس باستخدام الحروف والأرقام وعلامة \$, @, ويمنع استخدام \$, @,
والرموز التي على هذه الشاكلة.**

وسندرج أكثر الأنواع استخداما ومعناها..

النوع	ملاحظات
int	يعني المساحه المحجوزة تحوي ارقام صحيحه. ويمكنك هنا استخدام: long int box; أو short int box; أو long box أو short box ;
float	يعني المساحه المحجوزة تحوي أرقام عشرية مثال ٣,٥, ١,٢
double	يحوي أعداد كبيره.
char	يحتوي على حروف ورموز ولا بأس ان يحوي أرقام.
bool	تحوي true او false

٢- الثوابت (constants):

عندما تريد حجز مساحة لمتغير بنوع محدد وأسم تختاره انت وتريد ان تجعل هذه القيمة ثابتة , فعليك بأضافة const قبل تعريف المتغير فيصبح: `const int box =5;` فهنا تكون قيمة المتغير box تساوي خمسة دائما.

٣- العمليات الحسابية (Arithmetic Operation)

١- عملية الجمع (+).

مثال : `sum=num1 + num2;`

٢- عملية الطرح (-).

مثال : `sum=num1 - num2;`

٣- عملية الضرب (*).

مثال : `sum=num1 * num2;`

٤- عملية باقي القسمة (%).

مثال : `sum=num1 % num2;`

٥- عملية القسمة (/).

مثال : `sum=num1/num2;`

٣- عمليات المقارنه (relation operator).

١- (==) وهذه لمقارنة المساواة .

مثال : اذا قلنا `num 1 == num2` نحن نقصد هنا ان `num1` يساوي `num2` وانتبه ان تخط بين هذه العمليه واشاره = فأشاره ال = تعني ان القيمة التي باليمين تنتقل الى القيمة التي باليسار
مثلا: `num1=num2` يعني قيمة `num2` تنسخ في `num1` .

٢- (!=) يعني لا تساوي.

مثال : `num1 != num2` يعني `num1` لا يساوي `num2` .

٣- (>)

مثال : `num1>num2` يعني `num1` أكبر من `num2` .

٤- (<)

مثال : `num1<num2` يعني `num1` أصغر من `num2` .

٥- (>=)

مثال : `num1 >= num2` يعني `num1` أكبر من أو يساوي `num2` .

٦- (<=)

مثال : `num1 <= num2` يعني `num1` أصغر من أو يساوي `num2` .

وهنا نصل الى نهاية درسنا اليوم ولكن قبل ان ننهي اليكم هذا السؤال لتختبروا مدى فهمكم لما شرحنا وسنقوم بحله ان شاء الله بداية الدرس القادم... وساقوم دائما بكتابة السؤال بالعربية والانكليزية.

السؤال: كتابة برنامج يقوم بجمع وطرح وقسمة عددين صحيحين وكتابة الناتج للعمليات الثلاث على الشاشة كنتاج للبرنامج.

Questions: write a program which add, subtract and divide two integer numbers and show the result for three arithmetic operations as an output for the program.

°~*α@§(*§(أنت استغفرك وأتوب إليك *§)°~*α@§

`~*α!||!α*!~`((كاتبة الدرس بنت النور..... عفوا يمنع اجراء اي تعديل للملف))`~*α!||!α*!~`

`~*α!||!α*!~`((نسألكم دعوة صادقة لأختكم في ظهر الغيب))`~*α!||!α*!~`

السلام عليكم ورحمة الله وبركاته.....
الحمد لله والصلاة والسلام على افضل الأنبياء والمرسلين قدوتنا وحبیبنا وشفیعنا ان شاء الله محمد بن عبد الله
وعلى اله وصحبه اجمعين.....

نبدأ بأذن الله الدرس الثالث وقد انتهينا في الدرس السابق بطرح سؤال في آخر الدرس وسنبدا هذا الدرس بالأجابة
عن السؤال السابق وسنطرح اسأله أخرى والأجابة عنها ان شاء الله، ملاحظه لم انوه لها في الدرس السابق لاحظ
عند تسمية المتغيرات يجب مراعاة أن التسميه تكون حساسه للحروف الصغيره والكبيره يعني يجب الانتباه عند
كتابة الأسم في البرنامج ان يكون نفس الأسم بالضبط دون التغيير في شيء.
حاول ان تجيب على الأسئلة المرفقه بنفسك أولا ثم بعدها طابق حلك بالحل الذي وضعته حتى تستفيد لأن
البرمج لا يمكنك تعلمها من غير الممارسه وحل الأسئلة فبمجرد القراءة لا يمكنك ان تكون مبرمج.

**السؤال 1: كتابة برنامج يقوم بجمع وطرح وقسمة عددين صحيحين وكتابة الناتج للعمليات الثلاث على
الشاشة كنتاج للبرنامج.**

**Questions1: write a program which add, subtract and divide two integer numbers
and show the result for three arithmetic operations as an output for the program.**

الأجابة:

**ملاحظه: سنقوم بكتابة البرنامج وتوضيح الفكرة وقد أسلفنا في الدرس الاول شرح تفصيلي للأساسيات فأن
كنت قد نسيتها فأرجع للدرس الأول ثم واصل معنا هنا.**

```
#include <iostream>

int main()
{
    int integer1; // أسم المساحة لتخزين الرقم الأول
    int integer2; // أسم المساحة لتخزين الرقم الثاني
    int sum,sub; // أسم المساحة التي سنخزن فيها ناتج الجمع والطرح على التوالي
    float div; // جعلنا النوع له لعداد عشريه لن القسمة لا تعطي دائما اعداد صحيحه

    std::cout << "Enter first integer\n"; // لأعطاء المستخدم رسالة لأدخال رقم
    std::cin >> integer1; // لأدخال الرقم في المساحة المحجوزة

    std::cout << "Enter second integer\n"; // لأعطاء المستخدم رسالة لأدخال رقم
    std::cin >> integer2; // لأدخال الرقم في المساحة المحجوزة

    sum = integer1 + integer2; // للقيام بعملية الجمع
    sub = integer1 - integer2; // للقيام بعملية الطرح
    div = integer1 / integer2; // للقيام بعملية القسمة

    std::cout << "Sum is " << sum << std::endl; // لطباعة ناتج الجمع
    std::cout << "sub is " << sub << std::endl; // لطباعة ناتج الطرح
    std::cout << "div is " << div << std::endl; // لطباعة ناتج القسمة

    return 0; // indicate that program ended successfully
} // end function main
```

ملاحظه: الذى اللون الأزرق يمثل الكود وما سواه مجرد ملاحظات
للتوضيح.....

