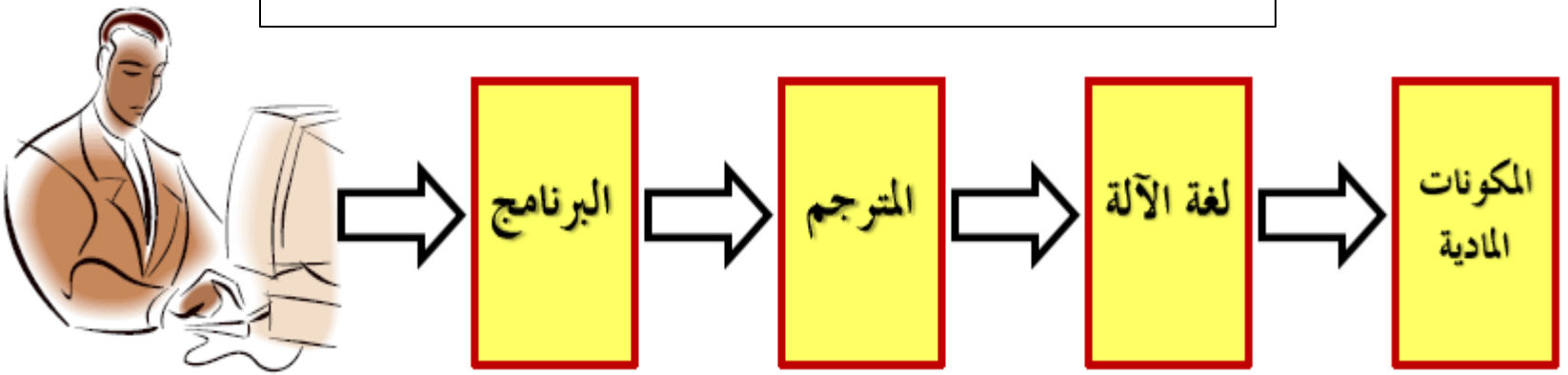
جامعة حماة
كلية العلوم في مصياف / السنة الأولى

المادة: البرمجة و الخوارزميات

المحاضرة الأولى: لغة البرمجة C++

C++ Programming Language

العام الدراسي ٢٠١٨ - ٢٠١٩



خطوات كتابة البرنامج

- قبل أن نتمكن من كتابة برنامج بلغة C++ أو أي لغة أخرى لا بد لنا أولا من فهم منطقي لجميع الخطوات الواجب إتباعها لحل مسألة ما بواسطة الحاسب.

١. **الخطوة الأولى**: تحديد و تعريف المشكلة

٢. **الخطوة الثانية:** تصميم البرنامج عن طريق كتابة خوارزمية حل المسألة Algorithm.

٣. **الخطوة الثالثة**: تحويل الخوارزمية إلى تعليمات بلغة C++ (برنامج) أو أي لغة برمجة أخرى.

٤. **الخطوة الرابعة**: تحويل تعليمات لغة C++ بدورها إلى لغة الآلة Binary Code المستعملة بواسطة برنامج المصنف Compiler و التأكد من خلوّها من كافة أنواع الأخطاء.

٥. **الخطوة الخامسة:** توثيق البرنامج

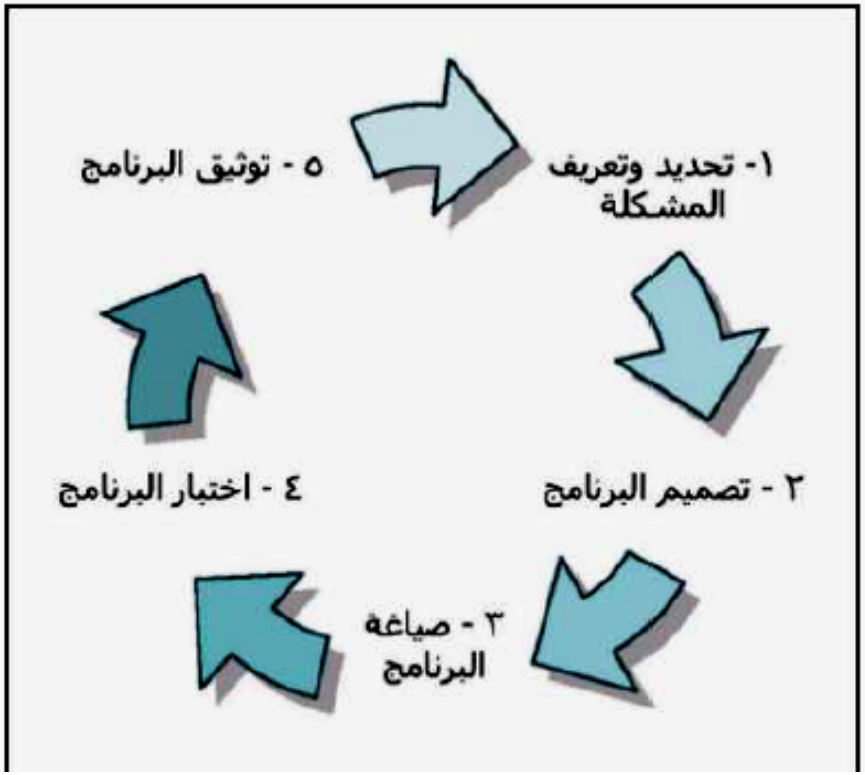
- بناءً عليه يجب التعرّف على مجموعة مفاهيم منها الخوارزمية و البرنامج و بعض المصطلحات و التسميات المستخدمة في برمجة الحواسيب.

