

برمجة مهيكلة/العملي

قسم هندسة الحاسوبات/المراحل الأولى

جامعة الشطرة
كلية الهندسة/قسم هندسة الحاسوبات

ما هي لغة C++

تكتب سي بلس بلس باللغة العربية و هي لغة برمجة تجميعية (Compiled) و كائنية (Object Oriented) تضم العديد من مميزات لغات البرمجة عالية المستوى (High Level) و منخفضة المستوى (Low Level) و هذا يعني أنك قادر على فعل ما تشاء بها. هذه اللغة الرائعة مصممة لتعمل على جميع أنظمة التشغيل المعروفة مثل ويندوز، ماك، لينكس، إندرويد إلخ.. و هي تعتبر من أقوى و أشهر لغات البرمجة على الإطلاق و يطلق عليها "أم اللغات" نظراً لأنها لغة قوية و موجودة منذ زمن طويل.

تم بناء هذه اللغة بالأساس كتطوير لغة C من قبل Bjarne Stroustrup أثناء عمله في مختبرات Bell لتكون أطروحته في رسالة الدكتوراه عام 1979.

من ذلك الحين و حتى وقتنا الحالي تم تطوير هذه اللغة بشكل مستمر و اخر إصدار رسمي لها هو C++ 17. لا يوجد شعار رسمي للغة C++ ولكن في العادة و عند البحث عن دورات لتعلمها أونلاين فإنك تجدهم يستخدموا شعار يشبه التالي:



ماذا يمكنني ان اطور بلغة C++

لغة C++ تعتبر لغة عامة الهدف (General Purpose) مما يعني أنها قادرة على بناء أي برنامج.

القدرة الحقيقة لها تتجلى في المشاريع و البرامج الكبيرة مثل:

- بناء و تطوير أنظمة التشغيل ومن أشهر أنظمة التشغيل المبنية بهذه اللغة Windows, Linux, MacOS, Android.
- بناء و تطوير البرامج الكبيرة مثل برامج أدوبى (مثل Firefox و Premier و Photoshop) و المتصفح .
- بناء و تطوير الألعاب و من أشهر الألعاب التي طورت بهذه اللغة Counter Strike, MacOSDoom, Warcraft.

مميزات لغة C++ عن باقي لغات البرمجة

1. مفتوحة المصدر

لن تدفع أي مبلغ لتعلم على لغة C++ فهي مصدر مفتوح و مجانية و ستبقى مجانية مدى الحياة.

2. خفيفة

تستطيع البرمجة بها حتى و لو كان حاسوبك ضعيفاً أو قديماً.

3. البساطة

تعلمها سهل جداً بعد أن تفهم برمجة الكائنات.

4. سرعة الترجمة

تتم ترجمة لغة C++ إلى أوامر يفهمها الجهاز بشكل سريع جداً.

5. لغة كائنية (Object Oriented)

مع أن لغة C++ صممت لتكون قريبة من عتاد الجهاز و قريبة للغات منخفضة المستوى إلا أنها تتيح لمستخدمها بناء الكلاسات و التعامل معها بطرق منهجيات البرمجة الكائنية (OOP) مثل الوراثة و تعدد الأشكال و غيرها من المبادئ التي ستتعلمها لاحقاً في الدورة.

6. متعددة النماذج

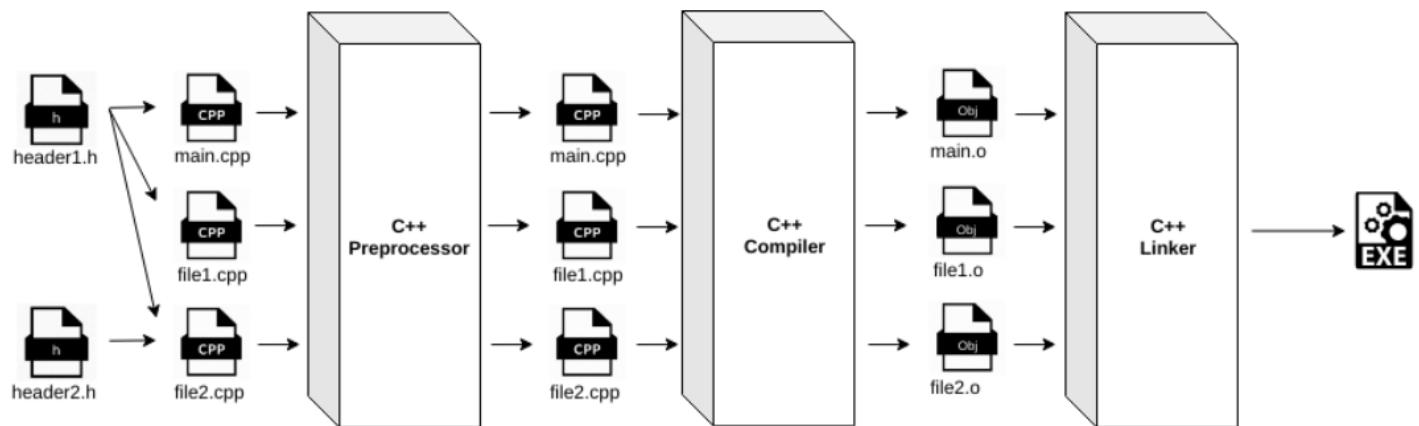
تسمح لغة C++ لمستخدمها باختيار الطريقة المناسبة له لكتابة الكود البرمجي من بين عدة نماذج مختلفة مثل النموذج الهيكلي (Structural) الشبيهة بلغة C و النموذج الكائني (Object Oriented) الشبيه بلغة جافا.

7. الدعم الكبير

لغة C++ من اللغات القديمة نسبياً و المنتشرة بشكل كبير مما جعل منها لغة ذات شعبية كبيرة و مصادر تعلم مختلفة و متعددة و دعم كبير من قبل مجتمع المطوريين.

طريقة عمل برنامج مكتوب بلغة C++

الكود الذي تكتبه على الكمبيوتر لا يعمل بشكل مباشر بل يمر بعدة مراحل تباعاً حتى يعمل تماماً كما في الصورة التالية.



إذا يمر الكود المكتوب بلغة C++ بثلاث مراحل حتى يصبح في النهاية برنامج يمكن تشغيله على الحاسوب.

في البداية يقوم الـ **preprocessor** بتشذيب وتجهيز الملفات النصية لتدخل بعدها في عملية التجميع (**Compiling**) حيث يتم تحويل الملفات النصية الى ملفات (نوعها **Binary**) يفهمها الحاسوب مع المحافظة على بعض صفات الكود مثل أسماء المتغيرات والدوال ومن ثم يقوم الموصل (**Linker**) بوصل كل أجزاء البرنامج المختلفة ودمجها مع بعضها لتصبح ملف تنفيذي واحد (نوعه **EXE**) يمكن تشغيله في أي وقت مثل أي تطبيق عادي.

معلومات تقنية

الـ **Linker**, الـ **Compiler**, و الـ **preprocessor** هي مجرد برامج صغيرة تعمل مع بعضها بشكل متناسق حتى تحول كل الكود الذي قمنا بكتابته في المشروع لبرنامج عادي يمكن تشغيله بنقرة واحدة.

تجهيز بيئة العمل لتطوير تطبيقات بلغة C++

يوجد الكثير من بيئات العمل التي تمكّنك من العمل على تطوير البرامج بلغة C++ مهما كان نظام التشغيل الذي تستخدّمه مثل:

Dev C++ - CLion - CodeBlocks - Eclipse . الخ.

الشكل العام لأي برنامج مكتوب بلغة C++

الملف الأساسي في المشروع يجب أن يكون شكله كالتالي:

```
#include<iostream>
using namespace std;
int main()
{
    return 0;
}
```

هنا يجب ان تضع الاوامر التي سوف تتنفذ عند تشغيل البرنامج

أهمية استخدام المكاتب في لغة C++:

:#include<iostream>

تساعد المستخدم(المبرمج) في ادخال البيانات للبرنامج عند التنفيذ من لوحة المفاتيح وطباعة النتائج على شاشة التنفيذ.

: std

مكتبة تحتوي على الاوامر والدوال والانواع البيانية التي تحتاجها في كتابة أي برنامج مثل:

1- اوامر الادخال Cin واوامر الارχاج(Cout الطباعة).

2- الانواع البيانية مثل Int ,Char,String,Float

امر الطباعة Cout

مبادئ الطباعة الأساسية

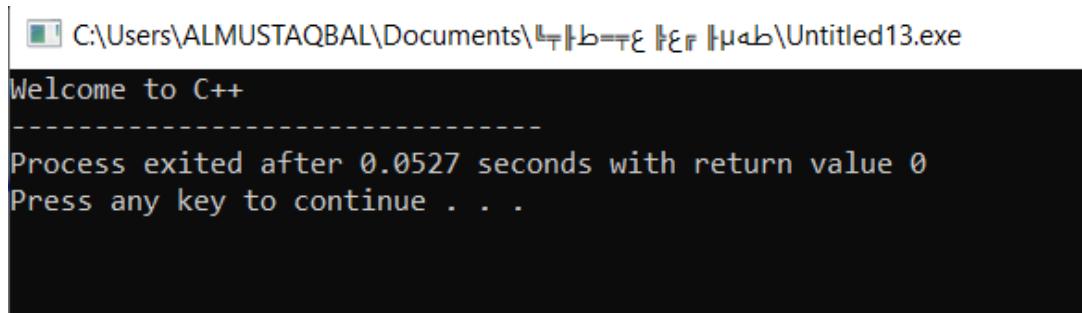
عليك مراعاة المبادئ التالية عند استخدام الأمر Cout :

- لعرض رقم, ضعه كما هو.
- لعرض قيمة متغير, ضعه كما هو.
- لعرض حرف يجب وضعه بين . ' ' .
- لعرض كلمة أو نص, يجب وضعه بين . " " .