الناتج على الشاشة سيكون(output):
Enter first integer
٤
Enter second integer
٥
Sum is ٩
sub is -1
div is 0.8

سؤال آخر: وسأقوم بكتابته بالعربية والإنكليزية

Questions2 : Write a C++program to read two integers (first and second), and two real numbers (third and fourth), then calculate the value of:

$$\text{Result1} = 7 \text{ first}^3 + \text{second}^2$$

$$\text{Result2} = \frac{(\text{third}^2 - \text{fourth})^{(1/2)}}{\text{third} - \text{fourth}}$$

Your program should print the values of Result1 and Result2.

Note: 2^3 means the number 2 is raised to the power of 3.

- السؤال ٢: المطلوب كتابة برنامج بلغة ال C++ يقوم بأدخال عدنان صحيحان وآخرين عشريين ويجب ان يقوم البرنامج بتطبيق المعادله الاولى على العددين الصحيحين وتطبيق المعادلة الثانيه على العددين العشريين والمعادلتين كالتالي:

$$\text{Result1} = 7 \text{ first}^3 + \text{second}^2$$

$$\text{Result2} = \frac{(\text{third}^2 - \text{fourth})^{(1/2)}}{\text{third} - \text{fourth}}$$

- ملاحظه: يجب ان يقوم البرنامج بطباعة الناتج الأول والثاني على الشاشة , و 2^3 تعني الرقم ٢ مرفوع للقوة ٣ او ٢ أس ٣ ويجب عليك تحويل هذه العبارة الى لغة السي بلس

Solution:

```
#include <iostream>
#include <math.h>

void main ()
{
    int first, second, Result1;
    float third, fourth, Result2;

    cout<<"enter the first number"<<endl;
    cin>>first;

    cout<<"enter the second number"<<endl;
    cin>>second;

    Result1 = 7* pow(first,3) + pow(second,2);

    Cout<<"the result1 is"<<Result1<<endl;

    Cout<<"enter the third number"<<endl;
    Cin>>third;

    Cout<<"enter the fourth number"<<endl;
    Cin>>fourth;

    Result2 = pow(pow(third,2) - fourth,1/2)/ (third - fourth);
    Cout<<"the result2 is"<<Result2<<endl;

}
```

[#include <math.h> ملاحظه: استخدمنا في البداية](#)
[لأننا أستخدمنا pow وهو ضمن هذه المكتبة فوجب تعريفها في البداية](#)

[ولمعرفة المزيد عن المكتبات المستخدمة في السي بلس بلس ودوالها انصحكم بزيارة](#)
[/http://www.cplusplus.com](http://www.cplusplus.com) [هذا الموقع الرائع للسي بلس بلس](#)

Question 3: write a program in C++ that accept a Fahrenheit temperature and output the equivalent Celsius temperature.

And the equation as follow:

Celsius=9/5 (Fahrenheit-32)

السؤال ٣: المطلوب كتابة برنامج باستخدام لغة ال C++ يقوم بأخذ درجة الحرارة بالفهرينهايت وأعطاء درجة الحرارة بالسيلسيوس كنتاج للبرنامج وطبعاً المعادله المستخدمه للتحويل من الفهرينهايت للسيلسيوس كالتالي:
 $Celsius = 9/5 (Fahrenheit - 32)$

Solution:

```
#include <iostream>

void main ()
{
    float Celsius, Fahrenheit;

    cout<<"please enter the temperature in Fahrenheit\n ";
    cin>>Fahrenheit;

    Celsius=9/5*(Fahrenheit-32)

    Cout<<" the temperature in Celsius is "<<Celsius<<endl;

}
```

Question 4:write a complete C program that ask the user to enter three integer number and store them in three variable k,L,and M.The program should find the value of N according to the following formula.
 $N=K+L-3M$.

SOLUTION:

```
#include <iostream>

void main ()
{
    int K,L,M,N;

    cout<<"please enter the value of K"<<endl;
    cin>>K;
    cout<<"please enter the value of L"<<endl;
    cin>>L;
    cout<<"please enter the value of M"<<endl;
    cin>>M;

    N=K+L-3*M;

    cout<<"the value of N is"<<N<<endl;
}
```

هنا نصل لنهاية درس اليوم اتمنى ان يكون واضح ومفيد وسننتقل باذن الله في
الدرس القادم الى مرحلة اكثر تطورا في ال C++ وأي استفسار يسرني الاجابه
عليه....

°~*α@§(*§)§@α*~°

°~*α!||!α*~°((كاتبه درس.....بنت النور.....عفا يمنع اجراء اي تعديل للملف))

°~*α!||!α*~°((نسألکم دعوة صادقة لأختکم في ظهر الغيب))

السلام عليكم ورحمة الله وبركاته.....
الحمد لله والصلاة والسلام على افضل الأنبياء والمرسلين قدوتنا وحبیبنا وشفیعنا ان شاء الله محمد بن عبد الله
وعلى اله وصحبه اجمعين.....

لقد انهينا التعرف على لغة ال C++ وكتابة برامج بسيطة بهذه اللغة القوية ولكن هل تعتقد ان يمكنك مواصلة
البرمجة بهذه المبادئ البسيطة؟ طبعا لا فقد يطلب منك كتابة برنامج لأدخال بيانات عدد كبير من الموظفين
وتذكر ان المبرمج الناجح يكتب البرنامج الذي ينفذ المهمة بأقل سطور ممكنه لك يكون برنامج سريع عند
التنفيذ، طبعا من الغير عملي جدا ان تدخل بيانات ١٠٠ موظف مثلا ب ١٠٠ cout و ١٠٠ cin . هنا تأتي
ضرورة استخدام هذه الجمل المفيدة التي سنتطرق اليها في دورتنا.

اليوم بأذن الله سنبدأ في التعرف على جمل مفيدة في السي ++ وهي التي ستقودنا الى بداية برمجة جيدة.....
سنتناول في هذه الدورة :

- if ,if/else, else if
- Switch
- do while
- while
- for
- break ,continue
- function
- arrays

في الدرس الرابع سنتناول شرح if ,if/else, else if :-

if (expression) {

do the code inside the two brackets;

}

نبدأ بال if :- كما ترى هذه الصيغة العامة لكتابة جملة ال if , يحتوي القوس الذي يأتي بعد ال if على
الشرط الذي بموجبه يتم تنفيذ العمليات التي بداخل القوسين { } , فعند انطباق الشرط يتم تنفيذ ما بداخل
القوسين { } ومن ثم الانتقال الى العمليات التالية ان وجدت ونقصد بالعمليات التالية العمليات التي تأتي
بعد تنفيذ العمليات التي بداخل قوسين ال if, اما اذا لم يتحقق الشرط فلا ينظر لا بداخل ال if وينتقل
مباشرة الى العمليات التالية التي تكون خارج نطاق ال if . ومنتقل للمثال التالي لتتمكن من فهم الموضوع
بشكل أكبر.