خطوات صياغة وتطوير البرامج
Program Development Steps

١. تحديد وتعريف المشكلة Defining the Problem

في هذه الخطوة يقوم المبرمج بتحديد وتعريف المشكلة وتتضمن هذه الخطوة تحديد التالي بالترتيب:

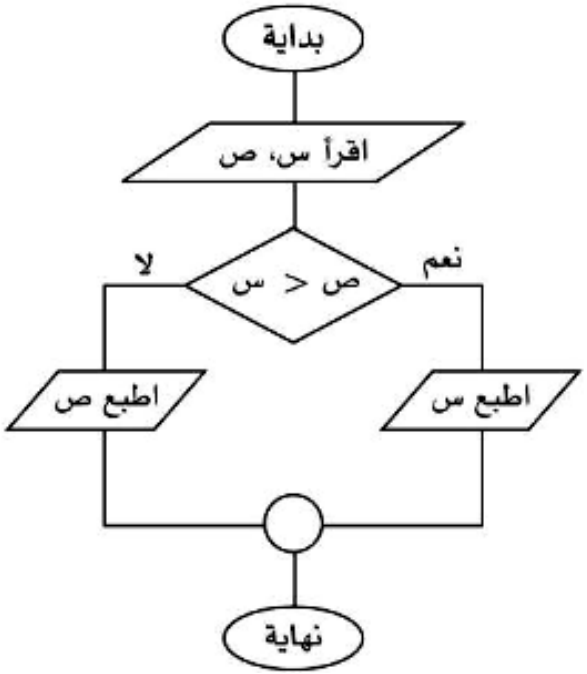
١. الهدف من البرنامج (حساب ارباح، فواتير استهلاك الماء والكهرباء، أو حساب معدل الطالب التراكمي)
٢. نوع وحجم المخرجات ووسائل الإخراج (تقارير – فواتير – شيكات – نقود ...)
٣. نوع وحجم البيانات المدخلة ووسائل الإدخال.
٤. مستخدمي البرامج والمستفيدين منه.



خطوات صياغة وتطوير البرامج
Program Development Steps

٢. تصميم البرنامج Design the Program

- يتم هنا تحديد المواصفات والخطوات الدقيقة والمرتبة منطقيا والتي تم فهمها ودراستها في الخطوة الأولى.
- ويتم ذلك بإستخدام عدة طرق منها **خرائط التدفق Flowchart** ويطلق عليها أيضاً خرائط سير العمليات وهي مجموعة من الرموز المتعارف عليها تستخدم لتوضيح الخطوات المنطقية اللازمة لحل مشكلة ما.



أهم الرموز المستخدمة في خرائط التدفق

الاسم	الرمز
بداية – نهاية Start - End	
مدخلات – مخرجات Input - Output	
معالجة Processing	
قرار Decision	

