الدرس الثالث : العمليات الرياضية .

أولاً : المتغيرات .

دعني أولا أن أذكرك ببرنامج الجمع ، الذي توقفنا معه :

```
using System;
class Addition   // Addition Program
{
        static void Main ()
        {
                string  firstNumber,//متغير يحفظ العدد الأول من عن المستخدم
                        secondNumber;//متغير يحفظ العدد الثاني من عن المستخدم
                int      x,              //متغير يحفظ العدد الأول كعدد صحيح
                         y,              //متغير يحفظ العدد الثاني كعدد صحيح
                         sum;            //متغير يحفظ الجمع
                // نطلب من المستخدم أن يدخل الرقم الأول
                Console.Write ("Please Enter the First Number: ");
                // نقرأ السلسلة  الحرفية التي سوف يدخلها المستخدم
                firstNumber = Console.ReadLine();
                //نطلب من المستخدم أن يدخل  الرقم الثاني
                Console.Write ("Please Enter the Second Number: ");
                // نقرأ السلسلة  الحرفية التي سوف يدخلها المستخدم
                secondNumber = Console.ReadLine ();
                //نقوم بتحويل سلاسل الحرفية إلى أعدد صحيحة
                x =Int32.Parse (firstNumber);
                y = Int32.Parse (secondNumber);
                //نقوم بعملية الجمع هنا
                sum = x + y;
                //نعرض ناتج عملية الجمع
                Console.WriteLine ("The Sum = {0}", sum);
        }// انتهت الوسيلة Main
} // انتهت الخلية
```

و الناتج كالتالي :

```
Please Enter the First Number: 15
Please Enter the Second Number: 6
The Sum = 21
```

ما الذي فعلناه في هذا البرنامج ؟ أعلنا عن أربعة متغيرات اثنان من نوع سلسلة حرفية string ، و اثنان من نوع عدد صحيح int ، ثم طلبنا من المستخدم إدخال عددين ، ثم حولناهما إلى أراقم صحيحة ، ثم أجرينا عملية الجمع ، و أظهرنا ، هذا بشكل مختصر ما فعلناه!!

و الآن أود أن أشرح لك بعض الأمور بعمق أكثر ، و انتبه معي ...

في السي شارب توجد الكثير من أنواع المتغيرات ، وهي تنتمي إلى صنفين لا أكثر : صنف ذو قيمة ، و صنف ذو مرجع ، الصنف ذو قيمة يحتوي على قيم سوءا كانت أعداد أو حروف وغيرها من الأنواع ، أما ذات المرجع فهو يحمل مرجع أو مؤشر إلى كائن من نفس النوع ، أمثالة على ذلك الخلايا و المصفوفات وغيرها ، لا تقلق سأشرح لك هذا الصنف بالتفصيل أكثر عندما نأتي إليه .

دعنا الآن في الصنف ذو القيمة ، قالنا أنه يحوي على أعداد ، و لكن الأعداد أو الأرقام مقسمة إلى عدة أنواع فهناك أعداد صحيحة int مثل 1 ، 2 ، -3 ، -4 ، و هناك أعداد ذات فاصلة عائمة float مثل 1.01، 5.3 ، 50.4- ، هذا من ناحية ، ومن ناحية أخرى أن لكل صنف من هذه الأصناف له حدود معينه ، فإذا أردنا أن نتعامل مع أعداد كبيرة نوعا ما فإنا نستخدم نوع يسمى double ، إلى هنا يكفي للأعداد سأشرحها بالتفصيل في دروس لاحقه ، فلنتقل إلى الحرف ، يوجد نوع char مخصص للحروف والرموز ، بحيث يحوي على حرف أو رمز واحد أو سلسة هروب واحدة ، على سبيل المثال 's' و 'Z' و '$' و '\n' ، لاحظ أني وضعتها بين علامتي اقتباس مفردة ' ' حتى أميزها عن السلاسل الحرفية string ، و التي يمكن أن تضع فيها أكثر من حرف أو رمز ، و تكون محصورة بين علامتي اقتباس " " .