السؤال: قم بكتابة برنامج بلغة ال C++ يقوم بأدخال رقمين وكتابة العلاقة بينهما . فمثلا يقوم البرنامج
بكتابة اذا ما كانا الرقمين متساويين واذا لم يكونا يقوم بكتابة ذلك وتوضيح اي العددين اكبر .

QUESTIONS: write program using C++ that accept two number and give
the relationships they satisfy.

Answer:-

```
1. #include <iostream>

2. using std::cout; // program uses cout
3. using std::cin; // program uses cin
4. using std::endl; // program uses endl

5. int main()
6. {
7.     int num1; // first number to be read from user
8.     int num2; // second number to be read from user

9.     cout << "Enter two integers, and I will tell you\n"
10.         << "the relationships they satisfy: ";

11.     cin >> num1 >> num2; // read two integers

12.     if ( num1 == num2 )
13.         cout << num1 << " is equal to " << num2 << endl;

14.     if ( num1 != num2 )
15.         cout << num1 << " is not equal to " << num2 << endl;

16.     if ( num1 < num2 )
17.         cout << num1 << " is less than " << num2 << endl;

18.     if ( num1 > num2 )
19.         cout << num1 << " is greater than " << num2 << endl;

20.     if ( num1 <= num2 )
21.         cout << num1 << " is less than or equal to "
22.             << num2 << endl;

23.     if ( num1 >= num2 )
24.         cout << num1 << " is greater than or equal to "
25.             << num2 << endl;

26.     return 0; // indicate that program ended successfully

27. }
```

سنبدأ بالشرح من السطر ال ١٢ وذلك لأننا قمنا بشرح السطور من ١ الى ١١ بالتفصيل الممل في الدرس السابق. في السطر الثاني عشر (`if (num1 == num2)` ومعناها اذا تحقق ان `num1==num2` نقوم بتطبيق السطر ١٣ وأكد انكم لاحظتم عدم وجود القوسين `{ }` وهذا لأنه اذا كان لدينا فقط سطر واحد في داخل ال `if` لا داعي لوضع هذين القوسين و لا مشكلة اذا وضعتهما اما اذا كان لديك اكثر من سطر وجب عليك وضعهما ,لأن المترجم اذا احتوت ال `if` اكثر من سطر ولم تضعهما داخل القوسين `{ }` سيقوم بأخذ اول سطر ضمن ال `if` والآخرين يحسبهما خارج ال `if` ويتعامل معهما على هذه الأساس.

وبقية السطور في البرنامج السابق تقارن بالمثل ولكن باختلاف وجه المقارنه .

هكذا سيكون الناتج للبرنامج بعد تجريبه (الناتج على الشاشة):

Enter two integers, and I will tell you
the relationships they satisfy:

4

5

ξ is not equal to 5

ξ is less than 5

ξ is less than or equal to 5

أرجو ان تكون قد وضحت الصورة بالنسبة لل `if/else` .وننتقل لل

ثانيا: `if/else`

```
if (expression) {  
code inside if;  
}
```

```
else {  
code inside else;  
}
```

نلاحظ في السابق عند استخدام ال `if` بمفردها اذا لم يتحقق الشرط يتجاوز المترجم ال `if` وما بداخلها من كود ويتابع طريقه اما في حالة وجود ال `else` فهنا كما ترون بالشكل العام لل `if/else` يمر المترجم بال `if` يرى الشرط اذا تحقق الشرط يطبق ما بداخل ال `if` واذا لم يتحقق يطبق ما بداخل ال `else` قد تسألون اذا ما الفرق بينه وبين ما سبق شرحه في ال `if` طبعا الفرق بهذا المثال :برنامج يطلب من المستخدم ان يدخل رقما, اذا ادخل المستخدم رقم 3 نطبع على الشاشة `hello` غير ذلك نطبع `error`. نلاحظ هنا اننا سنجعل `if` للمقارنه اذا كان الرقم الذي ادخله المستخدم 3 لنطبع `hello` , غير ذلك نطبع `error` وسيكون البرنامج كالتالي....

```
#include <iostream>  
  
using std::cout; // program uses cout  
using std::cin; // program uses cin  
  
// function main begins program execution  
int main()  
{  
    int num1; // first number to be read from user  
  
    cout << "Enter the integer number \n";  
    cin >> num1; // read the integer  
  
    if ( num1 == 3)  
        cout<<"hello";  
  
    else  
        cout<<"error";  
  
    return 0; // indicate that program ended successful  
} // end function main
```

نلاحظ في البرنامج السابق إذا ادخل المستخدم ٣ سيطلع hello اما إذا ادخل اي رقم آخر مهما كانت قيمته وهذا ما قصدناه ب else سيطلع error . أرجو ان يكون المعنى واضح.

ثالثًا: else if

هذه الجملة مفيدة جدا فهي تجعل برنامجك اكثر نكاء وحنكه والشكل العام لهذه الجملة كالتالي:-

```
if (expression) {  
do the code inside if;  
}
```

```
else if (expression) {  
  
do the code inside else if;  
}
```

```
else if (expression) {  
do the code inside else if;  
}
```

```
else {  
do the code inside else;  
}
```

هنا المترجم سيقوم بتأكد من القاعدة التي بداخل ال if اذا تطبقت نفذها ولا يمر على باقي الخيارات المتمثلة في ال if else وال else , إذا لم تطبق القاعدة يذهب للخيار الآخر في ال else if واذا لم يطبق يذهب للذي بعده واخيرا اذا لم يطبق شيء ينفذ ال else. اي انه اذا طبق احد الخيارات يخرج من هذه المقارنه وطبعا هنا يجب ان تلاحظ لو انك استبدلت ال else if ب if لثم المرور على كل ال if والتأكد من قاعدتها وتطبيقها اذا تحققت القاعدة لكن في حالة ال else if يمر على ال if الاولى اذا تحققت يتجاوز الجميع ويخرج لما وراء ال else اما اذا لم تتحقق يجرب البقية وبمجرد تحقق احدهم يخرج لذلك يكون البرنامج اسرع بال else if فلا داعي لمقارنة البقية مادام تحقق شرط من الشروط والمثال التالي لتوضيح الموضوع أكثر.

لاحظ في هذا البرنامج يمر المترجم على ال (`if (num1 > num2)`) اذا تحقق الشرط يكتب العبارة `The number1 is bigger than number2` وينتقل الى ما بعد ال else اي لا يقارن المقارنات الأخرى، اما اذا لم يتحقق ينتقل الى ال (`else if (num2 > num1)`) ويطبق عليه نفس ما قلناه في الأولى وهكذا اذا لم تتحقق القاعدة فيهم جميعا نصل الى ال else وننفذ ما بها ويطبع error .

```
1- #include <iostream>  
2- int main()  
3- {  
4- int num1,num2;  
5- cout<<"please enter two integer numbers";  
6- cin >> num1 >> num2 ;
```

```

7- if (num1 > num2 ) {
8- cout << "The number1 is bigger than number2" ;
9- }

10- else if (num2 > num1) {
11- cout << "The number2 is bigger than number1" ;
12- }

13- else if ( num2=num1) {
14- cout << "there is no bigger number" ;}

15- else { cout << "error" ; }
16- return 0;
17- }

```

هنا نصل لنهاية درس اليوم اتمنى ان يكون واضح ومفيد وسننتقل باذن الله في
 الدرس القادم الى مرحلة اكثر تطورا في ال C++ وسنقوم بمجل تمارين على درس
 اليوم وأي استفسار يسرني الاجابه عليه....

