سي شارب C# – ملف تمارين وتلخيص

للمبتدئين

اساسيات لغة برمجة – سي شارب – أنواع المتغيرات

نوع المتغير	قيمة المتغير	تعريف المتغير (يمكنكم استخدم اي اسم في التعريف)
int	عدد صحيح	int a; او int a,b,c;
double	عدد عشري	double a; او double a,b,c;
char	حرف او رقم او رمز	char a; او char a,b,c;
string	عبارة نصية	string a; او string a,b,c;
bool	قيمة منطقية true او false	bool a; او bool a,b,c;

ملاحظة: هنالك أنواع متغيرات أخرى لكن سنقوم بالتركيز في شرحنا على المتغيرات التي في الجدول.

امثلة عن تعريف المتغيرات:

```
int num1;
int num2, num3;
char c, d;
string str1, str2;
bool b;
```

امثلة عن اسناد القيم لكل متغير عرفناها سابقا (لا يمكن اسناد القيم للمتغير بدون تعريفه مسبقا):

```
num1 = 5;
num2 = 12;
num3 = 15.4; // هذه العملية خطا
```

يجب اسناد القيم الملائمة لكل متغير حسب نوعه.

```
str1 = “hello”;
c = ”world”; //هذه العملية خطا
c = ‘a’;
b = true;
b = false;
```

اساسيات لغة برمجة – سي شارب – أنواع المتغيرات

نوع المتغير	قيمة المتغير	تعريف المتغير (يمكنكم استخدم اي اسم في التعريف)
int	عدد صحيح	int a; او int a,b,c;
double	عدد عشري	double a; او double a,b,c;
char	حرف او رقم او رمز	char a; او char a,b,c;
string	عبارة نصية	string a; او string a,b,c;
bool	قيمة منطقية true او false	bool a; او bool a,b,c;

ملاحظة: هنالك أنواع متغيرات أخرى لكن سنقوم بالتركيز في شرحنا على المتغيرات التي في الجدول.

تمرين 1

- تعريف عدة متغيرات من أنواع مختلفة.
- اسناد قيم ملائمة لكل متغير حسب نوعه.
- تعريف متغير من نوع عبارة نصية واسناد اسمك الثلاثي بالإنجليزية لهذا المتغير.
- تعريف متغير من نوع عدد عشري ومتغير اخر من نوع عدد صحيح.
- تعريف متغير من نوع قيمة منطقية واسناد القيمة true لهذا المتغير.

اساسيات لغة برمجة – سي شارب – عمليات ال string

توفر لغة ال سي شارب بعض العمليات الجاهزة والخصائص التي تسهل علينا استعمال نوع المتغير string خلال كتابة البرنامج.
في الجدول التالي ستجدون بعض الخصائص والعمليات، تم تعريف متغيران من نوع عبارة نصية string كي يتم استخدامها في شرح الجدول.

```
string str1 = " Hello "; string str2 = " World ";
```

كود العملية/خاصية	وصف العملية/خاصية
str1.**CompareTo**(str2)	عملية مقارنة تعيد 1 اذا لم تتحقق المساواة وتعيد 0 غير ذلك
str1.**Contains**("he")	تفحص العملية اذا كان النص he موجود في str1
str1.**IndexOf**("e")	تعيد العملية رقم الخانة للحرف e
str2.**ToLower**()	تحول العملية النص لحروف صغيرة
str2.**ToUpper**()	تحول العملية النص لحروف كبيرة
str2.**Substring**(2,4)	تعيد العملية نص فرعي من النص الاساسي
str1.**Length**	تعيد هذه الخاصية طول النص

مثال:

```
str1.CompareTo(str2); // Return 1
str1.Contains("he"); // Return true
str1.IndexOf("e"); // Return 2
Console.WriteLine(str2.ToLower()); // print world
Console.WriteLine(str2.ToUpper()); // print WORLD
Console.WriteLine(str2.Substring(2,5)); // print orld
Console.WriteLine(str1.Length); // print 7
```

طباعة عبارة نصية

تتم عملية الطباعة النصية بطريقتين مختلفتين:

1. بالطباعة مع إبقاء المؤشر على نفس السطر والتي نستخدم في هذه العملية :

Console.Write(“عبارة نصية”);

مثال:

ستقوم هذه العملية بطباعة جملة Hello World بدون سطر جديد

Console.Write(“Hello World”);

2. عملية طباعة مع نقل المؤشر الى سطر جديد والتي نستخدم فيها هذه العملية:

Console.WriteLine(“عبارة نصية”);

مثال:

ستقوم هذه العملية بطباعة جملة Hello World مع سطر جديد

Console.WriteLine(“Hello World”);

طباعة قيمة المتغير

يجب تعريف متغير مسبقا واسناد قيمة له ومن ثم ادخال عملية الطباعة للبرنامج.