ولكي تنشئ متغير يجب عليك أولا أن تحدد نوعه ثم تكتب اسمه كالتالي :

```
int  Intger;
```

و تستطيع أن تعلن عن أكثر من متغير في نفس الجملة و مثال على ذلك ا لتالي :

```
string Sant1 , Sant2 , Sant3;
```

بحيث تفضل بين أسماء المتغيرات بفاصلة .

فإذا رجعنا إلى برنامج الجمع ، ستقول لماذا احتجنا إلى متغيرات من نوع السلاسل الحرفية و نحن بصدد جمع أعداد و ليس تكوين جمل ؟؟

وجواب ذلك ، أن السي شارب تتعامل مع أي مدخل على أنه سلسلة حرفية ، فلذا يجب أن يوضع في متغير من نوع سلسلة حرفية ، ثم يجرى عليه التحويل إلى العدد صحيح .

وهنا يظهر سؤال ألا و هو كيف يمكن إسناد قيمة إلى متغير معين ؟؟ (المتغير من الصنف قيمي) ، سنجيب أن هناك معامل يسمى معامل الإسناد وهذا رمزه = خذ المثال التالي :

```
x = 5;
```

في هذه الجملة أسندنا 5 إلى المتغير x ، بعد تنفيذ هذه الجملة صوف يصبح x يحمل قيمة 5 فقط ، مما يعني أنه إذا كان يحوي من قبل على قيمة أخرى ، فإنها سوف يكتب فوقها – إن صح التعبير –.

والآن ركز في السطر التالي :

```
firstNumber = Console.ReadLine();
```

أظن أنك قد فهمت مغزاها ، وسوف تقول مفتخراً : أن وظيفة هذه الجملة هو إسناد قيمة الراجع من الوسيلة **ReadLine** إلى المتغير **firstNumber** .

والآن بعد أن أسندنا قيم إلى المتغيرين `firstNumber` و `secondNumber` ، نريد تحويلها إلى من سلاسل حرفية إلى أعداد صحيحة ؛ حتى نجري عليها عملية الجمع ، لا تقلق فالسي شارب وفرت لنا وسيلة تقوم بذلك عنا ألا و هي `Int32.Parse` والسطر التالي يوضح كيفية استعمالها :

```
x =Int32.Parse (firstNumber);
```

حيث سيحمل المتغير **x** قيمة الراجع من الوسيلة `Int32.Parse` ، وسوف نكرر هذا مع المتغير الثاني .

وبعد أن حولنا المتغيرين إلى نوع صحيح يمكننا الآن أن نقوم بجمعهما ، و الجملة التالية تتكفل بذالك :

```
sum = x + y;
```

وقبل أن انتقل إلى السطر الأخير من البرنامج ، عندي إحساس بأن بعض من قراء هذا الدرس سوف يقول أنه لا داعي لإعلان متغيرين من نوع سلاسل حرفية ، بحيث يمكننا كتابة السطر الذي سوف يستعملهما من :

```
x =Int32.Parse (firstNumber);
```

إلى

```
x =Int32.Parse (Console.ReadLine() );
```

و نضعه بعد أن نطلب من المستخدم إدخال العدد الأول ، و نفعل نفس الشيء مع المتغير الثاني ، سأقول ما شاء الله عليك ، بالفعل وفرت علينا و قصرت من طول الشفرة ، أين أنت من زمان ؟؟ ، سأعدك في المرات القادمة سوف أتبع نصيحتك .