°~*α@\$(*)\$سبحانك اللهم وبحمدك أشهد ان لا اله الا انت استغفرک وأتوب إليك *)\$@α*~°

`~*α!||!α*!~`((كاتبة الدرس.....بنت النور.....عفوا يمنع اجراء اي تعديل للملف))

`~*α!||!α*!~`((نسألکم دعوة صادقة لأختکم في ظهر الغيب))

السلام عليكم ورحمة الله وبركاته.....
الحمد لله والصلاة والسلام على افضل الأنبياء والمرسلين قدوتنا وحبیبنا وشفیعنا ان شاء الله محمد بن عبد الله
وعلى اله وصحبه اجمعين.....

اتمنى ان تكونوا بصح جيدة.....بعد ان انتهينا من الدرس الرابع نأخذ هذه التمارين لنعزز ما تم دراسته وبعدها
نتنقل للدرس الخامس.....

Question1: write a program that reads an integer and determine whether it is odd or even. (Hint: use the modulus operator .an even number is a multiple of two. any multiple of two leaves a remainder of zero when divide by 2).

السؤال الأول: قم بكتابة برنامج باستخدام لغة ال C++ يقوم بأدخال عدد صحيح وتحديد اذا كان هذا العدد الصحيح عدد فردي ام زوجي(ملاحظه: استخدم % لتساعدك في الحل واي عدد زوجي يقسم على ٢ يكون الباقي دائما صفرا)
بهذه الملاحظه اصبحت الفكرة جاهزة لكم

```
#include <iostream.h >
int main ( )
{
int number ;

cout << " please enter the number : " ;
cin >>number ;

if (number % 2 == 0 )
cout <<" \n The number is even " ;

else (number % 2 == 1 )
cout << " \n The number is odd " ;

return 0 ;
}
```

Questions2 :write a program that inputs one five digit number, separates the number in to individuals digits and prints the digits separated from one another by three spaces each .(hint :use combinations of integer division and the remainder operation)

Example: enter a five digit number: 12345
1 2 3 4 5

السؤال الثاني:قم بكتابة برنامج باستخدام لغة ال C++ يقوم بأدخال عدد مكون من خمسة أرقام ويفصلهما ويطبع كل عدد مفصول عن الذي يليه بثلاث مسافات (ملاحظه استخدم /, %) والمثال التالي للتوضيح.....

Example: enter a five digit number: 12345
1 2 3 4 5

```
#include <iostream.h >
int main ( )
{

int number,num1,num2,num3,num4,num5 ;

cout << " please enter a number with five digit: " ;
cin >>number ;

number =number%10000;
num1= number/10000;

number =number%1000;
num2= number/1000;

number =number%100;
num3= number/100;

number =number%10;
num4= number/10;

num5= number;

cout<<num1<</t<<num2<</t<<num3<</t<<num4<</t<<num5<<end1;

return 0 ;

}
```

الدرس الخامس:

نتناول اليوم في درسنا هذا ال switch وهي مشابهة لل else if التي سبق ودرسناها وفيه اختلاف بسيط لكن له تأثير على كتابة البرنامج سنتطرق له ان شاء الله والشكل التالي هو الشكل العام للأستخدام ال switch

switch (هنا يكتب المتغير المراد مقارنته) {
case هنا تكتب المقارنة الأولى **do the following code ;**

case هنا تكتب المقارنة الثانية **do the following code ;**

.

.

default: do the following code;

}

لاحظ ان الشكل العام لل switch موضح على مكان وضع المتغير المراد مقارنته والحالات التي يقارن به هذا المتغير وستوضح الفكرة اكثر عندما نأخذ مثال عملي على ال switch ننتقل لمثال عملي

المثال ١: برنامج يقوم بأدخال أرقام صحيحة ويقوم بطباعتها بالحروف. مثلا عندما يدخل المستخدم ١ يقوم البرنامج بطباعة one. وهكذا طبعا سنجعله الى رقم خمسة فقط لتوضيح فكرة عمل ال switch .

```
1- #include <iostream>
2- int main()
3- {
4- int number;
5- cout << "Enter Number:\t" ;
6- cin >> number;

7- switch (number) {
8- case '1':
9-     cout<<"one";
10- break;
11-case '2':
12-     cout<<"two";
13- break;
14-case '3':
15-     cout<<"three";
16-break;
17-case '4':
18-     cout<<"four";
19- break;
20 case '5':
21-     cout<<"five";
22-break;
23-default:
24-     cout<<"error";

25- } //end the switch
26-     return 0;
27-     } // end the main
```

سنبدأ شرح البرنامج من السطر السابع لأن السطور السابقة تم شرحها سابقا، لاحظ بعد ان ادخل المستخدم الرقم وخرن في متغير اسمه number هنا جعلنا ال number داخل ال switch لأنه هو الذي سنقارنه، سيمر المترجم بأول حالة وهي '1' case اذا تطابقت قيمة المتغير number مع ١ فإنه يطبق ما يندرج تحت هذا ال case فيطبع على الشاشة one وبعدها ينتقل ال break وهذه الجملة تجعل المترجم يخرج خارج ال switch وهذا ما يجعل ال switch اكثر فائدة. فلو أنك لم تضع جملة break بعد كل case لم المترجم على كل case حتى ولو وجد ظالته في ال case السابق وهذه طبعا اذاعة للوقت ناهيك انه سيطبق ال case الذي يتطابق معه

السلام عليكم ورحمة الله وبركاته.....
الحمد لله والصلاة والسلام على افضل الأنبياء والمرسلين قوتونا وحبیبنا وشفیعنا ان شاء الله محمد بن عبد الله
وعلى اله وصحبه اجمعين.....

كيف حالكم أخواني واخواتي اسأل الله ان تكونوا في أحسن حال.....اليوم بعد ان انتهينا من جملة ال switch
نتنقل بأذن الله الى موضوع مهم وعملي جدا...الا وهو جمل التكرار وتتمثل في ال,for loop, while, do while
وتستخدم هذه الجمل بدرجة واسعه في البرمجة بل معظم البرامج تحتوي عليها.