مثال 1/ برنامج بلغة C++ يستخدم لطباعة عبارة Welcome to c++:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Welcome to C++";
    return 0;
}
```

نتائج تنفيذ البرنامج:



```
C:\Users\ALMUSTAQBAL\Documents\٢٠١٤\٢٠١٥\Untitled13.exe
Welcome to C++
-----
Process exited after 0.0527 seconds with return value 0
Press any key to continue . . .
```

- توضع الفارزة المنقوطة (;) في نهاية كل سطر برمجي للدلالة على نهاية السطر البرمجي.
- كل نص يوضع داخل علامة اقتباس مزدوجة (" ") يطبع كما هو بدون تغيير.

مثال 2/ برنامج بلغة C++ يستخدم لطباعة الحرف **z**:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<'z';
    return 0;
}
```

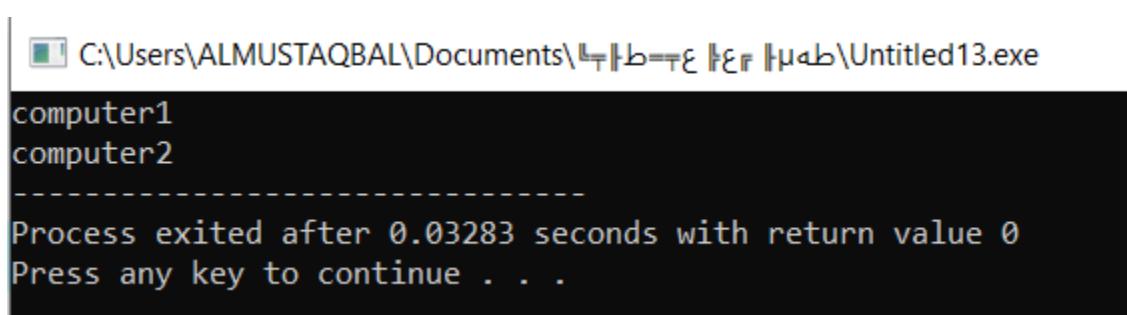
مثال 3/ برنامج بلغة C++ يستخدم لطباعة الرقم **100**:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<100;
    return 0;
}
```

مثال 4: برنامج بلغة C++ يستخدم لطباعة عبارة computer1 في السطر الاول وعبارة computer2 في السطر الثاني:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"computer1"<<endl;
    cout<<"computer2";
    return 0;
}
```

نتائج تنفيذ البرنامج:



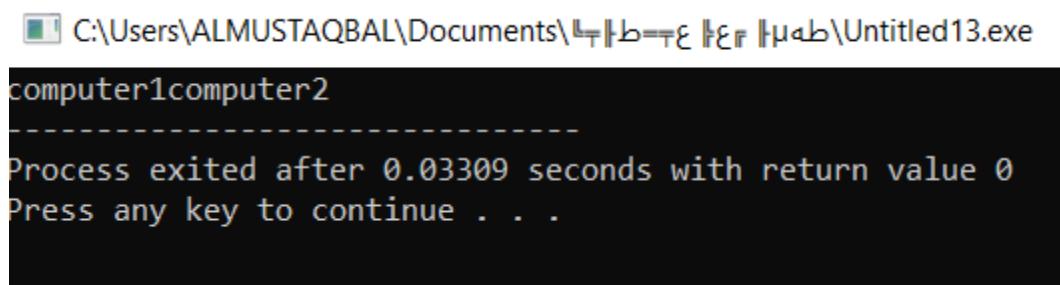
`.endl` هی اختصار لـ `endl`

. New Line ممكن استخدام الرمز ("\\n") للنزول لسطر جديد وهو اختصار لـ

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"computer1"<<"\n";
    cout<<"computer2";
    return 0;
}
```

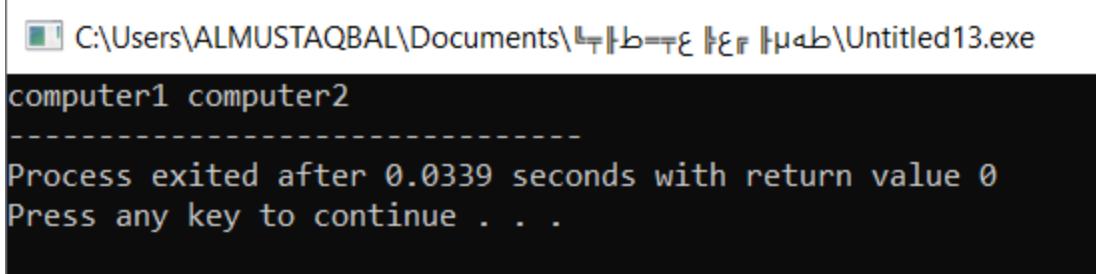
مثال 5: برنامج بلغة C++ يستخدم لطباعة عبارة computer1 وعبارة computer2 في سطر واحد:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"computer1"<<"computer2";
    return 0;
}
```



نلاحظ انه قمنا بطباعة العبارتين على سطر واحد لكن العبارة الاولى هي مجاورة للعبارة الثانية وبذلك كأنما اصبحت عبارة واحدة وليس عبارتين لذلك نحتاج الى شيء اضافي على الكود بحيث يضع مساحة فارغة (space) بين العبارتين من خلال طباعة المساحة الفارغة بينهما:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"computer1 "<<" " <<"computer2";
    return 0;
}
```



ممكن استخدام الرمز ("\\t") لاضافة مساحة فارغة وهو اختصار لـ Tab Space

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"computer1 "<<'\t'<<"computer2";
    return 0;
}
```

إصدار صوت تنبيه بواسطة الرمز \a في C++:

إذا أردت إصدار صوت تنبيه لجعل المستخدم ينتبه لشاشة التنفيذ، يمكنك ذلك بكل سهولة من خلال

وضع الرمز "\a" كنص للأمر cout وهو اختصار لـ Alert.