مثال لمقطع كود يقوم بطباعة نص ورقم من المتغيرات المعرفة:

```
int number = 5;
string str = ”hello”;
string w = “world”;
Console.WriteLine(str); //hello البرنامج يطبع العبارة النصية
Console.WriteLine(number); // 5 البرنامج يطبع رقم
Console.WriteLine(str + “  “ + w); // hello world البرنامج يطبع
```

استقبال معطيات

استقبال معطيات نصية

استقبال العبارات النصية من اجل اسنادها للمتغيرات من نوع string التي عرفناها في البرنامج يجب استخدام العملية التالية:

```
string str = Console.ReadLine();
```

او

```
string str;
str = Console.ReadLine();
```

بأسناد هذه القيم str. البرنامج سيقوم بانتظار المستخدم حتى يدخل له قيم من اجل ان يقوم للمتغير

استقبال معطيات رقمية

```
int age = int.Parse(Console.ReadLine());
```

لا يمكن استخدام عملية **Console.ReadLine();** من اجل استقبال عدد
يجب تحويل هذه القيمة بمساعدة **int.Parse**

مثال عن تعريف متغيرات واستقبال معطيات وطباعتها:

```
int num1,num2;
string str1, str2;

str1 = Console.ReadLine();
str2 = Console.ReadLine();
num1 = int.Parse(Console.ReadLine());
num2 = int.Parse(Console.ReadLine());
Console.WriteLine(str1 + " " + str2);
Console.WriteLine(num1 + " " + num2);
```

بعد تشغيل البرنامج سيقوم الحاسوب بالانتظار المستخدم لادخال اربع قيم، اول قيمتين عبارات نصية واخر قيمتين رقمية.
معطيات المستخدم:Hello world 4 5

```
Console.WriteLine(str1 + " " + str2);// Hello world
Console.WriteLine(num1 + " " + num2); // 4  5
```

تمرين 2

- اطبع كلمة Mathematics ومن ثم في سطر جديد جملة Hello World.

- اطبع كلمة Mathematics ومن ثم في نفس السطر جملة Hello World.

- اطبع اسمك الثلاثي بحيث يكون كل اسم في سطر منفرد.

- قم بتعريف متغيرات من أنواع مختلفة وطباعة قيمتها.

- قم بتعريف نص ورقم واسند قيم لهذه المتغيرات واطبع محتواها في سطر واحد.

تمرين 3

- استقبل معطى تعبير نصي واسنده لمتغير ملائم له واطبع قيمة المتغير.

- استقبل معطى رقمي واسنده لمتغير ملائم له واطبع قيمة المتغير.

- استقبل معطي تعبير نصي وأيضا معطى رقمي واسندهم لمتغيرات ملائمة وقم بطباعة قيمة المتغيرات في نفس السطر مع فراغ بينهم.

- استقبل معطيين من نوع تعبير نصي واسندهم لمتغيرات ملائمة وقم بطباعة قيمة كل متغير في سطر جديد.

اساسيات لغة برمجة – سي شارب – العمليات (الحسابية، المنطقية، المقارنة، الاسناد)

نوع العملية	أداة العملية	الوظيفة
عمليات رياضية	+	جمع
	-	طرح
	*	ضرب
	/	قسمة
	٪	باقي القسمة الصحيحة
عمليات منطقية	&&	و (AND)
	||	او (OR)

نوع العملية	أداة العملية	الوظيفة
عمليات مقارنة	> , <	اصغر ، اكبر
	>=	اكبر او يساوي
	<=	اصغر او يساوي
	!=	لا يساوي
	==	يساوي
عمليات اسناد	=	اسناد
	++ ، --	إضافة او طرح 1
	+= , -=	إضافة او طرح بقيمة معينة
	*= , /=	ضرب او قسمة بقيمة معينة

