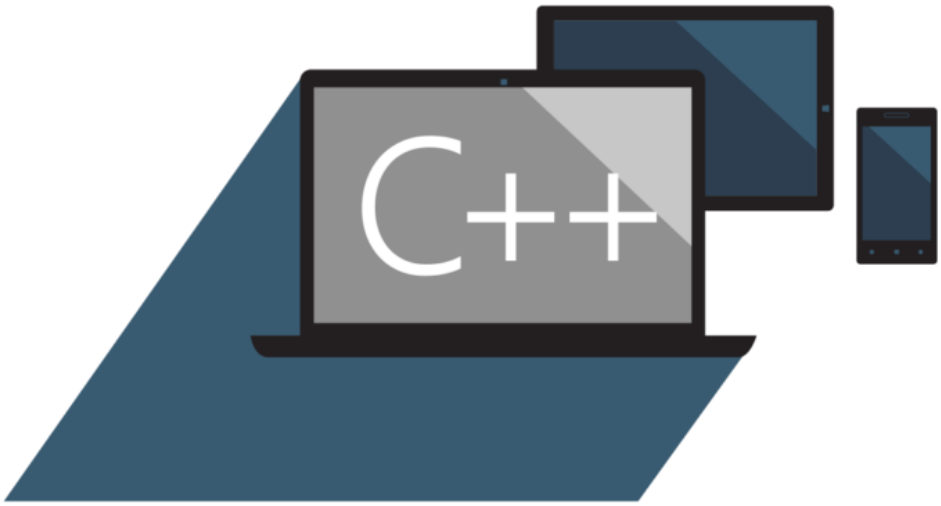
C++

The First lecture



الكلية التربية

قسم الرياضيات

اساسيات اللغة

١-١ مقدمة

يعتبر الحاسوب الالكتروني اداة مفيدة لحل العديد من المشاكل، وأي حل لمشكلة ما يسمى **خوارزمية Algorithm** والتي تصف سلسلة من الخطوات تؤدي الى حل المشكلة. وتحتوي الخوارزمية على مجموعة من التعابير المجردة والتي تحتاج الى التعبير عنها بلغة مفهومة الى الحاسوب، واللغة الوحيدة المفهومة للحاسوب هي **لغة الماكنة Machine Language**، اذ ان اي برنامج يكتب باي لغة اخرى يتم تحويله الى لغة الماكنة ليتم تنفيذه.

وتعتبر لغة الماكنة صعبة للاستخدام المباشر من قبل المبرمجين لذا يتم استخدام مجموعة من الترميزات التي يعبر عنها بـ **لغة التجميع Assembly Language** والتي تزود مجموعة من الاوامر والترميزات المفهومة للبيانات، اذ ان البرنامج المكتوب بواسطة لغة التجميع يتم ترجمته الى **لغة الماكنة** بواسطة مترجم يدعى **المجمع Assembler**.

تعد لغة التجميع صعبة للعمل مع اللغات عالية المستوى مثل لغة C++ اذ تزودنا هذه اللغة بترميزات اكثر ملائمة لتطبيق الخوارزميات.

اي برنامج كُتب بواسطة لغات عالية المستوى يتم تحويله الى لغة التجميع بواسطة مترجم يدعى بـ**المترجم Compiler** اذ تجمع هذه الرموز لإنتاج برنامج قابل للتنفيذ.

١-٢ لغة C++:

هي لغة برمجة ذات أغراض عامة ومرونة اعلى لغة C السابقة، وتخزن كل مجموعة من الايعازات في مكتبة خاصة يتم استدعاءها عند الحاجة الى استخدام اي من ايعازات هذه المكتبة، وتدعم لغة C++ **البرمجة الاجرائية**[١] و**البرمجة الكيانية الموجهة Object Oriented Programming**[٢]، كذلك تدعم البرمجة التي تعتمد على البيانات المجردة، مع العلم ان جميع البرامج التي سيتم سردها في هذا الكتاب تم تنفيذها مسبقاً من خلال محرر برامج اللغة Borland Turbo C++ الاصدار 4.5.