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"computer1"<<"\t"<<"computer2"<<"\a";
    return 0;
}
```

مفهوم العوامل في C++

العوامل (operators) عبارة عن رموز لها معنى محدد، ويمكننا تقسيم العوامل إلى 5مجموعات أساسية كالتالي:

العوامل الرياضية (Arithmetic Operators)

إسم العامل	رمزه	مثال	شرح الكود
Assignment	=	a = b	أعطي قيمة b لـ a
Addition	+	a + b	أضف قيمة b على قيمة a
Subtraction	-	a - b	إطرح قيمة b من قيمة a
Unary plus	+	+a	أضرب قيمة a بالعامل +
Unary minus	-	-a	أضرب قيمة a بالعامل -
Multiplication	*	a * b	أضرب قيمة a بقيمة b
Division	/	a / b	أقسم قيمة a على قيمة b
Modulo	%	a % b	للحصول على آخر رقم يبقى عندما نقسم قيمة a على قيمة b
Increment	++	a++	إضافة 1 على قيمة a و تستخدم في الحلقات
Decrement	--	a--	إنقصاص 1 من قيمة a و تستخدم في الحلقات

عوامل المقارنة : (Comparison Operators)



إسم العامل	رمزه	مثال	شرح الكود
Equal to	<code>==</code>	<code>(a == b)</code>	هل قيمة <code>a</code> تساوي قيمة <code>b</code> ؟ إذا كان الجواب نعم فإنها ترجع <code>true</code>
Not equal to	<code>!=</code>	<code>(a != b)</code>	هل قيمة <code>a</code> لا تساوي قيمة <code>b</code> ؟ إذا كان الجواب نعم فإنها ترجع <code>true</code>
Greater than	<code>></code>	<code>(a > b)</code>	هل قيمة <code>a</code> أكبر من قيمة <code>b</code> ؟ إذا كان الجواب نعم فإنها ترجع <code>true</code>
Less than	<code><</code>	<code>(a < b)</code>	هل قيمة <code>a</code> أصغر من قيمة <code>b</code> ؟ إذا كان الجواب نعم فإنها ترجع <code>true</code>
Greater than or Equal to	<code>>=</code>	<code>(a >= b)</code>	هل قيمة <code>a</code> أكبر أو تساوي قيمة <code>b</code> ؟ إذا كان الجواب نعم فإنها ترجع <code>true</code>
Less than or Equal to	<code><=</code>	<code>(a <= b)</code>	هل قيمة <code>a</code> أصغر أو تساوي قيمة <code>b</code> ؟ إذا كان الجواب نعم فإنها ترجع <code>true</code>

العوامل المنطقية : (Logic Operators)



إسم العامل	رمزه	مثال	شرح الكود
AND	&&	(a && b)	هل قيمة <code>true</code> و <code>b</code> تساويان ؟ هنا يجب أن يتم تحقيق الشرطين ليرجع <code>true</code>
OR		(a b)	هل قيمة <code>a</code> أو <code>b</code> أو كلاهما تساويان ؟ هنا يكفي أن يتم تحقيق شرط واحد من الشرطين ليرجع <code>true</code>
NOT	!	!a	هل قيمة <code>a</code> لا تساوي <code>true</code> ؟ إذا كان الجواب نعم فإنها ترجع <code>true</code>

خطوات كتابة برنامج متكامل متكون من عمليات الادخال والاخراج

1-تعريف او التصريح عن المتغيرات التي نحتاجها في البرنامج:

*مفهوم المتغيرات في C++:

المتغيرات (variables) عبارة عن أماكن يتم حجزها في الذاكرة بهدف تخزين بيانات فيها أثناء تشغيل البرنامج.

*أنواع البيانات في C++:

- 1 - int : يستخدم هذا النوع لتعريف عدد صحيح, أي عدد لا يحتوي على فاصلة عشرية مثل:

int x=5;

- 2 - float : يستخدم هذا النوع لتعريف عدد يمكن أن يحتوي على فاصلة عشرية . يمكن لهذا العدد أن يحتوي على 7 أرقام بعد الفاصلة مثل:

Float x=10.5;

- 3 - double : يستخدم هذا النوع لتعريف عدد يمكن أن يحتوي على فاصلة عشرية . يمكن لهذا العدد أن يحتوي على 15 رقم بعد الفاصلة مثل:

double x=20.6 ;

- 4 - char : يستخدم هذا النوع لتعريف حرف اجنبي مثل:

Char x= 'a' ;

- 5 - string : يستخدم هذا النوع لتعريف سلسلة رمزية(مجموعة حروف) مثل:

String x="mohammed";

الأحرف المستخدمة في وضع الأسماء في C++:

* أي إسم نضعه لمتغير، دالة، كلاس، كائن إلخ.. يسمى identifier في البرمجة.

* إذ يتم التمييز بين العناصر في C++ من خلال أسمائهم، أي من خلال الـ Identifiers.

قواعد الزامنية عند اعطاء الأسماء:

- البدء بحروف كبيرة مثل (A-Z) أو حروف صغيرة (a-z) أو الشرطة (_).

- يمنع البدء ب رقم.

- يمنع استخدام القيم true او false .

- يمنع استخدام الكلمات المحفوظة (Keywords) الموضحة بالأسفل ↓

- لغة C++ تطبق مفهوم الـ Case Sensitivity اي ان الحرف a يختلف عن الحرف A اي ليسوا متغير واحد.

C++ الكلمات المحفوظة في

جميع الكلمات التالية محفوظة لغة C++. أي لا يمكن استخدامها ك Identifiers .

alignas	char16_t	double	if	or	static
alignof and	char32_t	dynamic_cast	inline	or_eq	static_assert
and_eq	class	else	int	private	static_cast
auto	compl	enum	long	protected	struct
bitand	const	explicit	mutable	Public	switch
bitor	constexpr	export	namespace	register	template
bool	const_cast	extern	new	reinterpret_cast	this
break	continue	false	noexcept	requires	thread_local
case	decltype	float	not	return	throw
catch	default	for	not_eq	short	true
char	delete	friend	nullptr	signed	try
	do	goto	operator	sizeof	typedef

```
typeid
typename
union
unsigned
using
virtual
void
volatile
wchar_t
while
xor
xor_eq
```

اذن بالنتيجة النهائية كل متغير مستخدم بالبرنامج يجب ان يعرف :

```
int a;
int b;
float sum;
```

ممكن ان نعرف المتغيرات التي تكون من نفس النوع البياني في سطر واحد:

```
int a,b;
float sum;
```

2-عملية ادخال او قراءة قيم او بيانات للمتغيرات :

بعد تعريف او التصريح عن المتغيرات نحتاج الى ادخال قيم لهذه المتغيرات وتنتمي بطريقتين:

-**الطريقة الاولى (الادخال المباشر للمتغير)** وتتم اثناء (كتابة البرنامج) وتكون بصورتين:

```
int x=5;
```

اي اثناء التصريح في الخطوة الاولى عن المتغير تعطى له القيمة مباشرة.

```
int x;
```

```
x=5;
```

اي بعد التصريح عن المتغير في الخطوة الاولى يتم بعدها كتابة اسم المتغير واسناد له قيمة مباشرة.

-**الطريقة الثانية (الادخال غير المباشر للمتغير)** وتتم من خلال (شاشة التنفيذ) وتكون بصورتين ايضاً:

```
Cin>>x;
```

```
Cin>>y;
```

هنا الادخال على اكثر من سطر.

```
Cin>>x>>y;
```

هنا الادخال على سطر واحد فقط.

3-عملية المعالجة(اي كتابة المعادلات الرياضية) :

هنا نكتب العملية التي نحتاجها مثلاً نحتاج جمع عددين نكتب المعادلة التي من خلالها يتم جمع العددين $x+y$ نحن نعلم هذه المتغيرات المفروض تم تعريفها وادخال قيم لها لذلك نحتاج لاظهار ناتج هذى المعادلة او العملية الرياضية فنحتاج الى متغير اخر يكون مخزن لنتائج جمع x مع y فتكون بالشكل التالي:

$$z=x+y;$$

لنفترض قيمة $x=5$ و $y=5$ اذن قيمة $z=10$

4- الاخراج او الطباعة للقيم :

تطرقنا في المحاضرات السابقة للقواعد الخاصة بعملية الطباعة بالتفصيل.

$$z=x+y;$$

لنفترض انه لدينا

اذن نطبع قيمة المتغير Z والذي يمثل ناتج عملية الجمع بالشكل التالي:

$$\text{Cout} << z;$$

مثال(1)/برنامج بلغة C++ يستخدم لادخال قيمة متغيرين ويجد ناتج جمعهما ثم يقوم بطباعة ناتج عملية الجمع:

- هنا نلاحظ انه لدينا عملية ادخال قيم وايضا عملية جمع وعملية طباعة ناتج اذن نحتاج الى المضي في الخطوات الاربعة :

1-تعريف او التصريح عن المتغيرات المراد استخدامها في البرنامج.

2-عملية ادخال قيم او قراءة قيم للمتغيرات.

3-المعالجة والتي تمثل كتابة العمليات الرياضية.

4-عملية الارχاج او الطباعة.

```
#include<iostream>
using namespace std;
int main()
{
    int x,y,z;
    cin>>x;
    cin>>y;
    z=x+y;
    cout<<z;
    return 0;
}
```

-ممكن ان يكون الادخال على نفس السطر للبرنامج اعلاه وبالشكل التالي:

```
Cin>>x>>y;
```

يمكن وضع اي عبارة او نص بحيث تكون علامة للشيء المراد قراءته او طباعته :

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z;
cout<<"Enter Value x y:"<<"\n";
cin>>x>>y;
z=x+y;
cout<<"sum="<<z;
return 0;
}
```

مثال(2) / برنامج بلغة C++ يستخدم لادخال عددين من لوحة المفاتيح ويجد ناتج قسمة العددين ثم يقوم بطباعة ناتج عملية القسمة:

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z;
cout<<"Enter Value x y:"<<"\n";
cin>>x>>y;
z=x/y;
cout<<"x/y="<<z;
return 0;
}
```

مثال(3)/ برنامج بلغة c++ يستخدم لضرب العددين val1=10 , val2=20 ثم يقوم بطباعة الناتج:

```
#include<iostream>
using namespace std;
int main()
{
int val1,val2;
int mul;
val1=10;
val2=20;
mul=val1*val2;
cout<<"mul="<<mul;
return 0;
}
```

مثال(4)/ برنامج بلغة c++ يستخدم لضرب وقسمة وجمع وطرح ويجد باقى القسمة بين المتغيرين المستخدم هو الذى يقوم بادخال قيم المتغيرين:

```
#include<iostream>
using namespace std;
int main()
{
int x,y;
int mul,div,sum,sub,mod;
cout<<"Enter Value :"<<"\n";
cin>>x>>y;
mul=x*y;
div=x/y;
sum=x+y;
```

```

sub=x-y;
mod=x%y;
cout<<"mul="<<mul<<"\n";
cout<<"div="<<div<<"\n";
cout<<"sum="<<sum<<"\n";
cout<<"sub="<<sub<<"\n";
cout<<"mod="<<mod;

return 0;
}

```

مثال(5) / برنامج بلغة C++ يستخدم لحساب محيط ومساحة المستطيل:

```

#include<iostream>
using namespace std;
int main()
{
int length,width;
int area, perimeter;
cout<<"Enter Value :"<<"\n";
cin>>length>>width;
perimeter=2*( length+ width);
area = length * width;
cout<<" perimeter ="<< perimeter<<"\n";
cout<<" area ="<< area;
return 0;
}

```

واجب / برنامج بلغة C++ يستخدم لحساب مساحة المربع ؟؟

مثال 6 / برنامج بلغة C++ يستخدم لحساب مساحة الدائرة ومحيطها ؟؟

علما ان نصف القطر (r) يساوي 5.2 وان النسبة الثابتة $\pi=3.14$

```
#include<iostream>
using namespace std;
int main()
{
float r,pi;
float area, perimeter;
r=5.2;
pi=3.14;
perimeter=2*pi*r;
area = pi*r*r;
cout<<" perimeter ="<< perimeter<<"\n";
cout<<" area ="<< area;
return 0;
}
```

ملاحظة / في C++ عندما نكتب r^2 نكتبه بالشكل $r*r$

مثال 7 / برنامج بلغة C++ يستخدم لادخال ثلاث درجات لطالب ثم يقوم بآيجاد مجموع الدرجات ومعدل الطالب :