لننتقل الآن إلى جملة إظهار المخرجات و هي كالتالي :

```
Console.WriteLine ("The Sum = {0}" , sum);
```

سأختصر الشرح و أقول : عندما تنفذ هذه الجملة سوف يقوم المترجم باستبدال {0} بأول متغير يجده بعد انتهاء الجملة المراد إظهارها ، وإذا كانت أكثر من واحدة سوف يستبدلها بالترتيب ، افرض الآن أنك تريد أن تظهر العبارة التالية `x + y = sum` بعد استبدال المتغيرات بقيمها ماذا ستكتب ؟؟

فكر يا حبيبي فكر ....

أتوقع أن تكتب الجملة التالية :

```
Console.WriteLine ("{0} + {1} = {2} ", x, y, sum);
```

هنا طريقة أسرع من هذه الطريق ، سوف نذكرها لاحقا في هذا الدرس .

ثانيا : العمليات الرياضية .

إذا كنت من الذي لا يحبون الرياضيات فإن السي شارب عندها خبر غير مفرح لك ؛ حيث أن معظم البرامج التي سوف تكتبها سوف يكون فيها علميات الرياضية ، لا تقلق كثيرا فغالية هذه العمليات هي عمليات الجمع و الطرح و الضرب و القسمة لا غير .

في البرنامج السابق قمنا بعملية الجمع ، و يمكنك أن تقوم بعملية الطرح والضرب و القسمة على نفس المنوال ، أضف إلى هذه أنه يمكنك أن تقوم بعملية إضافية مهمة ، ألا وهي عملية باقي القسمة Modulus ، و تمثل بالسي شارب بالرمز ٪ ، خذ المثال التالي :

```
10 % 2 = 0
```

لأني ناتج قسمة 10 على 2 يساوي **5** و الباقي 0 .

هناك نقطة مهمة عندما تقوم بالعمليات الحسابية على أعداد صحيحة int ، و هي أنه إذا قمت بقسمة عدد صحيح على عدد صحيح سوف يكون الناتج عدد صحيح خذ المثال التالي :

```
5 / 2  = 2
```

رياضيا الناتج هو 2.5 و لكن بما أن العملية على أعداد صحيحة ؛ فيجب أن تكون النتيجة (برمجياً) عدد صحيح و 2.5 ليس عدد صحيح ، فلهذا سوف تحذف 0.5 و يبقى 2 .

سأفترض الآن أنك تحب الرياضيات ، فسأقول لك أن لغة سي شارب تتبع نفس المنطق الرياضي في تنفيذ المعادلات الرياضية ، خذ هذه المعادلة :

$$y = \frac{(x+z)\times 5}{2}\times 3+1$$

واستعن بهذا الجدول في تحديد أي من عمليات ينفذ أولا في كتابة جملة تؤدي هذه المعادلة :

المعامل	العملية	ترتيب التنفيذ
( )	الأقواس	تنفذ أولا ، فإذا كانت هناك أقواس داخلية فإنها تنفذ أولا تليها الخارجية ، و إذا كانت هناك عدة أقواس متجاورة فإن التنفيذ يبدأ من اليسار إلى اليمين
* أو / أو ٪	الضرب ، القسمة باقي القسمة	تنفذ ثانيا ، فإذا كانت هناك عدة عمليات منها متجاورة فإن التنفيذ يبدأ من اليسار إلى اليمين .
+ أو -	الجمع و الطرح	تنفذ أخر ، فإذا كانت هناك مجموعة منها متجاورة فإن ا لتنفيذ يبدأ من اليسار إلى اليمين.

سوف التكون الجملة المكافئة لتلك المعادلة كتالي :

```
y = (x + z) * 5 /2 *3 +1
```

ترتيب العمليات **6 1 2 3 4 5**

طيب ما رأيك أن تكتب جملة أخرى لهذه المعادلة :

$$y = ax^2 + bx + c$$

أتوقع إجابتك أن تكون كتالي :

```
y = a * x * x + b * x + c
```

ترتيب العمليات **6 1 2 4 3 5**

يمكنك استعمال الأقواس ، و لو لم تكن هناك ضرورة لها ، لتسهل عليك فهم و كتابة المعادلات المعقدة .