خطوات صياغة وتطوير البرامج
Program Development Steps

٣. صياغة البرنامج Coding the Program

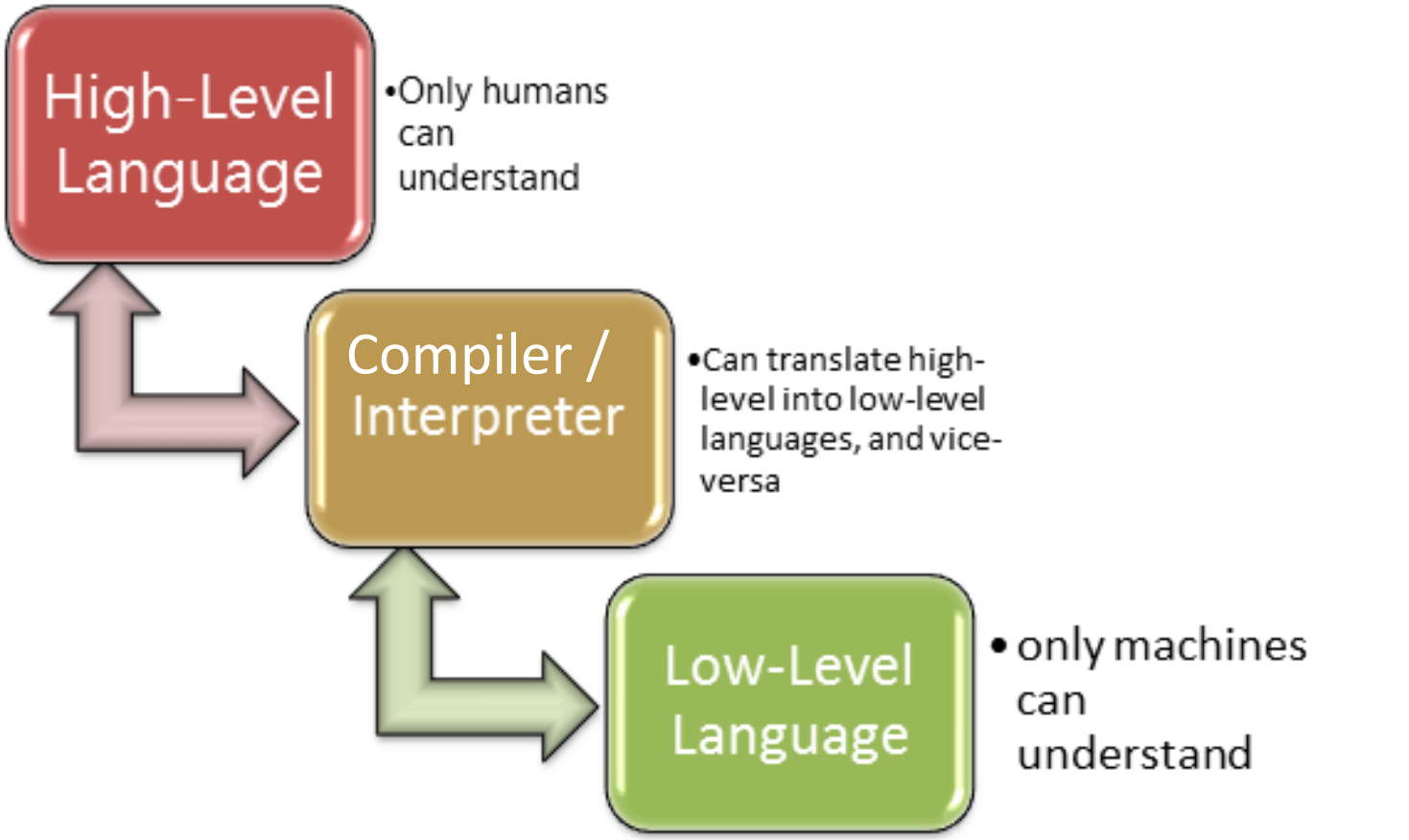
- بعد الانتهاء من تصميم البرنامج يتم اختيار إحدى لغات البرمجة المناسبة لصياغة أوامر البرنامج Coding وذلك بالاستعانة بخريطة التدفق Flow Chart أو غيرها.
- يجب عند صياغة البرنامج اتباع قواعد صيانة لغة البرمجة المستخدمة حيث ان لكل لغة برمجة قواعد خاصة بها ولا يعمل البرنامج اذا كان هنالك اخطاء املائية او اخطاء في قواعد اللغة Syntax Errors.

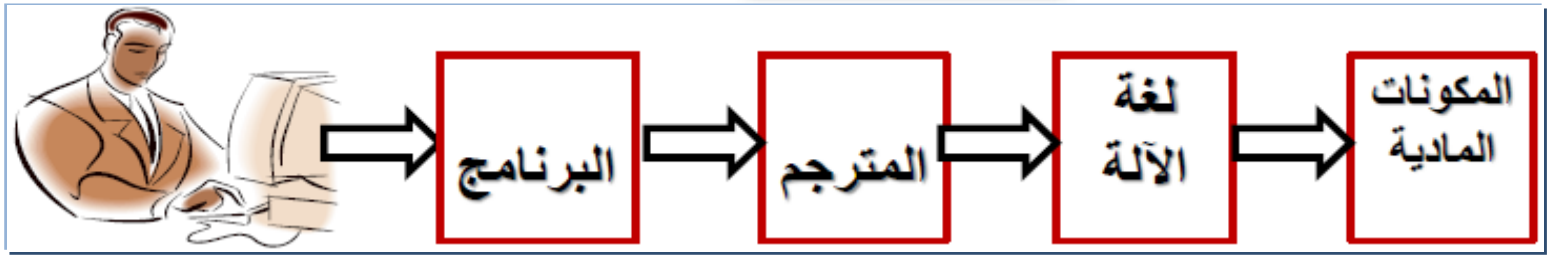
خطوات صياغة وتطوير البرامج
Program Development Steps