---

[١] **البرمجة الاجرائية Procedural Programming**: اذ يركز اسلوب البرمجة هذا على المعالجة في ترتيب خطوات البرمجة، اذ تقرر اي اجراء تريد تنفيذه وما الخوارزمية الانسب للحل، اذ يدعم اسلوب البرمجة المذكور تمرير المعاملات الى الدوال كافة المرتبطة بالبرنامج الرئيسي ويرجع القيم من الدوال المستدعاه.

[٢] **البرمجة الكيانية الموجهة Object Oriented Programming**: هي البرمجة بشكل جيد لحل مجموعة من المشاكل يتم التركيز في هذا الاسلوب من البرمجة على البيانات، حيث تزود الآليات المناسبة التي تتعامل مع الكيانات والاجسام Object.

تطورت لغة C ++ من C، والتي تطورت من اثنين من لغات البرمجة السابقة، BCPL و B. BCPL[٣] وتم تطويره في عام ١٩٦٧ من قبل مارتن ريتشاردز كلغة لكتابة برمجيات أنظمة التشغيل وكمترجمات لأنظمة التشغيل. واستخدمت لتكوين الإصدارات القديمة من نظام التشغيل UNIX في مختبرات بيل في عام ١٩٧٠.

وقد تطورت لغة C من B عن طريق دينيس ريتشي في مختبرات بيل. تستخدم C العديد من المفاهيم الهامة من BCPL و.B في البداية أصبح معروفاً على نطاق واسع كلغة تطوير لنظام التشغيل UNIX. اليوم، يتم كتابة معظم أنظمة التشغيل في C و/ أو C++ هي الآن متاحة لمعظم أجهزة الحاسوب وأجهزة مستقلة أخرى.

الاستخدام الواسع النطاق للغة C مع أنواع مختلفة من أجهزة الحاسوب (التي تسمى أحيانا **منصات Platforms الأجهزة)** أدى إلى العديد من الاختلافات، وكانت هذه مشكلة كبيرة لمطوري البرنامج، الذين هم في حاجة لكتابة البرامج المحمولة التي من شأنها أن تعمل على عدة منصات. وثمة حاجة إلى الإصدار القياسي من C. تعاون المعهد الوطني الأمريكي للمعايير (ANSI) مع المنظمة الدولية للتوحيد القياسي (ISO) لتوحيد C في جميع أنحاء العالم، فاليوم يمكن تشغيل التطبيقات المكتوبة في C في كثير من الأحيان مع تعديلات ضئيلة أو معدومة على مجموعة واسعة من أنظمة الحاسوب.

وقد وضعت C++ كأمتداد لـC، من خلال بيارن ستروستروب في ١٩٨٠ في وقت مبكر في مختبرات بيل. يوفر C++ عددا من الميزات التي "تجمل" لغة C، ولكن الأهم من ذلك، فإنه يوفر قدرات **للبرمجة الكيانية الموجهةObject-Oriented Programming**.

ويأتي هذا في الوقت الذي كان فيه الطلب على برامج جديدة وأكثر قوة في ارتفاع. **الكائنات Objects** هي أساسا مكونات البرامج التي يعاد استخدامها أن البنود النموذجية في العالم الحقيقي. مطوري البرمجيات يكتشفون أن استخدام وتصميم وتنفيذ نهج ووحدات كيانية المنحى يمكن جعلها أكثر إنتاجية بكثير مما يمكن أن يكون مع تقنيات البرمجة السابقة التقليدية. برامج كيانية المنحى هي أسهل للفهم والتصحيح والتعديل.

مثال (١): برنامج في لغة C++ يستخدم لطباعة الاسم.

```
#include <iostream.h>
void main ()
{
cout << "ahmed & ali \n";
}
```

#include: تستخدم لتضمين اسم المكتبة الرئيسية iostream.h في لغة C++ والتي تحتوي على اوامر الادخال والاخراج والعمليات الحسابية الرئيسية.

() main.: وتعتبر اسم الدالة الرئيسية ويمكن ان تحمل قيمة او اكثر او لا تحمل قيم لارجاعها.

}: يؤشر هذا الرمز الى بداية جسم البرنامج الرئيسي.