تخيل انه طلب منك عمل برنامج لأدخال بيانات موظفين هذا البرنامج يقوم بأدخال الاسم ورقم التسلسل
للموظف ووظيفته وعدد ساعات عمله طبعاً هنا ستحتاج تطلب من المستخدم اسمه ورقم التسلسل التابع له
ووظيفته و عدد ساعاته وهنا سيكون برنامجك يحوي اربع cout و اربع cin لكل موظف تخيل لو انه طلب
منك كتابة هذه البرنامج لمئة موظف او اكثر طبعاً المسألة غير مقبولة اذا استخدمت هذه الطريقة لأن البرنامج
سيكون طويل جداً وممل جداً وبدون معنى, فهنا يأتي دور جمل التكرار التي سنتقنك من هذا الموقف الحرج ومن
كتابة كل هذا الكلام.وسنشرحهم ان شاء الله بالتفصيل.

نصيحة ذهبية : المبرمج الناجح حينما يكتب برنامج يجعله مرناً بحيث يستخدم لأي استخدام للغرض الذي انشأ
من اجله مثلاً لو طلب منك عمل برنامج يحسب المتوسط طبعاً يجب ان تجعله برنامج يحسب المتوسط لأي عدد
من الدرجات ايا كان عددها وهكذا لك يستخدم في اي وقت وحاول تختار اسماء متغيرات واضحة وذات معنى
اي اذا اردت ان تختار متغير تخزن فيه الدرجات اجعل اسمه grade اي درجه لك تتذكر هدفه ومعناه اذا
رجعت له بعد زمن او اردت تطويره.....
أعتذر على الأطلاله نتنقل مباشرة لأول جمل التكرار وهي ال while.

جملة ال while:

نبدأ الآن بالشكل العام لل while

{ (هنا يكتب الشرط) while }

هنا تكتب الكود الذي تريد ان يطبقه البرنامج عند توفر الشرط فيه

}

لاحظ عند استخدام ال while يجب ان تستخدم الشكل السابق, بين القوسين تكتب الشرط لدخول
ال while ويجب ان تبدأ ال while بقوس مفتوح وتنتهيها بقوس مغلق كما هو موضح بالصورة
,وبين قوسين البداية والنهاية تكتب الكود الذي سيطبقه البرنامج عند توفر الشرط لدخول
الwhile.

طبعاً لك نفهم اكثر نتنقل لمثال عملي ستفهمون منه ان شاء الله كيفية التعامل مع ال while
أكثر.

-المطلوب كتابة برنامج يقوم بجمع الاعداد من 1 الى 10 يعني 1+2+3+4+5+.....+10.

```
1-#include <iostream>
```

```
2-using std::cout;
```

```
3-using std::endl;
```

```
4-int main()
```

```
5-{
```

```
6-    int sum=0;
```

```
7-    int x=1;
```

```

8-   while ( x <= 10 ) {
9-       sum =sum+x;
10-      x=x+1;

11-  } // end while

12-  cout << "The sum is: " << sum << endl;

13-  return 0;

14-      } // end function main

```

ننتقل الآن لشرح البرنامج:

-لاحظ في السطر السادس قمنا بحجز مساحة لنحفظ فيها المجموع واسميناها `sum` ونوعها `integer` لأن الناتج مع لأعداد صحيحة اكيد لن يكون الا عدد صحيح ,يمكن تتسائل لماذا جعلنا قيمة ال `sum =0` هنا صفرنا الخانه التي سنخزن فيها المجموع لأننا نحتاج لحفظ المجموع خانه خالية من اي رقم لك نحصل على ناتج صحيح وفي العاده الذاكره يكون موجود فيها 1 أو 0 لذلك لا نريد ان ندع المسأله للحظ اذا كان صفر حصلنا على ناتج صحيح واذا كان 1 كان الناتج غير صحيح بنصفر الخانه ونكون في الجانب الأيمن.

خذوا هذه العادة الحسنه: عندما تريدون ان تحجزوا مساحة لحفظ الناتج بداخلها في البداية عند تعريفها قوموا بتصغيرها كما فعلنا الآن.

- في السطر السابع قمنا بحجز مساحة لمتغير اسميناها `x` ونوعه عدد صحيح وهذا المتغير سيكون لنا بمثابة المؤشر الذي يؤشر على العدد الذي نصل له بعد جمع اي عدد وطبعا جعلنا قيمته المبدئية بواحد لأننا نريد جمع الأرقام من واحد الى 10 . ولو اردنا جمع الأعداد من 3 الى 10 لجعلنا قيمته المبدئية 3 .

-السطر السابع: هنا نبتدي ال `while` لاحظ يبدأ المترجم في البداية بالتأكد من الشرط , والشرط هنا (`x <= 10`) سيقارن المترجم قيمة ال `x` وهي حاليا واحد وسينتج ان الشرط تحقق لأن الواحد أقل من ال عشرة . فسيدخل التكرار ويطبق السطور الكودية التي بداخل القوسين لل `while` .

-السطر الثامن : هنا سيقوم البرنامج بجمع العدد الذي بالخانه المسماه `x` مع قيمة ال `sum` وسينتج `1=1+0` وستتغير قيمة ال `sum` الى واحد .

بعدها ينتقل البرنامج للسطر التاسع وقوم بزيادة قيمة المتغير `x` بمقدار واحد حتى ننتقل للرقم 2 ونقوم بجمعه مرة أخرى .لاحظ لن يخرج البرنامج من ال `while` اولا سيقارن الشرط وفعلا ال 2 اقل من العشرة فسيدخل التكرار مرة اخرى ويطبق السطور التي بداخل ال `while` , فيجمع الناتج وهو الآن يحتوي القيمة واحد بالقيمة التي في المتغير `x` ويضيف القيمة الجديدة وهي 3 في المتغير `sum` وينزل للسطر الذي يليه وهو السطر التاسع ويزيد قيمة ال `x` بمقدار واحد فتصبح قيمة ال `x =3` ثم يرجع البرنامج لل `while` ويقارن الشرط مرة اخرى وفعلا يتحقق لأن ال 3 أقل من 10 ويدخل مرة اخرى داخل ال `while` ويطبق السطور الكودية التي بداخل ال `while` ويستمر هكذا الى ان تصل قيمة ال `x` الى عشرة يقارنها بالشرط فيتحقق الشرط فيطبق الأسطر الكودية التي بداخل ال `while` هنا ستصل قيمة ال `sum =45` يضيف لها قيمة ال `x` الحالية وهي 10 فيصبح لدينا `10+45=55` فتتغير قيمة ال `sum=55` بعدها تزيد قيمة ال `x`

مقدار واحد فتصبح $x=11$ يرجع البرنامج ويقارن الشرط هنا لا يتطبق الشرط فيخرج البرنامج من ال while ويذهب الى السطر ١٢ الناتج على الشاشة ٥٥: The sum is. وهكذا ينتهي البرنامج وبهذا النمط والتكرار تكون طريقة سير البرنامج بأستخدام ال while اتمنى ان تكون قد وصلت الفكرة وسأسرد أمثلة اخرى لتوضيح عمل ال while ولكن لن نفضل سير البرنامج بهذا التفصيل السابق لأننا قد سردنا التفصيل في المثال الأول ...