ثالثا : اتخاذ القرار بواسطة if

ما رأيك أن نجعل برنامجنا يتخذ بعض القرارات في ضوء توافر بعض الشروط ؟؟ أليست فكرة رائعة ؟؟

حسناً ، يوجد في السي شارب كملة محجوزة لهذا الغرض ، ألا وهي if ، و الشكل التالي يوضح كيفية عملها :

```
if (يوضع الشرط هنا)
        ; نفذ هذه الجملة إذا كان الشرط صحيح
; تابع تنفيذ البرنامج إذا كان الشرط غير صحيح
```

الجدول التالي سوف يساعدك على فهم كيفية كتابة الشروط باستخدام معاملات التساوي و ا لعلاقات :

نوع الشرط	ما يمثله بالسي شارب	مثال عليه	معناه
معاملات التساوي			
==	==	x == y	x تساوي y
≠	!=	x != y	x لا تساوي y
معاملات العلاقات			
>	>	x > y	x أكبر من y
<	<	x < y	x أصغر من y
≤	<=	x <= y	x أصغر أو يساوي y
≥	>=	x >= y	x أكبر أو يساوي y

بعد أن تكتب الشرط ، تكون آلية تنفيذ if كالتالي : إذا كان الشرط صحيح (true) فنفذ الجملة التي تلي الشرط مباشرة ، أما إذا كان خاطئ (false) فقفز إلى الجملة التي تلي الجملة المباشرة للشرط .

تصور الآن أنك كتبت الجملة الشرط و وضع بعدها بالخطأ ; ماذا سوف يحدث ؟؟الذي سوف يحدث أنه عندما يكون الشرط صحيح فإن البرنامج لا ينفذ أي شيء ؛ لأنه اعتبر ; جملة كاملة ولكن بدون فائدة ...

مثال على ذلك :

```
if  (x == 0) ;
Console.WriteLine ( " Hello , Mistake ");
```

إن ناتج هذا البرنامج هو انه يظهر عبارة Hello , Mistake في كلا الحالتين ، سواء كان الشرط صحيح أو غير صحيح .

طيب ، إذا أردت أن تكتب أكثر من جملة تنفذ عند تحقق الشرط فماذا تفعل ؟

الجواب هو أنك تحصرها بين قوسين معقوفين { } والمثال التالي يوضح ذلك :

```
if ( x != 5 )
{
        Console.WriteLine( " Again and Again ");
        Console.WriteLine (" Hello , C#");
}
```

طيب أريد أسألك عن الفرق بين = و == ؟؟

الأول هو معامل إسناد ، أي تسند قيمة إلى المغير الذي في الجهة اليمين إلى المتغير الذي في جهة اليسار ، أما الثاني فهو معامل علاقة التساوي ، و نتاجه هو صح true أو خطأ false .

و ا لآن افتح الفيجوال استيديوا و اكتب البرنامج التالي :

```
// sixth Program
using System;
class Comparison
{
        static void Main ()
        {
                int number1,  //تعلن عن متغير الأول من نوع عدد صحيح
                        number2;  //تعلن عن متغير ثاني من نوع صحيح
                //تطلب من المستخدم أن يدخل ا لرقم الأول
                Console.Write ("Please Enter Number One : ");
                //تقرأ العدد و تحوله مباشرة إلى عدد صحيح و تسند قيمته إلى المتغير الأول
                number1 =Int32.Parse( Console.ReadLine());

                 // تطلب من المستخدم أن يدخل العدد الثاني
                Console.Write ("Please Enter Number Tow : " );
                // تقرأ العدد الثاني و تحوله إلى عدد صحيح و تسند قيمته إلى المتغير الثاني
                number2 =Int32.Parse( Console.ReadLine ());
                // تقوم بإجراء بعض الشروط على العددين
                if (number1 == number2)
                        Console.WriteLine (number1 + " == " + number2);
                if (number1 != number2)
                        Console.WriteLine (number1 + " != " + number2);
                if (number1 > number2)
                        Console.WriteLine (number1 + " > " + number2);
                if (number1 < number2)
                        Console.WriteLine (number1 + " < " + number2);
                if (number1 >= number2)
                        Console.WriteLine (number1 + " >= " + number2);
                if (number1 <= number2)
                        Console.WriteLine (number1 + " <= " + number2);
        } // انتهت الوسيلة
}// انتهت الخلية
```