أداة الشرط

استخدامات أداة الشرط

الهدف من استخدام أداة الشرط هو إضافة عمليات في البرنامج مشروطة في شرط معين، ويتم كتابة عملية الشرط بعدة طرق، وهي:

- **الطريقة الأولى** (اذا تحقق الشرط قم بتنفيذ العملية الداخلية التي بين { }) :

```
if ( a > b){ }
```

- **الطريقة الثانية** (اذا تحقق الشرط قم بتنفيذ العملية الداخلية التي بين { } اذا لم يتحقق نفذ العملية الموجودة في ال else) :

```
if ( a > b){ }
else{ }
```

- **الطريقة الثالثة** (يمكنك إضافة عملية else if قدر ما تحتاج) يقوم البرنامج بتنفيذ العملية تحت الشرط الذي تحقق من بين الشروط التي عينتها اذ لم يتحقق ولا شرط سوف ينفذ العمليات الموجودة تحت else:

```
if ( a > b){ }
else if (a < b){ }
else if (a < 0){ }
else { }
```

مثال 1:

```
int a = 5;
int b = 9;
 if( a < b){ Console.WritLine(“Yes”) }
```

مثال 2:

```
int a = 5;
int b = 9;
 if( a < b){ Console.WritLine(“Yes”) }
 else { Console.WriteLine(“no”); }
```

مثال 3:

```
int a = 5;
int b = 9;
 if( a < b){ Console.WritLine(“Yes”) }
 else if ( a > 0 ) { Console.WriteLine(“pos”);}
 else if ( a < 0 ) { Console.WriteLine(“neg”);}
 else { Console.WriteLine(“no”); }
```

اساسيات لغة برمجة – سي شارب – العمليات واداة الشرط – امثلة

مثال 1 – عمليات حسابية وعمليات اسناد قيم

```
int a,b,c,d;
---------- عملية اسناد ----------
a = 12; b = 6;
c = 8;
----------عمليات حسابية----------
d = a + b; ( d = 18 )
d = a * b; ( d = 72 )
d = a/b; ( d = 2 )
d = a%b; ( d = 0 )
d = a– b; ( d = 6 )
----------عمليات اسناد----------
d--; ( d = 5 )
d++; ( d = 6 )
d+= c; ( d = d + c; d = 14 )
d-= c; ( d = d - c; d = 6 )
```

مثال 2 – عمليات مقارنة وعمليات منطقية

```
int x,y,z;
string str1 = “cond1”; string str2 = “cond2”; string str3 = “cond3”;
---------- عملية اسناد ----------
x = 19; y = -5;
z = 38;
----------عمليات مقارنة وعمليات منطقية----------
// يجب ان يتحقق الشرطين في حال استخدام عملية ال(&& و)
if( x + z < y && z – y < x ){ Console.WriteLine(str1);}
// يجب ان يتحقق شرط واحد على الاقل في حال استخدام عملية ال(|| او)
else if ( x < y  ||  z <= y){Console.WriteLine(str2);}
else {Console.WriteLine(str3);}
```

في هذا المثال لم يتحقق أي شرط من خانة ال If
ولم يتحقق أي شرط من خانة ال Else If
اذا سيقوم البرنامج بطباعة cond3

اساسيات لغة برمجة – سي شارب – العمليات واداة الشرط – اسئلة

تمرين 4

- قم بطباعة مجموع ومضروب عددين تم تعريفهم من نوع int اسند القيم التي تريد لهذه المتغيرات.

- قم بتعريف قيمتين من نوع عدد صحيح int وقم باستقبال معطيات رقمية
واطبع طرح العددين.

- قم بتعريف ثلاث متغيرات من نوع عدد صحيح int واطبع مضروب الثلاث متغيرات معا.

- قم بتعريف متغير من نوع عدد صحيح int واطبع باقي القسمة الصحيحة لهذا المتغير من العدد 10.

تمرين 5

- عرف متغيرين من نوع عدد صحيح int واستقبل معطيات (Console.ReadLine) لهذه المتغيرات من المستخدم واطبع العدد الأكبر بينهم بمساعدة عملية الشرط.