مثال ٢: برنامج يقوم بطباعة مضروب العدد في نفسه من واحد الى عشرة وطباعة مجموعهم.
مثال الناتج على الشاشة سيكون كالتالي :

```
١
٤
٩
١٦
٢٥
٣٦
٤٩
٦٤
٨١
١٠٠
The total is :385
```

الحل:

```
#include <iostream>

using std::cout;
using std::endl;

int main()
{
    int y;           // declare y
    int x = 1;       // initialize x
    int total = 0;   // initialize total

    while ( x <= 10 ) { // loop 10 times
        y = x * x;      // perform calculation
        cout << y << endl; // output result
        total += y;     // add y to total
        ++x;           // increment counter x
    } // end while

    cout << "Total is " << total << endl; // display result

    return 0; // indicate successful termination
} // end function main
```

ملاحظه : التعليقات التي بعد علامة // مجرد تعليق لك انت اي ان البرنامج لا يعطيه اي اهمية وقد وضحت هذه النقطة في الدروس السابقة لكن اعادة هذه الملاحظه للتنبيه لا تضر شيئاً.

كما تعودنا اليكم هذا السؤال الذي اتمنى ان تحاولوا ان تجدوا الحل له وهو سهل جدا وسأضيف اجابته في الدرس القادم ان شاء الله.

- قم بكتابة برنامج باستخدام لغة السي ++ يقوم بطلب الاساس والاس من المستخدم وايجاد الناتج .
مثال .

```
Enter base as an integer:
٣
Enter exponent as an integer
٢
Power =9
```

السؤال سهل اليس كذلك 😊
هنا نصل الى نهاية الدرس اليوم وفي الدرس القادم سنأخذ امثلة اكثر على ال while وتمارين أعمق . فيجب ان تتمكن من استخدامها جيدا للأهميتها البالغة في برمجة السي ++.

°~*α@§(*§(أنت استغفرك وأتوب إليك *§)§@α*~°

~*α!||!α*!~((كاتبه الدرس بنت النور..... عفوا يمنع اجراء اي تعديل للملف))~*α!||!α*!~

~*α!||!α*!~((نسألکم دعوة صادقة لأختکم في ظهر الغيب))~*α!||!α*!~

بسم الله الرحمن الرحيم

السلام عليكم ورحمة الله وبركاته.....
الحمد لله والصلاة والسلام على افضل الأنبياء والمرسلين قدوتنا وحبیبنا وشفیعنا ان شاء الله محمد بن عبد الله
وعلى اله وصحبه اجمعين.....

أجمل تحية لكم اخوتي واخواتي في الله
اتهينا في الدرس السابق من شرح ال while وذكرنا مثالين عليها وقمنا بطرح سؤال والآن ان شاء الله نواصل
الجزء الثاني من درس ال while حيث سنتدرب على برامج أكثر باستخدام ال while.

نبدأ بحل السؤال الذي ذكرناه في الدرس السابق

السؤال ١:

- قم بكتابة برنامج باستخدام لغة السي ++ يقوم بطلب الاساس والاس من المستخدم وايجاد الناتج
مثال .

```
Enter base as an integer:
٣
Enter exponent as an integer
٢
Power =9
```

الجواب :

```
#include <iostream>

using std::cout;
using std::cin;
using std::endl;

int main()
{
    int x;        // base
    int y;        // exponent
    int i;        // counts from 1 to y
    int power;    // used to calculate x raised to power y

    i = 1;        // initialize i to begin counting from 1
    power = 1;    // initialize power

    cout << "Enter base as an integer: "; // prompt for base
    cin >> x; // input base

    cout << "Enter exponent as an integer: ";
    cin >> y; // input exponent

    // count from 1 to y and multiply power by x each time
    while ( i <= y ) {
        power *= x;
        ++i;
    } // end while

    cout << "power=" << power << endl; // display result
```

```
return 0;
} // end function main
```

ملاحظه : التعليقات التي بعد علامة // مجرد تعليق لك انت اي ان البرنامج لا يعطيه اي اهميه وقد وضحت هذه النقطة في الدروس السابقه يعني ليست من ضمن الكود للبرنامج ...

تستطيع استخدام تكرار داخل تكرار اي **while loop** داخل **while loop** والمثال التالي سيوضح الفكرة. وايضا ينقسم التكرار في ال **while** الى محدد (ثابت لعدد معين) وغير محدد يحدده المستخدم فالرقم للتكرار في الغير المحدد ليس ثابت يتحكم فيه المستخدم وهذا الأفضل حيث يكون اكثر مرونة. وسنوضح الحالتين سنبدأ بالمحدد ومن ثم الغير محدد.

السؤال ٢ (التكرار المحدد) :

مطلوب عمل برنامج بأستخدام لغة السي++ يقوم هذا البرنامج بحساب المتوسط لدرجات ٥٠٠ طالب ويجب على البرنامج ان يطلب من المستخدم ادخال الدرجات ويتأكد البرنامج من ان الدرجة التي تم ادخالها بين الصفر وال ١٠٠ ولا تتجاوز هذا الحد.

```
#include <iostream>

using std::cout;
using std::cin;
using std::endl;

int main()
{
Float grade,total=0,avg;
int count=1;

While(count<=500)
{
cout<<"please enter the grade";
cin>>grade;

while(grade<0 || grade>100)
{

Cout<<"sorry you enter invalid grade, enter grade 0-100;
Cin>>grade;

} //second while

total+=grade;

count++;

} //first while
```

```

avg=total/500;

Cout<<"the avg is "<<avg;

} // end function main

```

الشرح:

نلاحظ في هذا السؤال العدد محدد ل 500 طالب اي ان التكرار سيكون 500 مرة سنقوم بشرح الاجزاء التي تتعلق بدرسنا اليوم والبقية شرحناهم في دروس سابقه.

لاحظ في البداية حجزنا خانة لمتغير اسميناها ال count لك يسير لنا العد ففي البدايه كانت قيمته واحد فمر على ال while تأكد من تحقق الشرط والشرط هو ان قيمة ال count لا تزيد عن 500 لأننا نريد التكرار 500 مره فقط بعدها يطلب البرنامج من المستخدم ان يدخل له الدرجه ومن ثم ينتقل الى ال while الداخليه التي يقوم بالدخول لها اذا تحقق شرطها وهو ان تكون الدرجه التي ادخلها المستخدم اكبر من منه او اصغر من صفر ،طبعا اذا تحقق هذا الشرط هذا يعني ان المستخدم اخل درجه خاطئه فيطلب منه البرنامج اعاده ادخال الدرجه كما ترون داخل ال while الداخليه ولا ينتقل خارج ال while الداخليه الا اذا ادخل درجه صحيحه هنا نضمن ان جميع الدرجات صحيحه ،فاذا ادخل المستخدم درجه في حدود 100-0 هنا ينتقل المترجم الى جمع الدرجه للحصول على المجموع وقد شرحنا هذا بالتفصيل في درس سابق ومن ثم يزيد قيمة ال count بمقدار واحد.