```
#include<iostream>
using namespace std;
int main()
{
int mark1,mark2,mark3;
int sum;
float avg;
cout<<"Enter Value :"<<"\n";
cin>>mark1>>mark2>>mark3;
sum= mark1+mark2+mark3;
avg=sum/3;
cout<<" sum ="<< sum<<"\n";
cout<<" avg ="<< avg;
return 0;
}
```

مثال 8 / برنامج بلغة C++ يقوم بالتبديل بين قيمة متغيرين:

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z;
cout<<"Enter Value :"<<"\n";
cin>>x>>y;
z=x;
x=y;
y=z;
cout<<" x ="<< x<<"\n";
cout<<" y ="<< y;
return 0;
}
```

مثال 9/ برنامج بلغة C++ يقوم بحساب ناتج المعادلة:

$$z = x^2 + \frac{y}{2}$$

الحل //

ملاحظة/ x^2 في برنامج C++ تتفك إلى $x*x$. كذلك الحال اذا كانت x^3 فأنها تتفك إلى $x*x*x$ وهذا.....

```
#include<iostream>
using namespace std;

int main()
{
    int x,y,z;
    cout<<"Enter Value :"<<"\n";
    cin>>x>>y;
    z=(x*x)+(y/2);
    cout<<" z ="<< z;
    return 0;
}
```

واجب 1 / برنامج بلغة C++ يقوم بحساب ناتج المعادلة :

$$z=x+1/y+2$$

واجب 2 / برنامج بلغة C++ يقوم بحساب ناتج المعادلة :

$$e = \frac{2xy}{(x+y)} + \frac{2x}{2(x-z)}$$

الجمل الشرطية

أنواع الجمل الشرطية:

1- الجملة الشرطية if و if ..else و else if

1- الجملة الشرطية if ذات المسار الواحد:

(if) في اللغة العربية تعني "إذا" ، و هي تستخدم فقط في حال كنت تريدين تنفيذ كود معين (مثلاً عملية رياضية او طباعة الخ....) حسب شرط معين.

If(الشرط)

; التعليمية البرمجية التي يتم تنفيذها اعتماداً على تحقق الشرط

مثال 1 / برنامج بلغة C++ يستقبل عدد معين ويقوم بطباعة big اذا كان العدد المدخل اكبر من 10.

ملاحظات:

-كلمة (يستقبل) تعني ان الادخال يكون غير مباشر اي من شاشة التنفيذ.

-نلاحظ ان طباعة big مشروطة فقط اذا كان العدد المدخل اكبر من 10 لذلك نستخدم if ذات المسار الواحد.

```
#include<iostream>
using namespace std;
int main()
{
    int x;
    cout<<"enter value for x:"<<"\n";
    cin>>x;
    if(x>10)
        cout<<"big";
    return 0;
}
```

مثال2/ برنامج بلغة C++ يستقبل عدد معين ويقوم بطباعة big اذا كان العدد المدخل اكبر او يساوي 10.

ملاحظة:

لو تلاحظ في المثال السابق لو ادخلنا 10 لا يطبع شيء لأن أداة الشرط ($x >= 10$) تطلب فقط الذي هو أكبر من 10 . لكن في هذا المثال لو ادخلنا 10 سوف يطبع لنا شيء لأن أداة الشرط تكون أكبر او تساوي 10 ($= 10$).

```
#include<iostream>
using namespace std;
int main()
{
    int x;
    cout<<"enter value for x:"<<"\n";
    cin>>x;
    if(x>=10)
        cout<<"big";
    return 0;
}
```

2-الجملة الشرطية if....else ذات المسارين:

(else) في اللغة العربية تعني " اي شيء اخر", يجب وضعها دائماً في الأخير, لأنها تستخدم في حال لم يتم تنفيذ أي جملة شرطية قبلها.

If(الشرط)

; التعليمية البرمجية 1 التي يتم تنفيذها اعتماداً على تحقق الشرط

else

; التعليمية البرمجية 2 التي يتم تنفيذها في حالة عدم تتحقق الشرط

مثال3/ برنامج بلغة C++ يستخدم لادخال عدد ويقوم بطباعة big اذا كان العدد اكبر من 10 وطباعة small اذا كان اصغر من 10.

```
#include<iostream>
using namespace std;
```

```

int main()
{
int x;
cout<<"enter value for x:"<<"\n";
cin>>x;
if(x>10)
cout<<"big";
else
cout<<"small";
return 0;
}

```

مثال 4

اكتب برنامج يستقبل عدد من المستخدم ثم يطبع negative إذا كان العدد سالب ويطبع positive إذا كان العدد موجب ؟

```

#include<iostream>
using namespace std;
int main()
{
int i;
cout<<"enter value for i:"<<"\n";
cin>>i;
if(i<0)
cout<<"negative";
else
cout<<"positive";
return 0;
}

```

مثال 3/ برنامج بلغة c++ يستخدم لادخال عدد معين ثم يختبر هذا العدد فإذا كان زوجي يطبع even و اذا كان فردي يطبع odd.

```

#include<iostream>
using namespace std;
int main()
{
int i;

```

```

cout<<"enter value for i:"<<"\n";
cin>>i;
if(i%2==0)
cout<<"even";
else
cout<<"odd";
return 0;
}

```

مثال4/ برنامج بلغة C++ يستخدم لادخال درجة طالب في مادة معينة ثم يختبر هل طالب ناجح ام راسب.

```

#include<iostream>
using namespace std;
int main()
{
int mark;
cout<<"Enter Mark:"<<"\n";
cin>>mark;
if(mark>=50)
cout<<"Successful";
else
cout<<"Unsuccessful";
return 0;
}

```

مثال5/ برنامج بلغة C++ يستخدم لادخال عددين ثم يطبع العدد الاكبر بينهما.

```

#include<iostream>
using namespace std;
int main()
{
int val1,val2;
cout<<"Enter values:"<<"\n";
cin>>val1>>val2;

```

```

if(val1>val2)
cout<<"The MAX="<

مثال 6/ برنامج بلغة C++ يستخدم لادخال عدد معين ثم يطبع العدد اذا كان ضمن الاعداد المحسورة بين 1 الى 10 والا يطبع . nothing


```

```

#include<iostream>
using namespace std;
int main()
{
int x;
cout<<"Enter your Value:"<<"\n";
cin>>x;
if(x>=1 && x<=10)
cout<<x;
else
cout<<"nothing";
return 0;
}

```

مثال 7/ برنامج بلغة C++ يستخدم لطباعة الالوان black,yellow,green على اكثر من سطر اذا المستخدم قام بادخال رقم 1 ويطبع السنوات 2022,2023,2024 على اكثر من سطر اذا قام المستخدم بادخال اي رقم اخر .

ملاحظة: اذا كان هناك اكثر من سطر برمجي داخل if نحتاج الى اقواس لكي يتم السيطرة من if على كل الاسطرا البرمجية الموجودة داخلها وهذه الحال ينطبق على كل من العبارات الشرطية والحلقات التكرارية.

```

#include<iostream>
using namespace std;
int main()
{

```

```

int x;
cout<<"Enter Value:"<<"\n";
cin>>x;
if(x==1)
{
cout<<"black"<<"\n";
cout<<"yellow"<<"\n";
cout<<"green"<<"\n";
}
else
{
cout<<2022<<"\n";
cout<<2023<<"\n";
cout<<2024<<"\n";
}

```

3-الجملة الشرطية if else متعددة المسارات:

جملة **else if** تستخدم إذا كنت تريد وضع أكثر من إحتمال (أي أكثر من شرط) .

جملة أو جمل **else if** يوضعون في الوسط, أي بين الجملتين **if** و **else**.

```

If(1) الشرط1
;التعليمية البرمجية1
else
If(2) الشرط2
;التعليمية البرمجية 2
else
If(3) الشرط3
;التعليمية البرمجية 3
:
:
else
;التعليمية البرمجية n

```

مثال 8/ برنامج بلغة **c++** يقوم بأخذ ثلاثة اعداد ويختبر ايهمما العدد الاكبر وطباعة العدد الاكبر والا .**nothing** يطبع

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z;
cout<<"Enter x,y,z:"<<"\n";
cin>>x>>y>>z;
if(x>y && x>z)
cout<<"The Max="<<x;
else
if(y>x && y>z)
cout<<"The Max="<<y;
else
if(z>x && z>y)
cout<<"The Max="<<z;
else
cout<<"nothing";
return 0;
}
```

مثال 9//اكتب برنامج بلغة **c++** لادخال مدخل طالب وطباعة تقدير الطالب حيث يطبع التالي:

Excellent: اذا كان المعدل داخل 90

Very Good: اذا كان المعدل داخل 80

Good: اذا كان المعدل داخل 70

Middle: اذا كان المعدل داخل 60

Accepted: اذا كان المعدل داخل 50