- عرف متغير من نوع عدد صحيح int واستقبل معطى لهذا المتغير من المستخدم(Console.ReadLine) واطبع اذا كان العدد إيجابي او سلبي.

- عزف متغيرين من نوع عدد صحيح int واستقبل قيم من المستخدم لهذه المتغيرات واطبع كلمة pos اذا كان مجموع هذه الاعداد موجب او اطبع neg اذا كان مجموع هذه الاعداد سالب، غير ذلك اطبع zero.

اساسيات لغة برمجة – سي شارب – العمليات المتقدمة

توفر لك C # مجموعة من العمليات الرياضية الجاهزة من أجل القيام بالعمليات الرياضية المتقدمة.
في الجدول التالي، سوف تجد العديد من العمليات الحسابية المتقدمة وكيف بالإمكان استخدامها بمساعدة مكتبة Math:

العملية الحسابية	الوظيفة
X = Math.Abs(Y);	القيمة المطلقة
X = Math.Pow(Y,Z);	القوى
X = Math.PI;	الجذر
X = Math.Log(Y);	عملية الln
X = Math.PI	قيمة PI
X = Math.Max(Y,Z);	القيمة القصوى من بين العددين
X = Math.Min(Y,Z);	القيمة الصغرى من بين العددين
X = Math.Round(Y);	عملية تقريب للعدد الصحيح الاقرب

مثال :

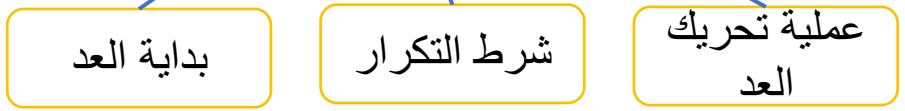
```
double d = 15.76;
int y = -4;
int z = 4;


Console.WriteLine("Abs: " + Math.Abs(y));
Console.WriteLine("Power: " + Math.Pow(y,z));
Console.WriteLine("Sqrt: " + Math.Sqrt(z));
Console.WriteLine("Max: " + Math.Max(y,z));
Console.WriteLine("Min: " + Math.Min(y,d));
Console.WriteLine("Round: " + Math.Round(d));
```

أداة التكرار for

الهدف من استخدام عملية التكرار هي تسهيل عمل البرنامج وكتابة عدد سطور كود اقل وتبسيط عملية البرنامج، المبنى الأساسي للعملية يكون بالشكل التالي:

```
for (int i = 0; i < عدد التكرار; i++){ }
```



يمكننا بدء العد من أي رقم نريد واضافة شروط حسب الطلب وتحريك العداد بعدد الخطوات التي نريد، مثال اولي:

بدل من ان نقوم بكتابة سطرين طباعة لنفس النص يمكننا استخدام عملية التكرار

```
Console.WriteLine("Hello World");
Console.WriteLine("Hello World");
```

نحسن من مقطع الكود اعلاها بهذه الطريقة:

```
for( int i = 0; i < 2; i++){ Console.WriteLine("Hello World"); }
```

مثال :

```
for ( int i = 0; i < 5 ; i++) { Console.WriteLine(i); }// طباعة 1 حتى 5
for ( int i = 5; i > 0 ; i--) { Console.WriteLine(i); }// طباعة 5 حتى 1

for ( int i = 0; i < 20 ; i+=2)
{ Console.WriteLine(i); }//طباعة الاعداد الزوجية حتى الرقم 20

for ( int i = 0; i < 100 ; i+=5)
{ Console.WriteLine(i); }//طباعة الاعداد التي تقسم على 5
```

أداة التكرار While

أداة التكرار هذه تشبه عمل أداة تكرار ال for لكن يتم استخدامها بطريقة مختلفة، تتكون هذه العملية فقط من شرط التكرار وتكتب هكذا:

while (شرط التكرار){ }

تبدأ عملية التكرار وتستمر فقط في تحقق شرط التكرار وتتوقف عند عدم تحقق الشرط وتستمر في باقي أسطر البرنامج.

نستخدم هذه العملية في حال كان مطلوب منا عملية تكرار معين ليست مشروطة في عدد تكرار معين او مشروطة بشرط ما مثل: اذا كان العدد موجب، اذا كان العدد = صفر، اذا كان العدد اكبر من عدد معين ... والعديد من شروط التكرار الأخرى.