و المخرجات كالتالي :

```
Please Enter Number One : 3
Please Enter Number Tow : 3
3 == 3
3 =< 3
3 => 3
```

```
Please Enter Number One : 2
Please Enter Number Tow : 1
2 != 1
1 < 2
1 =< 2
```

```
Please Enter Number One : 3
Please Enter Number Tow : 7
3 != 7
7 > 3
7 => 3
```

تفحص البرنامج ، وانظر كيف جعلته يتخذ قراراته بنفسه ، هنيئا لك ...

و لا تنسى أنك تعلمت أيضا طريقه جديد لربط السلاسل الحرفية بواسطة معامل الربط + .

قد يختلط عليك الأمر قليل ، فمتى نستعمل + للجمع و متى نستعمله للربط ؟ فـأقول إذا كـان مـا قبلـه سلسلة حرفية فإنه يعمل للربط و إلا فإنه للجمع .

خذ المثال التالي :

```
" y + 2 =" + y +2
```

يكون نتاجه إذا كانت قيمة y = 5 كتالي : " y + 2 = 52" و ليس 7 حيث أن + عمل معامـل ربـط هنا ، طيب عدل الشفرة السابقة إلى :

```
" y + 2 = " + ( y +2)
```

ستجد أن الناتج هو 7 .

قبل أن نختم هذا الدرس أود أن أختبر فهمك للمعامل الإسناد خذ المثال التالي :

```
x = 6 ;
y = x =0;
```

فكم قيمة y و x ؟

إذا أجبت أن y = 6 و x = 0 فقد أخطأت ، و لكل جواد كبوة ، لا عليك سوف تتعلم من أخطاءك ، و سبب في ذلك أن معامل الإسناد يعمل بعكس كل المعاملات الأخرى فهو يعمل من اليمين إلى اليسار و ليس العكس فالمثال السابق يمكن أن تكتبه كالتالي :

```
y = ( x = 0 );
```

والآن جاء دور الواجب ، فشد الهمة :

1 – اكتب برنامج لحساب ميزانيتك الخاصة بحيث تكون المدخلات الراتب و العلاوات و مصاريف الكهرباء و المياه و الهاتف ، و المخرجات إجمالي الدخل ، و إجمالي المصاريف ، و المتبقي .

2 – لمحبي الرياضيات اكتب برنامج لحساب مساحة و محيط الدائرة ، بحيث تكون المدخلات نصف قطر الدائرة و المخرجات مساحة الدائرة و محيطها .[محيط الدائرة = 2 * r * Π و مساحتها =$(r)^2$ * Π حيث r هو نصف قطر الدائرة] .

3 – اكتب برنامج يقوم باستقبال عددين ، فإن كان العدد الثاني هو عامل من عوامل العدد الأول يظهر لنا "OK" و أما إذا كان غير ذلك فيظهر لنا " Try Again " . [استخدم معامل باقي القسمة والقسمة] .

4 – اكتب برنامج يقوم بفصل خانات عدد مكون من خمس خانات ، يستقبله من المستخدم ، ثم يفصل كل خانة عن الأخرى بثلاث فراغات ، مثال على ذلك العدد 42339 يفصله إلى :

4 2 3 3 9

[استخدم معامل باقي القسمة مع القسمة].

****

****

زايد السعيدي

منتدى الإبداع الإسلامي