```

#include<iostream>
using namespace std;
int main()
{
int avg;
cout<<"Enter avg:<<"\n";
cin>>avg;
if(avg>=90)
cout<<"Excellent";
else
if(avg>=80)
cout<<"Very Good";
else
if(avg>=70)
cout<<"Good";
else
if(avg>=60)
cout<<"Middle";
else
if(avg>=50)
cout<<"Accepted";
else
cout<<"Fail";
return 0;
}

```

واجب 1/ برنامج بلغة C++ يستخدم لادخال حرف معين عدد معين ويقوم بطباعة green اذا كان الحرف المدخل g وطباعة yellow اذا كان الحرف المدخل y.

واجب 2/ برنامج بلغة C++ يقوم بأدخال ثلث اعداد ويختبر ايهمما العدد الاصغر وطباعة العدد الاصغر والا يطبع .nothing

2-الجملة الشرطية Switch

نستخدمها إذا كنا نريد اختبار قيمة متغير معين مع لائحة من الإحتمالات نقوم نحن بوضعها، و إذا تساوت هذه القيمة مع أي إحتمال وضعناه ستتتفذ الأوامر التي وضعناها في هذا الإحتمال فقط.

كل إحتمال وضعه يسمى **case**.

(المتغير المراد اختبار قيمته) switch

```
{  
    case قيمة1:  
        التعليمية البرمجية 1  
        break;  
  
    case قيمة2:  
        التعليمية البرمجية 2  
        break;  
    :  
    :  
  
    default:  
        تعليمية برمجية  
}  
:
```

مثال 10/ برنامج يستقبل عدد من المستخدم بين 1 و 4 ويطبع بالحروف، ويطبع العدد خارج المدى .(switch) اذا لم يكن العدد المدخل بين الاعداد 1 و 4 باستخدام (out of range)

```
#include<iostream>  
using namespace std;  
int main()  
{  
    int x;  
    cin>>x;  
    switch(x)  
    {  
        case 1:cout<<"one";break;  
        case 2:cout<<"two";break;  
        case 3:cout<<"three";break;  
        case 4:cout<<"four";break;  
        default:cout<<"out of range";  
    }
```

```
}

return 0 ;
}
```

مثال 11 / برنامج بلغة C++ يستخدم لتصميم الالة الحاسبة حيث الادخال الاول عبارة عن عددين والادخال الثاني يكون للعملية المراد تنفيذها من العمليات التالية (+,-,*) بستخدام (switch).

```
#include<iostream>
using namespace std;
int main()
{
char x;
int x1,x2,p,m,n,r;
cout<<"enter value x:"<<"\n";
cin>>x;
cout<<"enter value x1,x2:"<<"\n";
cin>>x1>>x2;
switch(x)
{
case '+': p=x1+x2 ; cout<<p; break;
case '-':m=x1-x2;cout<<m;;break;
case '*':n=x1*x2;cout<<n;break;
case '/':r=x1/x2;cout<<r;break;
default:cout<<"nothing";
}
return 0;
}
```

وضع نفس الأوامر لأكثر من حالة:

```
#include <iostream>
using namespace std;

int main()
{
    int x = 3;

    switch( x )           // اختبر قيمة المتغير x
    {
        case 1:          // في حال كانت تساوي 1 أو 2 أو 3 سيتم تنفيذ أمر الطباعة
        case 2:
        case 3:
            cout << "IRAQ";
            break;

        default:         // في حال كانت لا تساوي أي قيمة من القيم الموضوعة سيتم تنفيذ أمر الطباعة الموضوع فيها
            cout << "nothing";
    }

    return 0;
}
```

الحلقات التكرارية (Loops)

(For , Whlie , Do...while)

*نستخدم الحلقات (Loops) بهدف تكرار نفس الكود عدة مرات.

١) حلقة التكرار (for):

نستخدم الحلقة **for** إذا كنا نريد تنفيذ الكود عدة مرات محددة، فمثلاً إذا كنا نريد تنفيذ كود معين ٠ ١مرات، نضعه بداخل حلقة تعيد نفسها ٠ ١دورات.

طريقة استخدامها:

(قيمة الزيادة او النقصان للعداد؛ شرط التوقف للحلقة التكرارية ؛ القيمة الابتدائية للعدد =**i**)

مثال ١/ برنامج بلغة C++ يستخدم لطباعة عبارة Welcome خمس مرات باستخدام حلقة التكرار .(for)

```
#include<iostream>
using namespace std;
int main()
{
int i;
for(i=1;i<=5;i=i+1)
cout<<"Welcome"<<"\n";
return 0;
}
```

مثال ٢/ برنامج بلغة C++ يستخدم لطباعة العبارتين java , c++ خمس مرات باستخدام حلقة التكرار .(for)

```

#include<iostream>
using namespace std;
int main()
{
int i;
for(i=1;i<=5;i=i+1)
{
cout<<"c++"<<"\n";
cout<<"jave"<<"\n";
}
return 0;
}

```

ملاحظة:

-وضعنا العبارتين بين قوسين في داخل الحلقة كما بینا سابقا حتى تكون للحلقة سيطرة على العبارتين.
-نضع الاقواس في حالة انه الحلقات في داخلها اکثر من سطر برمجي واحد.

مثال ٣/ برنامج بلغة C++ يستخدم لطباعة الاعداد من ١ الى ١٠ بـاستخدام حلقة التكرار (for).

```

#include<iostream>
using namespace std;
int main()
{
int i;
for(i=1;i<=10;i=i+1)
cout<<i<<"\n";
return 0;
}

```

مثال ٤/ برنامج بلغة C++ يستخدم لطباعة الاعداد من ١ الى ١٠ بالعكس بـاستخدام حلقة التكرار (for).

```

#include<iostream>
using namespace std;
int main()
{
for(int i=10;i>=1;i=i-1)
cout<<i<<"\n";
return 0;
}

```

مثال ٥ / برنامج بلغة C++ يستخدم لطباعة الاعداد الفردية المقصورة بين ١ و ١٠ باستخدام حلقة التكرار .(for)

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1;i<=10;i=i+2)
cout<<i<<"\n";
return 0;
}
```

مثال ٦ / برنامج بلغة C++ يستخدم لطباعة الاعداد الزوجية المقصورة بين ١ و ١٠ باستخدام حلقة التكرار .(for)

```
#include<iostream>
using namespace std;
int main()
{
for(int i=٢;i<=10;i=i+2)
cout<<i<<"\n";
return 0;
}
```

مثال ٧ / برنامج بلغة C++ يستخدم لحساب مجموع الاعداد من ١ الى ١٠ باستخدام حلقة التكرار (for).

```
#include<iostream>
using namespace std;
int main()
{
int sum=0;
for(int i=1;i<=10;i=i+1)
sum=sum+i;
cout<<"sum=""<<sum;
return 0;
}
```

مثال ٨ / برنامج بلغة C++ يقوم المستخدم بادخال خمس اعداد صحيحة ثم يجد مجموعها باستخدام حلقة التكرار (for).

```

#include<iostream>
using namespace std;
int main()
{
int x,sum=0;
for(int i=1;i<=5;i=i+1)
{
cout<<"enter number:"<<"\n";
cin>>x;
sum=sum+x;
}
cout<<"sum="<<sum;
return 0;
}

```

مثال ٩/ برنامج بلغة C++ يقوم المستخدم بادخال اربع اعداد صحيحة ثم يجد مجموع الاعداد الزوجية فقط باستخدام حلقة التكرار (for).

```

#include<iostream>
using namespace std;
int main()
{
int x,sum=0;
for(int i=1;i<=4;i=i+1)
{
cout<<"enter number:"<<"\n";
cin>>x;
if(x%2==0)
sum=sum+x;
}
cout<<"sum="<<sum;
return 0;
}

```

واجب/ برنامج بلغة C++ يقوم المستخدم بادخال اربع اعداد صحيحة ثم يجد مجموع الاعداد الفردية فقط باستخدام حلقة التكرار (for).

مثال ١٠ / برنامج بلغة C++ يستخدم لحساب مضروب العدد الصحيح n المدخل من قبل المستخدم بأستخدام حلقة التكرار (for).

ملاحظة:

مثلاً مضروب العدد ٤ هو ناتج ضرب جميع الأعداد الصحيحة الموجبة التي تسبق العدد ٤ بدءاً من ١. وبالتالي:
$$4! = 4 \times 3 \times 2 \times 1 = 24$$

إذن، مضروب العدد ٤ هو ٢٤.

```
#include<iostream>
using namespace std;
int main()
{
    int n,f=1;
    cout<<"enter value for n:"<<"\n";
    cin>>n;
    for(int i=n;i>=1;i=i-1)
        f=f*i;
    cout<<"factorial="<<f;
    return 0;
}
```

/ ١١ مثال

اكتب برنامج يطبع السلسلة التالية:

10 20 30 40 50 60 70 80 90 100

```
#include<iostream>
using namespace std;
int main()
{
    for (int i=10;i<=100;i=i+10)
        cout<<i<<" ";
    return 0;
}
```

واجب/

اكتب برنامج يطبع السلسلة التالية:

1 5 9 13 17 21

٢) حلقة التكرار (while)

نستخدم الحلقة **while** إذا كنا نريد تنفيذ الكود عدة مرات، ولكن عندما يكون الشرط متحقق.