مثال :

```
int a = 5;
int b = 3;

while( a < 100 ){
 Console.WriteLine("a < 100" + " " + a);
 a *= b; // b  كل عملية نضرب قيمة المتغير a بقيمة المتغير
}

while( a > 0 ){
 Console.WriteLine("a > 0" + " " + a);
 a -= b; // b  كل عملية نطرح قيمة المتغير a بقيمة المتغير
}
```

اساسيات لغة برمجة – سي شارب – أداة التكرار while/for – اسئلة

تمرين 6

- اطبع مجموع الاعداد من العدد 0 حتى العدد 25.
- اطبع مجموع الاعداد من العدد 15 حتى العدد 25.
- اطبع معدل الاعداد من العدد 0 حتى العدد 10.
- اطبع حاصل ضرب العدد 13 مع الاعداد من 0 حتى 10.
- اطبع الاعداد التي تقسم على 6 والتي بين 0 حتى 1000.
- اطبع الاعداد بين 0 حتى 20 مع القوى 2.

تمرين 7

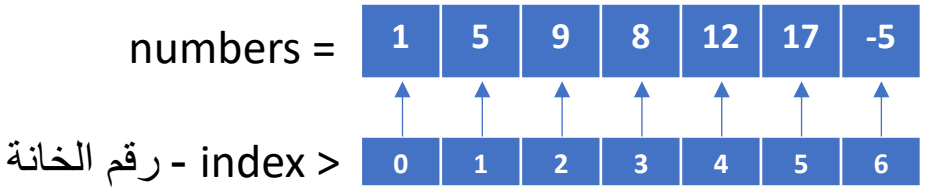
- عرف متغيرين من نوع عدد صحيح واسند لهما قيم العدد الأول أكبر من العدد الثاني، صغر العدد الأكبر بواحد وكبر العدد الاصغر بواحد واطبع الفرق بينهما حتى يكون الفرق مساوة لصفر ومن ثم اطبع كلمة finish.

- عرف عرف متغير من نوع عدد صحيح واسند له العدد 1024 وقسم العدد على 2 في كل مرة، وتوقف عندما يكون العدد يساوي ل 2 او اصغر.

المصفوفات

المصفوفة هي متغير من نوع معين ويمكن اسناد العديد من القيم في الخانات الخاصة بها، يمكن تعريف المصفوفة بعدة طرق ولكل أنواع المتغيرات.

تعريف مصفوفة من نوع int// int[] numbers = {1,5,9,8,12,17,-5}



مثال 1: Console.WriteLine(numbers[0]) // print 1

مثال 2: Console.WriteLine(numbers[4]) // print 12

مثال 3: Console.WriteLine(numbers.length) // print 7 - طول المصفوفة

تعريفات المصفوفة المختلفة:

- **string[] cars;// مصفوفة فارغة ولم يتم تحديد طولها**
- **double[] arr = new double[4];// مصفوفة فارغة بحجم أربع خانات**
- **string[] names = new string[] {“Ahmad”,”Omar”,”Yosif”};**

عمليات طباعة قيم / تغيير قيم / إضافة قيم في المكان الفارغ، نستخدم عملية التكرار for او while حسب الطلب, بهذه الطريقة:

```
int[] integerArray = new int[3];// مصفوفة فارغة بطول 3

int a = 0;
//تعبئة خانات المصفوفة
for (int i = 0; i < integerArray.Length; i++) {
 a += 10;
 integerArray[i] = a;}
//طباعة قيم خانات المصفوفة
for (int i = 0; i < integerArray.Length; i++) {
Console.Write(integerArray[i] + " ");
}
```

اساسيات لغة برمجة – سي شارب – المصفوفات – اسئلة

تمرين 8

- عرف مصفوف فارغة تحتوي 4 عناصر من نوع عدد صحيح، وقم بإسناد معطيات رقمية لكل عنصر بواسطة readline() وطباعة محتوى المصفوفة.

- عرف مصفوفة تحتوي 5 عناصر من نوع عدد صحيح واسند قيمة في كل خانة واطبع قيم الخانات من اخر خانة لأول خانة.