ربما يتبادر الى ذهنكم ما هذا total+=grade; طبعا هذا يساوي total=total+grade ولكن اختصارا كتبناها بالطريقة السابقه وايضا ++count; تعني count=count+1; ولكن اختصرناها بالطريق السابقه . دعونا نتدرب على هذه الاختصارات:

```

count--; تعني count=count-1;
sum*=grade; تعني sum=sum*grade;
أرجو ان تكون قد وضحت الفكره.....نعود الى برنامجنا.

```

بعد ان يجمع البرنامج الدرجه في ال total ويزيد قيمة ال count بمقدار واحد يعيد هذه الخطوات الى ان يصبح مقدار ال count=500 عندها يخرج من ال while ويذهب ليقوم بقسم المجموع للدرجات على 500 للحصول على المتوسط ومن ثم يقوم بطباعته ...

لاحظ جعلنا العمليه الحسابية لأيجاد المتوسط خارج ال while لأننا نريد المجموع للدرجات واذا وضعنا العمليه لأيجاد المتوسط ;avg=total/500; ووضعنا ايضا cout<<"the average is "<<avg;

هنا سيقوم البرنامج بأيجاد المتوسط وطباعته في كل مرة يدخل فيها لل while يعني خمسمائة مرة وطبعا هذا لا يعطينا المطلوب لذلك وضعناه بالخارج لك عندما ينتهي البرنامج من الحصول على المجموع يقوم بأيجاد المتوسط وطباعتهوهنا سيطلع مره واحده.

اتمنى ان تكونوا قد فهمتم كيف يكون التكرار محدد وهو يتمثل في تحديد قيمة للتكرار وفي مثالنا هذا كانت 500 وجعلنا ال count كمؤشر يعد لنا 500 مرة.

مطلوب عمل برنامج باستخدام لغة السي++ يقوم هذا البرنامج بحساب المتوسط لدرجات عدد غير محدد من الطلاب يتحكم في هذا العدد المستخدم ويجب على البرنامج ان يطلب من المستخدم ادخال الدرجات ويتأكد البرنامج من ان الدرجه التي تم ادخالها بين الصفر وال 100 ولا تتجاوز هذا الحد.

```

#include <iostream>

using std::cout;
using std::cin;
using std::endl;

int main()
{
    Float grade,total=0,avg;
    int count=0;

    cout<<"please enter the student grade(enter -1 to end)";
    cin>>grade;

    while(grade!=-1)
    {

        while(grade<0 || grade>100)
        {

            Cout<<"sorry you enter invalid grade, enter grade 0-100;
            Cin>>grade;

            } //second while

        total+=grade;

        count++;

    cout<<"please enter the student grade(enter -1 to end)";
    cin>>grade;

    }//first while

    Arg=total/count;

    Cout<<"the avg is "<<avg;

} // end function main

```

الشرح:

نلاحظ في هذا السؤال التكرار غير محدد اي يدخل المستخدم درجات الى ان يكتفي ويطلب انهاء البرنامج...
 تابع معي البرنامج في البداية سيطلب البرنامج من المستخدم ادخال اول درجة أو ١- لأنهاء البرنامج هنا جعلنا الإشارة التي نعرف اذا اكتفى المستخدم من ادخال الدرجات ام لا هي ١- أي، عندما ينتهي من ادال الدرجات وتظهر له رساله ادخال الدرجه يدخل ١- وسينتهي البرنامج ويعطيه المتوسط.

وأخترنا -١ لأن لا يمكن ان توجد درجه بال -١ وتستطيع اختيار اي شيء ماعدا ان يكون من ضمن الدرجات ٠-١٠٠.

بعد ان يدخل المستخدم درجة الطالب يقوم البرنامج في ال while بالتأكد من ان الدرجة ليست -١ واذا كانت -١ يقف البرنامج ويعطي المتوسط. إذا لم تكن -١ يدخل البرنامج داخل ال while الأولى ومن ثم يدخل داخل ال while الثانية التي تتأكد من ان الدرجة من ٠-١٠٠ وان ادخل المستخدم درجه خارج هذا النطاق يبقى البرنامج داخل ال while ويطلب من المستخدم ادخال درجه صحيحه ومن ثم بعد ادخال درجة صحيحه ينتقل البرنامج الى جمع الدرجه مع المجموع وزيادة ال count بمقدار واحد. قد تسأل نفسك هنا لا نحتاج count لتسيير البرنامج وتحديد قيمة التكرار لن العدد غير محدد والمستخدم هو الذي يحدد العدد. لكننا وضعنا ال count هنا لأننا نريد معرفة كم عدد الدرجات التي ادخلت لأننا نحتاج قيمة ال count لحساب المتوسط لأن المتوسط عباره عن مجموع الدرجات على عددها. مجموع الدرجات نحصل عليه من ال total و عددها من ال count بعدها وضعنا

```
cout<<"please enter the student grade(enter -1 to end)";
cin>>grade;
```

تخيل لو اننا لم نضع هذه الجملة اخر ال while الأولى لكان البرنامج بعد ان ينتهي من زيادة ال count بواحد لرجع لل while الأولى وطبعا قيمته ال grade ستبقى قيمة ال grade الأولى ولن تتغير وسيكون البرنامج في تكرار غير منتهي مما يسبب مشكلة كبيرة ولن يعطينا المتوسط ابدا.. لأننا لم نعطي المستخدم مجال لأدخال درجة أخرى.

لكن مع وضع هذين السطرين يسطلب البرنامج من المستخدم ادخال درجة جديده ويرجع لل while الأولى ويتأكد ما اذا كان الشرط صحيح وان الدرجه لا تساوي -١ واذا كانت -١ يتوقف ويعطي المتوسط. اما اذا كانت لا تساوي -١ دخل البرنامج الى ال while الثانية وتأكد من ان الدرجه في النطاق ٠-١٠٠ وبعدها جمع الدرجه الجديده مع المجموع السابق وزاد قيمة ال count بواحد ويطلب البرنامج مرة اخرى من المستخدم ادخال درجة جديده فأذا كانت -١ توقف واعطى المتوسط واذا كانت غير ذلك واصل ما قلناه سابقا وهكذا يستمر البرنامج حتى يدخل المستخدم -١. فقط عندها يتوقف ويعطي المتوسط والا يبقى مستمر في أخذ الدرجات وجمعها.