طريقة استخدامها:

```
initialization;  
  
while( condition )  
{  
    // statements  
  
    decrement; أو increment  
}
```

مثال ١٢ / برنامج بلغة C++ يستخدم لطباعة عبارة Welcome خمس مرات باستخدام حلقة التكرار .(while)

```
#include<iostream>  
using namespace std;  
int main()  
{  
int i=1;  
while(i<=5)  
{  
cout<<"welcome"<<"\n";  
i=i+1;  
}  
return 0;  
}
```

مثال ١٣ / برنامج بلغة C++ يستخدم لطباعة الاعداد من ١ الى ١٠ باستخدام حلقة التكرار (while).

```
#include<iostream>  
using namespace std;  
int main()  
{  
int i=1;  
while(i<=10)  
{cout<<i<<"\n";  
i=i+1;}  
return 0;  
}
```

واجب / برنامج بلغة C++ يستخدم لطباعة الاعداد من ١ الى ١٠ بالعكس باستخدام حلقة التكرار .(while)

مثال ١٤ / برنامج بلغة C++ يستخدم لطباعة الاعداد الفردية الممحصورة بين ١ و ١٠ باستخدام حلقة التكرار (while).

```
#include<iostream>
using namespace std;
int main()
{
int i=1;
while(i<=10)
{
cout<<i<<"\n";
i=i+2;
}
return 0;
}
```

ملاحظة // ممكن تطبيقه ايضاً باستخدام if.

واجب / برنامج بلغة C++ يستخدم لطباعة الاعداد الزوجية الممحصورة بين ١ و ١٠ باستخدام حلقة التكرار .(while)

مثال ١٥ / برنامج بلغة C++ يستخدم لحساب مجموع الاعداد من ١ الى ١٠ باستخدام حلقة التكرار .(while)

```
#include<iostream>
using namespace std;
int main()
{
int i=1,sum=0;
while(i<=10)
{
sum=sum+i;
i=i+1;
}
cout<<"sum="<<sum;
return 0;
}
```

مثال ١٦ / برنامج بلغة C++ يقوم المستخدم بادخال اربع اعداد صحيحة ثم يجد مجموع الاعداد الزوجية فقط باستخدام حلقة التكرار (While).

```
#include<iostream>
using namespace std;
int main()
{
int x,i,sum=0;
i=1;
while(i<=4)
{
cout<<"enter number:"<<"\n";
cin>>x;
if(x%2==0)
sum=sum+x;
i=i+1;
}
cout<<"sum="<<sum;
return 0;
}
```

مثال ١٧ / برنامج بلغة C++ يستخدم لحساب مضروب العدد الصحيح n المدخل من قبل المستخدم باستخدام حلقة التكرار (While).

```
#include<iostream>
using namespace std;
int main()
{
int n,i,f=1;
cout<<"enter value for n:"<<"\n";
cin>>n;
i=n;
while(i>=1)
{
f=f*i;
i=i-1;
}
cout<<"factorial="<<f;
return 0;
}
```

واجبات: مستخدما الحلقة التكرارية :While

اكتب برنامج يطبع السلسلة التالية:

10 20 30 40 50 60 70 80 90 100

اكتب برنامج يطبع السلسلة التالية:

1 5 9 13 17 21

: حلقة التكرار (do...while)

Here the loop continues until the condition is met.

do...while : This loop continues until the condition is met. It is similar to the while loop, but it checks the condition at the end of the loop body.

initialization;

```
do{  
    // statements  
    decrement; أو increment  
}  
While( condition );
```

Example 18 / A C++ program that prints the numbers from 1 to 10 using a do...while loop.

```
#include<iostream>  
using namespace std;  
int main()  
{  
int i;  
i=1;  
do  
{  
cout<<i<<"\n";  
i++;  
}  
while(i<=10);  
return 0;  
}
```

Another example:

```

#include<iostream>
using namespace std;
int main()
{
int i;
i=1;
do
{
cout<<i<<"\n";
i++;
}
while(i>=20);
return 0;
}

```

هنا سوف يتم تنفيذ جملة طباعة العدد ١ بعد **do**, وبعدها تم زيادة العداد بمقدار واحد فأصبح **i=2** لكن هنا تم ادخالها مع الشرط فثبتت ان ٢ لم تتحقق الشرط **i=20>** وبالتالي تم التوقف عن طباعة الاعداد اي تم طباعة العدد ١ فقط.

مثال ١٩ / برنامج بلغة C++ يستخدم لطباعة عبارة Welcome خمس مرات باستخدام حلقة التكرار .(do...while)

```

#include<iostream>
using namespace std;
int main()
{
int i;
i=1;
do
{
cout<<"Welcom"<<"\n";
i++;
}
while(i<=5);
return 0 ;
}

```

***المصفوفة (Array)** عبارة عن هيكل بياني يتكون من مجموعة من العناصر والتي جميعها من نفس النوع البياني. و كل عنصر في المصفوفة يمكن تخزين قيمة واحدة فيه.

* عناصر المصفوفة تتميز عن بعضها من خلال رقم محدد يعطى لكل عنصر يسمى **index**.
أول عنصر في المصفوفة دائماً يكون رقمه **0**.

فوائد المصفوفات:

- تقليل عدد المتغيرات المتشابهة، فمثلاً إذا كنا نريد تعريف 10 متغيرات نوعهم **int** نقوم بتعريف مصفوفة واحدة تتالف من 10 عناصر.
- التعامل مع الكود يصبح أسهل، لأنك إذا قمت بخزن المعلومات داخل مصفوفة، تستطيع تعديلهن، مقارنتهم أو جلبهم كلهم دفعة واحدة بکود صغير جداً باستخدام حلقه.
- تستطيع الوصول لأي عنصر من خلال رقم الـ **index** الخاص به.

(1) المصفوفة احادية البعد:

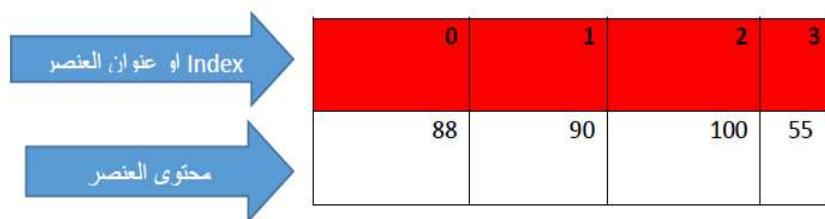
في هذا النوع تكون العناصر على شكل صف وكل عنصر في المصفوفة تسلسلي.

التصريح عن المصفوفة احادية البعد:

; [حجم المصفوفة او سعتها] اسم المصفوفة النوع البياني;

مثلاً التصريح عن مصفوفة احادية البعد اسمها **a** و سعتها 4 عناصر صحيحة:

و تمثل بالشكل التالي:



a[0]=88 حيث العنصر الاول
a[1]=90 حيث العنصر الثاني
a[2]=100 حيث العنصر الثالث
a[3]=55 حيث العنصر الرابع

ملاحظة // كل العمليات (الادخال و المعالجة وطباعة) في المصفوفات تحتاج إلى دوارة for وذلك لانه يحتاج إلى تسلسل العناصر في المصفوفة لكي يتم كل هذه العمليات على العناصر .

مثال 1/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد n سعتها خمس عناصر من النوع الصحيح ثم طباعة هذه العناصر.

```

#include<iostream>
using namespace std;
int main()
{
int a[4];
cout<<"Enter Elements:"<<"\n";
for(int i=0;i<=3; i=i+1)
cin>>a[i];

for(int i=0;i<=3; i=i+1)
cout<<a[i]<<" ";
return 0;
}

```

ممكن الادخال يكون بشكل مباشر بطريقتين سوف نطبقهما على المثال السابق:

```

#include<iostream>
using namespace std;
int main()
{
int a[4]={33,66,77,88};
for(int i=0;i<=3; i=i+1)
cout<<a[i]<<" ";
return 0;
}

```

او ممكن بادخال قيمة لكل عنصر على حدة:

```

#include<iostream>
using namespace std;
int main()
{
int a[4];
a[0]=90;
a[1]=100;
a[2]=70;
a[3]=200;
for(int i=0;i<=3; i=i+1)
cout<<a[i]<<" ";

```

```
return 0;
```

```
}
```

مثال 2/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها خمس عناصر من النوع الصحيح ثم يقوم بأيجاد اكبر قيمة موجودة في المصفوفة وطباعتها.

```
#include<iostream>
using namespace std;
int main()
{
int a[5],max;
cout<<"Enter Elements:"<<"\n";
for(int i=0;i<=4;i=i+1)
cin>>a[i];
max=a[0];
for(int i=1;i<=4; i=i+1)
if(a[i]>max)
max=a[i];
cout<<"Max="<<max;
return 0;
}
```

واجب 1/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها خمس عناصر من النوع الصحيح ثم يقوم بأيجاد اصغر قيمة موجودة في المصفوفة وطباعتها.

مثال 3/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها خمس عناصر من النوع الصحيح ثم يقوم بطباعة القيم الزوجية فقط.

```
#include<iostream>
using namespace std;
int main()
{
int a[5];
cout<<"Enter Elements:"<<"\n";
for(int i=0;i<=4;i=i+1)
cin>>a[i];
for(int i=0;i<=4; i=i+1)
```

```

if(a[i]%2==0)
cout<<a[i]<<" ";
return 0;
}

```

واجب 2/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها 10 عناصر من النوع الصحيح ثم يقوم بطباعة القيم الفردية فقط.

مثال 4/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها خمس عناصر من النوع الصحيح ثم يقوم بطباعة القيم الموجودة في الموضع الزوجية للمصفوفة فقط.