- عرف مصفوفة تحتوي 10 عناصر مع قيم واطبع مجموع خانات المصفوفة.

تمرين 9

- عرف مصفوفة تحتوي 10 عناصر مع قيم واطبع أكبر عدد وأصغر قيمة في المصفوفة.

- عرف مصفوفة تحتوي 10 عناصر مع قيم واطبع أكبر عدد وأصغر قم بالبحث عن قيمة معينة في المصفوفة واطبع رقم الخانة.

- عرف مصفوفة تحتوي 10 عناصر مع قيم واطبع ثاني اكبر قيمة في المصفوفة ورقم الخانة.

اساسيات لغة برمجة – سي شارب – الفئات

تعتبر الفئات مدخل للبرمجة كائنية التوجه والتي نرمز لها باللغة الإنجليزية OOP , وبإمكاننا اعتبار أي شيء هو كائن Object وهي تقريبا شبيه بمبدأ الحياة الحقيقة على الحياة البرمجية، فلكل كائن هنالك خصائص تميزه عن الكائنات الأخرى وافعال (methods) يقوم بها.

لنفترض اننا نريد بناء كائن من نوع طالب، وبمعرفتنا بالحياة اليومية فالطالب يملك رقم هوية خاص به، اسم، عمر، معدل، طول، وزن صف ينتمي اليه واهل والعديد من الخصائص الأخرى ويمكنه القيام بالعديد من الأفعال مثل الجلوس، التقدم للامتحان الحصول على علامة، القراءة، الكتابة، الركض، اللعب .. وعمليات يومية أخرى.

لكي نقوم بأنشاء هذا الطالب يجب في البداية ان نقوم بتعريف صفات وتشيد كائن جديد من نوع طالب مع معطيات الطالب مثل عمره اسمه وزنه ورقم هويته وأيضا بناء العمليات التي نريد ان يقوم بها الطالب في برنامجنا.

مثال

```
class Student {
// ---------- الخصائص ----------
public int id;
public string name;
public int age;
public double weight;
// --------- المشيد: يقوم باستقبال المعطيات لانشاء الكائن ---------
public Student(int id, string name, int age, double weight)
   {
   this.id = id; this. name = name; this. age = age;
   this.weight = weight;
   }
}
```

اساسيات لغة برمجة – سي شارب – الفئات

تعتبر الفئات مدخل للبرمجة كائنية التوجه والتي نرمز لها باللغة الإنجليزية OOP , وبإمكاننا اعتبار أي شيء هو كائن Object وهي تقريبا شبيه بمبدأ الحياة الحقيقة على الحياة البرمجية، فلكل كائن هنالك خصائص تميزه عن الكائنات الأخرى وافعال (methods) يقوم بها.

لنفترض اننا نريد بناء كائن من نوع طالب، وبمعرفتنا بالحياة اليومية فالطالب يملك رقم هوية خاص به، اسم، عمر، معدل، طول، وزن صف ينتمي اليه واهل والعديد من الخصائص الأخرى ويمكنه القيام بالعديد من الأفعال مثل الحضور والغياب، التقدم للامتحان الحصول على علامة، القراءة، الكتابة، الركض، اللعب .. وعمليات يومية أخرى.

لكي نقوم بأنشاء هذا الطالب يجب في البداية ان نقوم بتعريف صفات وتشيد كائن جديد من نوع طالب مع معطيات الطالب مثل عمره اسمه وزنه ورقم هويته وأيضا بناء العمليات التي نريد ان يقوم بها الطالب في برنامجنا.

مثال

```
class Student {
// ---------- الخصائص ----------
public int id;
public string name;
public int age;
public double weight;
// --------- المشيد: يقوم باستقبال المعطيات لانشاء الكائن ---------
public Student(int id, string name, int age, double weight)
   {
   this.id = id; this. name = name; this. age = age;
   this.weight = weight;
   }
}
```

اساسيات لغة برمجة – سي شارب – الفئات

أحيانا نحتاج الى عمليات جاهزة تقوم بتغيير او الوصول الي متغير معين معرف بتعريف افتراضي Private

التعريفات الافتراضية: Private, Public & Protected
Public:يمكن الوصول اليها من أي فئة معرفة في البرنامج.
Private: لا يمكن الوصول اليها الى بمساعدة عملية داخلية للفئة مثل ال get او ال set.
Protected: الوصول لهذه الخاصية او الصفة فقط من خلال الفئات الي ترثها او تتبع لها هذه الفئة التي تملك الخاصية.