بعد ادخال -١ يخرج البرنامج من ال while ويقسم المجموع للدرجات على عددها الذي يتمثل في ال count ويعطي المتوسط ويطبعه على الشاشة.

الفرق بين المثال الأول والثاني اي بين التكرار المحدد والغير محدد ان في الأول قيمة التكرار تحدد في ال while الأولى أما في الثاني الشرط الذي يوقف البرنامج يوضع في ال while الأولى وطبعا في التكرار الغير محدد لا تنسى من وضح cout لك تخبر المستخدم بأدخال قيمة جديده و cin لك تحفظ هذه القيمة في المتغير المخصص لها كما فعلنا في هذين السطرين

```
cout<<"please enter the student grade(enter -1 to end)";
cin>>grade;
```

أرجو ان يكون الفرق واضح لديكم

الآن ننتقل لمثال قمنا بحله سابقا لكن بدون استخدام التكرار وكان لأدخال رقم واحد فقط طبعا كان برنامج غير مرن لأنه بعد ان يعطيك نتيجة عدد واحد ادخلته ينتهي عمله انظروا للبرنامج بعد تطويره بأستخدام التكرار وقارنوا بينهم وانظروا الفرق.....

Question1: write a program that reads integers and determine whether it is odd or even. (Hint: use the modulus operator .an even number is a multiple of two. any multiple of two leaves a remainder of zero when divide by 2).

Make sure to make this program for non fix number.

السؤال ٣: قم بكتابة برنامج بأستخدام لغة ال C++ يقوم بأدخال اعداد صحيحة وفي كل مره يدخل المستخدم عدد يقوم البرنامج بتحديد اذا كان العدد فردي او زوجي (ملاحظه: استخدم % لتساعدك في الحل واي عدد زوجي يقسم على ٢ يكون الباقي دائما صفرا)
بهذه الملاحظه اصبحت الفكرة جاهزة لكم

```
#include <iostream.h >
int main ( )
{
char flag=y
int number ;

while(flag!=n)
{
cout << " please enter the number : " ;
cin >>number ;

if (number % 2 == 0 )
cout <<" \n The number is even " ;

else (number % 2 == 1 )
cout << " \n The number is odd " ;

cout<<"do u want to check another number,y=yes,n=no"<<end1;
cin>>flag;

while(flag!=y && flag !=n)
{
cout<<"sorry rong command";
cout<<"do u want to check another number,y=yes,n=no"<<end1;
cin>>flag;

}

}
```

```
return 0 ;  
}
```

نصل هنا لنهاية الدرس السابع وانهاء دراسة ال while ارجو ان تكونوا قد فهمتوها جيدا فهي جملة مفيدة جدا
وسنتقل في الدرس القادم بأذن الله الى ال for loop ..

الى ذلك الحين دتم بخير وعافيه والسلام عليكم ورحمة الله وبركاته.

°~*α@§(*§ استغفرك وأتوب إليك *§)@α*~°

~*α!||!α*~ ((كاتبة الدرسبنت النور.....عفوا يمنع اجراء اي تعديل للملف))~*α!||!α*~

~*α!||!α*~ ((نسألکم دعوة صادقة لأختکم في ظهر الغيب))~*α!||!α*~


```

#include <iostream>

using std::cout;
using std::endl;

int main()
{
    for ( int counter = 1; counter <= 10; counter++ )
        cout << counter << endl;

    return 0;
}

```

شرح البرنامج :
 كتابة البرنامج لا تتجاوز سطور قليلة مقارنة بكتابه باستخدام ال while والبرنامج بكامله يتمثل في السطر التالي

```

for ( int counter = 1; counter <= 10; counter++ )

```

نلاحظ كما شرحنا في البداية الشكل العام لجمله ال for سنستخدم للعد المتغير counter وسنجعل قيمته الابتدائية واحد `int counter = 1`, وبعدها جعلنا القاعده التي يتوقف عندها هذا العد عندما يتجاوز العد الرقم عشرة `counter <= 10` وطبعاً نزيد قيمة ال counter بمقدار واحد في كل مره. تابع معي سيداً العد وقيمة ال `counter = 1` عندها يتأكد من القاعده وسيجد ان ال `counter < 10` وسيزيد قيمة ال counter بمقدار واحد وينزل للسطر `cout << counter << endl;` ثم يدخل التكرار مره اخرى ويتأكد ان الرقم ٢ أقل من العشره ويزيد ال counter بمقدار واحد ثم يطبع ٢ ويواصل الى ان يصل العد الى الرقم عشرة يتأكد من ان ال counter الذي وصلت قيمته عشرة أقل من او يساوي ١٠ , وطبعاً هنا يساوي العشره يزيد ال counter بمقدار واحد ويطبع العشره ويخرج البرنامج من ال for لأن ال counter اصبحت قيمته تساوي ١١. وكما تلاحظون بكل بساطه ينتهي البرنامج هنا , اضنكم لاحظتم سهوله واهميه استخدام ال for.

ننتقل الى مثال عملي آخر :

المطلوب كتابة برنامج يقوم بطباعة المتوسط ل ١٠٠ طالب وايجاد اقل درجه بين الطلاب واعلى درجه وطباعتهم:

```

#include <iostream>

using std::cout;
using std::endl;

int main()
{float total=0,max=0,min=100,avg;
    for ( int counter = 1; counter <= 100; counter++ )
        { cout << "enter the grade" << endl;
          cin>>grade;
          total=total+grade;
          if(grade>max)
            { max=grade; }

          If(grade<min)
            {min=grade; }
        }
}

```

```

}
Avg=total/100;
Cout>>"the average is ">>avg>>end`;
Cout>>"the maximum grade is">>max>>endl;
Cout>>"the minimum grade is">>min>>endl;

return 0;
}

```

نأخذ مثال آخر برنامج يقوم بجمع الاعداد الزوجية من ٠ الى ال ١٠٠ :-

```

#include <iostream>

using std::cout;
using std::endl;

int main()
{
    int sum = 0;

    for ( int number = 2; number <= 100; number += 2 )
        sum += number;

    cout << "Sum is " << sum << endl;
    return 0;
}

```

هنا نصل الى نهاية درس ال for الذي اتمنى ان يكون درس مفيد لكماترككم برعاية الله وحفظه حتى
الدرس القادم...

الى ذلك الحين دمتم بخير وعافيه والسلام عليكم ورحمة الله وبركاته.

°~*α@§(*§استغفرك وأتوب إليك *§)§@α*~°

~*α!||!α*~((كاتبة الدرسبنت النور.....عفوا يمنع اجراء اي تعديل للملف))~*α!||!α*~

~*α!||!α*~((نسألکم دعوة صادقة لأختکم في ظهر الغيب))~*α!||!α*~