ملاحظة// الموقع يقصد به تسلسل العناصر في المصفوفة.

```

#include<iostream>
using namespace std;
int main()
{
int a[5];
cout<<"Enter Elements:"<<"\n";
for(int i=0;i<=4;i=i+1)
cin>>a[i];
for(int i=0;i<=4; i=i+1)
if(i%2==0)
cout<<a[i]<<" ";
return 0;
}

```

واجب 3/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها خمس عناصر من النوع الصحيح ثم يقوم بطباعة القيم الموجودة في الموضع الفردية للمصفوفة فقط.

مثال 5/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها خمس عناصر من النوع الصحيح ثم يقوم بترتيب عناصر المصفوفة تصاعديا(من الصغر الى الاكبر).

```
#include<iostream>
using namespace std;
int main()
{
int a[5],t;
cout<<"Enter Elements:"<<"\n";
for(int i=0;i<=4;i++)
cin>>a[i];
```

قراءة عناصر المصفوفة

```
cout<<"Elements before change:"<<"\n";
for(int i=0;i<=4;i++)
cout<<a[i]<<" ";
cout<<"\n";
```

طباعة عناصر المصفوفة

قبل الترتيب

```
for(int k=0;k<=3;k++)
{
for(int j=k+1;j<=4;j++)
if(a[j]<a[k])
{
t=a[k];
a[k]=a[j];
a[j]=t;
}
}
```

عملية الترتيب لعناصر المصفوفة

```
cout<<"Elements after change:"<<"\n";
for(int i=0;i<=4;i++)
cout<<a[i]<<" ";
return 0;
}
```

طباعة عناصر المصفوفة

بعد الترتيب

واجب 4 برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة احادية البعد سعتها خمس عناصر من النوع الصحيح ثم يقوم بترتيب عناصر المصفوفة تنازليا(من الافضل الى الاقل).

(2) المصفوفة ثنائية البعد:

تتميز بكون عناصرها تتوزع الى مجموعة من الصفوف والاعمدة التي تمثل حجم المصفوفة.

اسم المصفوفة النوع البياني [n][m];

int a[3][3];

مثال التصريح عن مصفوفة ثنائية البعد اسمها a وسعتها 3*3 عناصر صحيحة:

مثال 6 / برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة ثنائية البعد n سعتها 3*3 عناصر من النوع الصحيح ثم طباعة هذه العناصر.

```
#include<iostream>
using namespace std;
int main()
{
int a[3][3];
int i,j;
for(i=0;i<=2;i++)
for(j=0;j<=2;j++)
{
cout<<"Enter Elements:"<<"\n";
cin>>a[i][j];
}
for(i=0;i<=2;i++)
for(j=0;j<=2;j++)
cout<<a[i][j]<<" ";
}
```

مثال 7 / برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة ثنائية البعد سعتها 4 عناصر من النوع الصحيح ثم يقوم بأيجاد اكبر قيمة موجودة في المصفوفة وطباعتها.

```
#include<iostream>
```

```

using namespace std;
int main()
{
int a[2][2];
int i,j,max;
for(i=0;i<=1;i++)
for(j=0;j<=1;j++)
{
cout<<"Enter Elements:"<<"\n";
cin>>a[i][j];
}
max=a[0][0];
for(i=0;i<=1;i++)
for(j=0;j<=1;j++)
if(a[i][j]>max)
max=a[i][j];
cout<<"Max="<<max;
}

```

مثال 8/ برنامج بلغة C++ يستخدم لقراءة عناصر مصفوفة ثنائية البعد سعتها 9 عناصر من النوع الصحيح ثم يقوم بطباعة عناصر القطر الرئيسي للمصفوفة.

```

#include<iostream>
using namespace std;
int main()
{
int a[3][3];
int i,j;
for(i=0;i<=2;i++)

```

```
for(j=0;j<=2;j++)
{
cout<<"Enter Elements:"<<"\n";
cin>>a[i][j];
}
for(i=0;i<=2;i++)
for(j=0;j<=2;j++)
if(i==j)
cout<<a[i][j]<<" ";
}
```

الدالة (Function) عبارة عن مجموعة أوامر مجمعة في مكان واحد و تتنفذ عندما نقوم باستدعائها.

```
returnType functionName(Parameter)
{
    // Function Body
}

int main()
{
}
```

: يحدد النوع الذي سترجعه الدالة عندما تنتهي أو إذا كانت لن ترجع أي قيمة ويكون : **returnType**

: اذا كانت الدالة تقوم بارجاع قيمة صحيحة او كسرية للبرنامج الرئيسي ونحتاج الى **return** دخل الدالة الفرعية.

: اذا كانت الدالة لا تقوم بارجاع اي قيمة للبرنامج الرئيسي.

بالاضافة للأنواع البياناتية الاخرى.....

: يمثل الاسم الذي نعطيه للدالة، و الذي من خلاله يمكننا استدعاءها. **functionName**

: المقصود المتغيرات التي تحتاجها الدالة. **Parameter**

: تعني جسم الدالة، و المقصود بها الأوامر التي نضعها في الدالة. **Function Body**

ملاحظة: يمكن وضع الدالة الفرعية قبل او بعد البرنامج الرئيسي

مثال 1/ برنامج بلغة C++ يستخدم لطباعة عبارة welcome to c++ مستخدما الدوال الفرعية من نوع .void

```
#include<iostream>
using namespace std;
void print()
{
    cout<<"welcome to c++";
}
int main()
{
    print();
}
```

مثال 2 / برنامج بلغة C++ يستخدم لجمع عددين مستخدما الدوال الفرعية من نوع .int

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z;
cin>>x>>y;
z=x+y;
cout<<z;
}
```

برنامـج اعتـيادي بـدون استـخدام دـالة

```
#include<iostream>
using namespace std;
```

```

int add(int x,int y)
{
int z;
z=x+y;
return(z);
}
int main()
{
int x,y,z;
cin>>x>>y;
z=add(x,y);
cout<<z;
}

```

ملاحظة: النوع البياني للدالة int لأن عملية الطباعة داخل البرنامج الرئيسي وليس داخل الدالة نفسها.

. مثال 3/ برنامج بلغة C++ يستخدم لطرح عددين مستخدما الدوال الفرعية من نوع void .

```

#include<iostream>
using namespace std;
void sub(int x,int y)
{
int z;
z=x-y;
cout<<"z="<<z;
}
int main()
{
int x,y,z;
cin>>x>>y;
sub(x,y);
}

```

ملخص:

-اذا كانت الدالة من نوع (void) اي ان عملية المعالجة ان وجدت والطباعة تكون داخل الدالة الفرعية.
-اذا كانت الدالة من نوع int, float, string,.... اي ان عملية المعالجة داخل الدالة الفرعية والطباعة تكون داخل البرنامج الرئيسي.

مثال 4/ برنامج بلغة C++ يستخدم لايجاد مكعب العدد المدخل مستخدما الدوال الفرعية من نوع int.