عمليات الgetter وال setter تساعدنا في الوصول ال خاصيات معرفة بداخل الفئة، بمساعدة الget نحصل على القيمة الخاصة بالمتغير اما بمساعدة الset نقوم بتغير القيمة الخاصة بالمتغير المعرف التابع لكائن معين.

مثال

```
class Student {
// ---------- الخصائص ----------
public int id; public string name; public int age;
public double weight;
// --------- المشيد: يقوم باستقبال المعطيات لانشاء الكائن ---------
public Student(int id, string name, int age, double weight)
   {
   this.id = id; this. name = name; this. age = age;
   this.weight = weight;
   }
// ----------set and get for id ----------
public string id
{ get { return id; }  set { id = value; } }
}
```

اساسيات لغة برمجة – سي شارب – الفئات – امثلة

```
class Student {
// -------------------- الخصائص --------------
public string name; public int age;
// ---------- المشيد: يقوم باستقبال المعطيات لانشاء الكائن ---------
public Student(string name, int age)
   {
    this. name = name; this. age = age;
   }
// ----------set and get for id --------------
public string age
{ get { return age; }  set { age= value; } }
public string name
{ get { return name; }  set { name= value; } } }
```

```
// -------------------- البرنامج --------------
class Program
{
static void Main(string[] args)
{
Student stu = new Student("Mohamad",9);
Console.WriteLine(stu.name);
Console.WriteLine(stu.age);
stu.name = "Hasan";
stu.age = 22;
Console.WriteLine(stu.name);
Console.WriteLine(stu.age);
}
}
```

في المثال التالي سنقوم بالتعرف على عملية بناء فئة من نوع طالب وتعريف صفة لهذه الفئة وبناء المشيد الخاص فيها واستخدامها بداخل برنامج معين.

وصف المثال:

سينشئ البرنامج كائن من نوع Student مع الأسم Mohamad والعمر 9 ومن ثم يطبع البرنامج في البداية Mohamad والعمر 9 ومن ثم يتم تغيير القيم ل Hasan و22 ويطبع الاسم Hasan والعمر 22.

اساسيات لغة برمجة – سي شارب – الفئات – امثلة

في المثال التالي سنقوم بالتعرف على عملية بناء فئاتين من نوع طالب وصف وتعريف صفات لطالب وبناء فئة الصف مع عملية إضافة طالب لمصفوفة من نوع طلاب.

وصف المثال:

يقوم البرنامج بتشييد طالبيين وصف واضافة الطلاب ال المصفوفة التابعة لصف ومن ثمة طباعة أسماء الطلاب الموجودين في قائمة الصف.

```
//----------فئة الطالب--------
class Student {

public string name;
public Student(string name)
   { this. name = name;  }
public string name
{ get { return name; }  set { id = name; } } }

//----------فئة الصف--------
class Grade {
public int currStudentNumber;
public Student[] stuClass = new Student[10];
public Grade(string name)
   {currStudentNumber = 0; }

public void addStu(Student s){
If (currStudentNumber < 10){
   stuClass[currStudentNumber] = s;
   currStudentNumber++;
}else{ Console.WriteLine("Class is full");}
}
public void printAll()
{for(int i =0; I <currStudentNumber; i++)
 {Console.WriteLine(stuClass[i].name);}
}}
// ---------- البرنامج ----------
class Program
 {
 static void Main(string[] args)
 {Student s1 = new Student("Mohamad");
  Student s2 = new Student("Hasan");
  Grade gra = newGrade();
  gra.addStu(s1);
  gra.addStu(s2);
  gra.printAll();
  }}
```

شكراً لكم على اختياركم لنا وبالتوفيق بأذن الله

بِسْمِ اللَّـهِ الرَّحْمَـٰنِ الرَّحِيمِ

{ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِير }

هذا الملف تم انشاؤه لتقديم المساعدة وتسهيل عملية التعلم عليكم، في حال وجدتم خطا ما يرجى التواصل معنا