{: يؤشر هذا الرمز الى نهاية جسم البرنامج الرئيسي.

>>cout: وهي عبارة الاخراج او الطباعة والتي ستظهر اي نص مطبوع بين علامتي الاقتباس " " على شاشة المخرجات، ورمز \n الموجود ضمن العبارة هو للاشارة الى ان الطباعة القادمة ستكون على سطر جديد.

١-٣ المتغيرات Variables:

المتغير هو اسم رمزي لاي نوع من البيانات يمكن ان تخزن في الذاكرة ويتم استدعاءها فيما بعد، المتغيرات تستخدم للاحتفاظ بقيم البيانات اذ يمكن ان تستخدم فيما بعد في حسابات مختلفة، كل المتغيرات لها خاصيتان مهمتان:

a- النوع type وهي القيم الافتراضية للمتغير كأن تكون قيمة (صحيحة، عشرية، رمز،...، الخ)، متى ما حدد النوع لا يمكن تغيير القيمة المعطاة اثناء كتابة البرنامج.

b- القيمة value والتي يمكن ان تغير بتخصيص قيمة جديدة للمتغير، اذ نوع القيم المعطاة للمتغير تعتمد على نوع المتغير المعرف، فمثلا متغير العدد الصحيح يمكن فقط ان يأخذ قيم صحيحة (مثلا ٢، ١٠٠،-١٢، ... الخ).

يجب ان يتضمن **اسم المتغير Variable Name** او كما يدعى **المعرف identifier** **مجموعة من الشروط الواجب** توفرها لكي نطلق عليها (المعرف) وهي:

١- ان لا يكون من ضمن الاسماء الدليلة او المحفوظة او المفتاحية Keyword (اي لا نسمي متغير باسم if او while).

٢- ان يبدء بحرف واحد على الاقل، مثلا x123.

٣- ان لا يحوي على فراغات داخل الاسم، واذا كان هناك حاجة لان يحتوي اسم المتغير اكثر من كلمة يتم استخدام علامة (_) للربط بين الكلمات، مثلا first_name.

ملاحظة: ان لغة C++ حساسة لتمييز الحروف الانكليزية الكبيرة عن الحروف الصغيرة، ولذا يفضل استخدام الحروف الانكليزية الصغيرة لكتابة البرامج، وعادة لغة C++ لا تفرض حدود على عدد الرموز الخاصة بكل اسم متغير ولكن يفضل ان لا تتجاوز ٢٥٥ رمز كي لا تسبب ارتباك اثناء كتابة البرامج.

١-٤ الكلمات المفتاحية Keyword C++:

هي الكلمات الدليلة التي لا يمكن استخدامها كمعرفات ضمن لغة C++، وهنا نورد معظمها للتوضيح في الجدول (١-١).

الجدول (١-١)

asm	do	if	return	try
auto	double	inline	short	typedef
bool	dynamic_cast	int	signed	typeid
break	else	long	sizeof	typename
case	enum	mutable	static	union
catch	explicit	namespace	static_cast	unsigned
char	export	new	struct	using
class	extern	operator	switch	virtual
const	false	private	template	void
const_cast	float	protected	this	volatile
continue	for	public	throw	wchar_t
default	friend	register	true	while
delete	goto	reinterpret_cast		

ان لغة C++ تعرف مجموعة من الانواع الحسابية والرموز والسلاسل الحرفية والرموز المنطقية وأن لكل نوع من هذه الانواع المساحة الخزنية التي يحتلها في الذاكرة وتختلف حسب الحاسبة المستخدمة والتي نعني بها كم بت bit يحتاج لتمثيل هذا النوع، ونورد في الجدول (١-٢) مجموعة من انواع المتغيرات ومعانيها والحجم الذي يحتاجه كل نوع من الذاكرة.

الجدول (١-٢)

Type	Meaning	Minimum Size
`char`	character	8 bits
`short`	short integer	16 bits
`int`	integer	16 bits
`long`	long integer	32 bits
`float`	single-precision floating-point	6 significant digits
`double`	double-precision floating-point	10 significant digits

ان المتغير الصحيح integer variable عندما يعرف int او short فالفرق الوحيد انه يحجز 2byte من الذاكرة بينما تعريف العدد الصحيح باستخدام long سيحجز 4byte من الذاكرة مما يمكن من تخصيص ارقام اعلى من المراتب الصحيحة.