```
#include<iostream>
using namespace std;
int sqr(int x)
{
int z;
z=x*x*x;
return(z);
}
int main()
{
int x,z;
cin>>x;
z=sqr(x);
cout<<z;
}
```

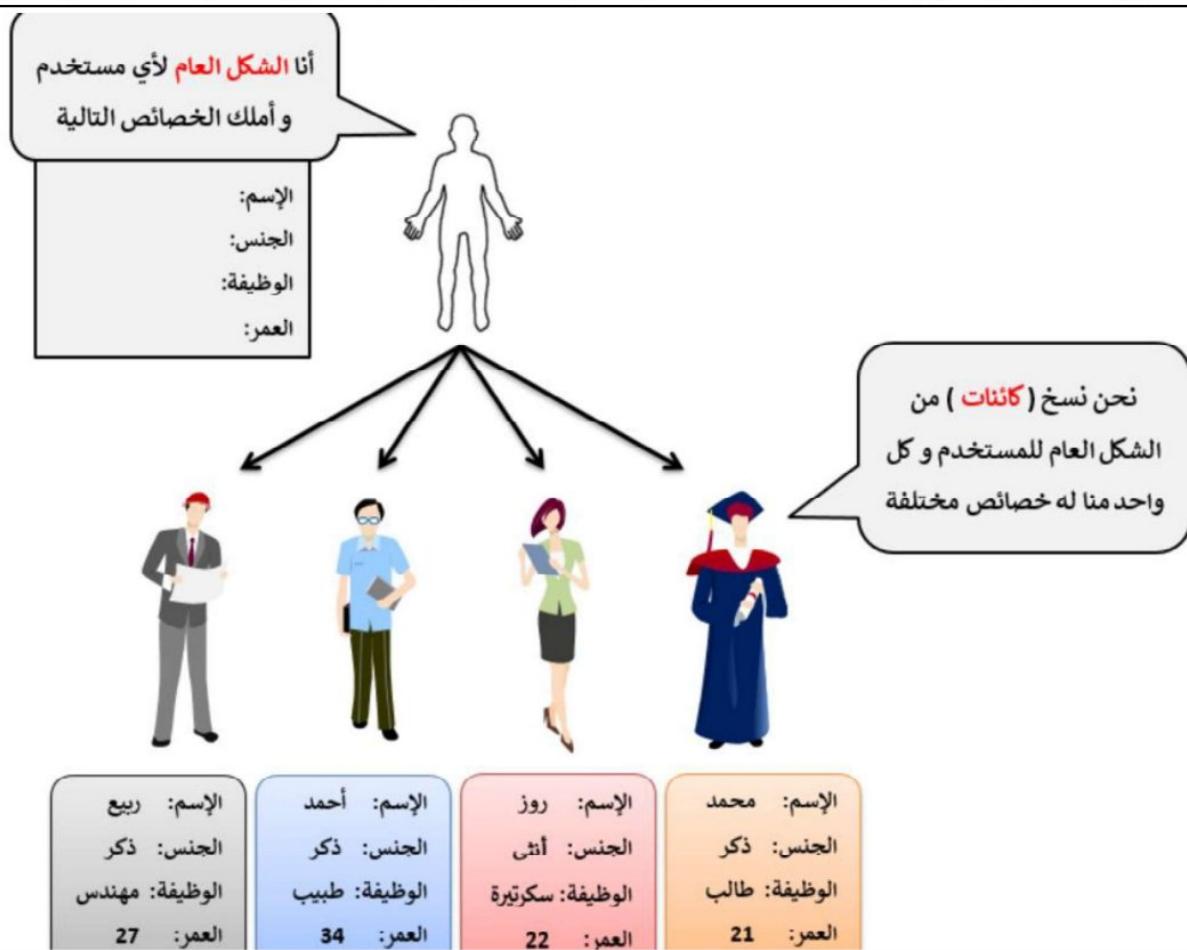
واجب1: برنامج بلغة C++ يستخدم لايجاد معدل ثلاث مواد مستخدما الدوال الفرعية من نوع void .

واجب2: برنامج بلغة C++ يستخدم لايجاد معدل خمس مواد مستخدما الدوال الفرعية من نوع float .

مفهوم البرمجة الكائنية (OOP)

البرمجة الكائنية (Object Oriented Programming) تختصر بكلمة OOP وهي عبارة عن أسلوب تتبعه في كتابة الكود لجعل كتابة الكود أكثر سهولة.

إذاً البرمجة الكائنية هي مجرد أسلوب في العمل لا أكثر و هي ليست خاصة بلغة C++ حيث أنها تطبق في باقي لغات البرمجة. فكرة البرمجة الكائنية بشكل عام هي تجهيز الشكل الذي سيتم فيه حفظ المعلومات مما يجعل الوصول إليها و التعديل عليها سهل للغاية. كمثال بسيط، إذا كنت تنوی بناء برنامج لحفظ معلومات المستخدمين، ستقوم بتجهيز **الشكل العام** للمعلومات التي تنوی حفظها لكل مستخدم. بعدها، أي مستخدم جديد تنوی إنشاؤه تجعله نسخة من **الشكل العام** لأي مستخدم و تجعله يدخل القيم الخاصة به كما في الصورة التالية.



الكلاس (Class): عبارة عن حاوية كبيرة تستطيع أن تحتوي على كل الكود من متغيرات و دوال و كائنات إلخ....

الكائن (Object): عبارة عن نسخة مطابقة لкласс معين.

هيكل الكلاس:

Class class_name

{

Private:

متغيرات

دوال فرعية

Public:

متغيرات

دوال فرعية

};

Int main()

{

class_name object_name;

}

ملاحظة 1:

في حالة ان المتغير كان معرفا داخل جزء **public** اذن يتم ادخال قيمة لهذا المتغير فقط من داخل البرنامج الرئيسي بطريقتين:

1- الادخال غير المباشر مثل **cin>> class_name ; اسم المتغير.**

2- الادخال المباشر مثل **class_name = القيمة ; اسم المتغير.**

مثال 1 / برنامج بلغة C++ يستخدم لطباعة عنوان الكتاب واسم الكاتب و عدد صفحات الكتاب
مستخدما تعريف كلاس اسمه books :

Public:

```
string title  
string author  
int number_pages
```

علما ان الطباعة تشمل كائن واحد فقط (كتاب واحد).

```
#include<iostream>  
using namespace std;  
class books  
{  
private:  
public:  
string title;  
string author;  
int number_pages;  
};  
int main()  
{  
books book1;  
cout<<"info for book1:"<<"\n";  
cin>>book1.title>>book1.author>>book1.number_pages;  
cout<<book1.title<<" "<<book1.author<<" "<<book1.number_pages<<"\n";  
}
```

مثال 2/ نطبق المثال اعلاة ولكن لـ كاثنين (كتابين):

```
#include<iostream>
using namespace std;
class books
{
private:
public:
string title;
string author;
int number_pages;
};
int main()
{
books book1;
books book2;

cout<<"info for book1:"<<"\n";
cin>>book1.title>>book1.author>>book1.number_pages;

cout<<"info for book2:"<<"\n";
cin>>book2.title>>book2.author>>book2.number_pages;

cout<<book1.title<<" "<<book1.author<<" "<<book1.number_pages<<"\n";

cout<<book2.title<<" "<<book2.author<<" "<<book2.number_pages<<"\n";
}
```

```
info for book1:
opp ali 356
info for book2:
programming kareem 600
opp ali 356
programming kareem 600
```

مثال 3/ برنامج بلغة C++ يستخدم لايجاد مساحة مستطيل مستخدما تعريف كلاس اسمه :rectangular

Public:

طول المستطيل=l

عرض المستطيل=w

لارجاع قيمة الخاصية بمساحة المستطيل للبرنامج الرئيسي.

علماء ان ايجاد مساحة المستطيل تكون لكائن واحد فقط.

```
#include<iostream>
using namespace std;
class rectangular
{
private:
public:
int l,w,ar;
int area()
{
ar=l*w;
return(ar);
}
int main()
{
rectangular r;
int z;
cin>>r.l>>r.w; او r.l=5; r.w=5;
z= r.area();
cout<<"Area="<<z;
}
```

تدريب : نطبق المثال اعلاه ولكن الدالة area تكون من نوع void

الدوال الاعضاء في الكلاس

1- يتم تعريف الدالة **فقط** داخل الكلاس بدون ذكر عمل هذه الدالة ضمن الجزء المخصص لها.

2-تعريف عمل الدالة خارج الكلاس من خلال ذكر تعريف النوع البياني ثم اسم الكلاس ثم اسم الدالة.

```
نوع البيانات   اسم الدالة :: اسم الكلاس
{
```

عمل الدالة

```
}
```

مثال 4/ برنامج بلغة C++ يستخدم لايجاد مساحة مستطيل مستخدماً تعريف كلاس اسمه :rectangular

Public:

طول المستطيل= l

عرض المستطيل= w

لارجاع قيمة الخاصة بمساحة المستطيل للبرنامج الرئيسي.

علماء ان ايجاد مساحة المستطيل تكون لكائن واحد فقط. استخدم الدوال كدوال اعضاء خارج الكلاس.

```
#include<iostream>
using namespace std;
class rectangular
{
private:
public:
int l,w,ar;
int area();
};
```

1

```
int rectangular::area()
{
ar=l*w;
return(ar);
}
int main()
{
rectangular r;
int z;
cout<<"Enter L,W:"<<"\n";
cin>>r.l>>r.w;
z=r.area();
cout<<"Area="<<z;
}
```

ملاحظة 2:

في حالة ان المتغير كان معرفا داخل جزء الـ **private** اذن يتم ادخال قيمة لهذا المتغير فقط من داخل الكلاس من خلال دالة خاصة والادخال يتم بطريقتين:

1- الادخال غير المباشر مثل `cin>> اسم المتغير`

2- الادخال المباشر مثل `قيمة= اسم المتغير`

مثال 1/ مستخدما تعريف الكلاس(**info**) ببرنامج بلغة C++ يستخدم لطباعة المعلومات (رقم الطالب,اسم الطالب,معدل الطالب) علما ان

Private

`st_id`
`st_name`
`st_avg`

public

`:void input()`: تستخدم لادخال قيم للمعلومات اعلاه.

`:Void print()`: تستخدم لطباعة المعلومات اعلاه.

علما ان الطباعة تكون لطالب واحد فقط.

```
#include<iostream>
using namespace std;
class info
{
private:
int st_id;
string st_name;
int st_avg;
public:
void input()
{
    cin>>st_id>>st_name>>st_avg;
}
void print()
{
cout<<st_id<<" "<<st_name<<""
"<<st_avg;
}
};
```

```
int main()
{
info st;
st.input();
st.print();
}
```

مثال 2 / استخدام نفس المثال اعلاه ولكن استخدام الدوال دوال اعضاء للكلاس.

```
#include<iostream>
using namespace std;
class info
{
private:
int st_id;
string st_name;
int st_avg;
public:
void input();
void print();
};
void info::input()
{
cin>>st_id>>st_name>>st_avg;
}
void info::print()
{
cout<<st_id<<" "<<st_name<<" "<<st_avg;
}
```

```
int main()
{
info st;
st.input();
st.print();
}
```

تدريب: برنامج بلغة C++ يستخدم لإيجاد مساحة مستطيل مستخدماً تعريف كلاس اسمه `:rectangular`

private:

طول المستطيل = `l`

عرض المستطيل = `w`

Public:

تستخدم لادخال قيم المتغيرات. `Void read()`

لارجاع قيمة الخاصة بمساحة المستطيل للبرنامج الرئيسي. `() Int area()`

علماً ان ايجاد مساحة المستطيل تكون لكائن واحد فقط.