مثال (٢):

```
short kilo = 20;
int salary = 32767;  // -32769 <int<32767
long price = 4500000;
```

اما المتغير الحقيقي Real Variable فيستخدم لتمثيل الكسور العشرية والاعداد الحقيقية وبعدد معين من المراتب بعد الفارزة.

مثال (٣):

```
float Rate = 0.06;
double pi = 3.141592654;
```

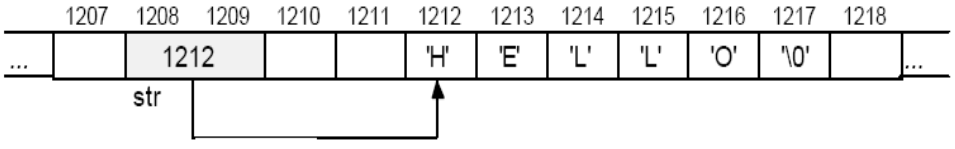
كما ان المتغير الرمزي character variable والذي يعرف بـ char يحجز بايت واحد من الذاكرة، والذي يأخذ قيمة رقمية التي تمثل الرمز وحسب النظام المستخدم في الحاسبة للتحويل الى لغة الماكنة وهو في الغالب نظام الآسكي ASCII (American Standard Code for Information Interchange) وهي الشفرة القياسية الامريكية لتبادل المعلومات، على سبيل المثال ان الرمز 'A' له رمز آسكي يساوي ٦٥ والرمز 'a' له رمز آسكي يساوي ٩٧، ويكون التعريف بالشكل الاتي:

```
char ch = 'A';
```

كما ان السلسلة string هي عبارة عن مجموعة من الرموز المتعاقبة، اذ ان متغير السلسلة يعرف كعنوان pointer لموقع اول رمز في الذكرة، اذ يمكن ان نوضح ذلك من خلال التعريف الاتي:

```
char *str = "HELLO";
```

اذ ان السلسلة ومتغير السلسلة في الذاكرة. كما في الشكل (١-١).



الشكل (١-١)

اذ ان كل سلسلة تنتهي في الذاكرة بـ "\0"، واذا كانت السلسلة الحرفية المدخلة اطول من السطر المخصص في البرنامج يمكن استخدام علامة backslash لكتابة سطر طويل.

مثال (٤):

```
"Example to show \
the use of backslash for \
writing a long string"
```

اذ ان السلسلة اعلاه تأتي في سياق انها تعمل عمل السطر الواحد، وكأنها مكتوبة بالشكل الاتي:

```
"Example to show the use of backslash for writing a long string"
```

مثال (٥): برنامج لحساب الاجر الاسبوعي لعامل باستخدام بعض المتغيرات البسيطة.

```
#include <iostream.h>
int main (void)
{
int workDays;
float workHours, payRate, weeklyPay;
workDays = 5;
workHours = 7.5;
payRate = 38.55;
weeklyPay=workDays*workHours*payRate; //arithmetic statement
cout << "Weekly Pay = ";    //output statement
cout << weeklyPay;
cout << '\n';
}
```

١-٥ عبارات الادخال والاخراج البسيطة:

تستخدم عبارة cout<< لإخراج وطباعة نتائج المعالجات والحسابات على شاشة المخرجات، كما تستخدم عبارة cin>> لإدخال قيم المتغيرات المحددة في البرنامج اذا لم يتم تخصيص اي قيم لهذه المتغيرات مسبقاً.

مثال (٦): يحسب مجموع رقمين مدخلين من لوحة المفاتيح:

```
#include<iostream.h>
void main()
{ int x,y,z;
cin>>x>>y;
z=x+y;
cout<<"the Result of z is: ";
cout<<z;
}
```

بعد تنفيذ البرنامج وادخال قيم x وy ولتكن x=3 وy=2 سيظهر الناتج الاتي:

```
the Result of z is: 5
```