٤. اختبار البرنامج وتصحيح الأخطاء Program Debugging and Testing

- يسمى البرنامج بعد صياغتة باحدى لغات البرمجة **البرنامج المصدر Source Program** ولا يتم تنفيذه مباشرة على الحاسوب بل يتم ترجمته الى **برنامج مكتوب بلغة الآلة Object Program**.
- تسمى عملية تحويل البرنامج المصدر الى برنامج الهدف **بالترجمة Compilation** ويقوم بها برنامج يسمى **المترجم Compiler**.
- خلال عملية الترجمة Compilation قد تظهر أخطاء في صياغة البرنامج المصدر ينبغي

مراحل تنفيذ البرنامج





المفّسر (Interpreter) والمترجم (Compiler)

- برنامجي المفّسر والمترجم الخاصين بلغات برمجة الحاسب يقومان بترجمة اللغات العالية إلى لغة الآلة ليتمكن جهاز الحاسب من تنفيذ التعليمات والأوامر بسرعة عالية جداً.
- **برنامج المفسّر Interpreter:** يقوم بترجمة أوامر وتعليمات البرنامج المكتوب بإحدى لغات البرمجة سطراً سطراً إلى لغة الآلة أثناء قرائنها ليتم تنفيذها سطراً سطراً أيضاً. ويمتلك المفسّر القدرة على التفاعل مع البرنامج أثناء تنفيذه وإجراء أي تغيير في البرنامج ثم متابعة التنفيذ.
- **برنامج المترجم Compiler:** يقوم بترجمة أوامر وتعليمات البرنامج المكتوب بإحدى لغات البرمجة ككتلة واحدة إلى لغة الآلة أثناء قرائنها، ليتم تنفيذها دفعة واحدة، ويعتبر أسرع من برنامج المفّسر Interpreter بعدة مرات في تنفيذ البرنامج، ولكن عند وجود أي خطأ في البرنامج لابد من إصلاح الخطأ وإعادة تنفيذ كل البرنامج من جديد.

مقاطع الذاكرة Memory Segment's

يقوم المترجم Compiler بتقسيم ذاكرة الحاسب إلى المقاطع التالية:

- **مقطع الترميز Coding Segment**: لتخزين نص البرنامج بعد تحويله إلى لغة الآلة (1 , 0).
- **مقطع المعطيات Data Segment** : لتخزين قيم المتحولات المعرّف عنها في بداية البرنامج.
- **مقطع المكّدس Stack Segment** : لتخزين قيم المتحولات المصّرح عنها في الدوال والإجراءات .
- **الكوّمة Heap** : القسم الحر من ذاكرة RAM، ويمكن أن يستخدم لتخزين قيم المتحولات الديناميكية فقط.
- تجدر الإشارة إلى وجود مجموعة من الدوال والإجراءات الخاصة بالتعامل مع ذاكرة الحاسب حيث تمكننا من التحكم ببعض العمليات ضمن الذاكرة بمختلف مقاطعها.

خطوات صياغة وتطوير البرامج
Program Development Steps

- هناك ثلاث انواع من الأخطاء:

١. **أخطاء في قواعد اللغة Syntax Errors:** اخطاء املائية في كتابة الأوامر.

٢. **أخطاء منطقية Logical Errors:** لا يكتشفها الحاسوب وتظهر عند تنفيذ البرنامج على عينه من البيانات فنحصل على نتائج خاطئه او غير متوقعة، ويقوم المبرمج بتتبع خطوات البرنامج لمعرفة مصدر الخطأ وتصحيحه وتسمى هذه العملية Tracing.

٣. **أخطاء اثناء التشغيل Run-Time Errors:** تظهر عند تنفيذ البرنامج مثل عدم حجز مساحة كافية للمدخلات او الدخول في دوران بلا نهاية، وتظهر رسالة بنوع الخطاء.

٥. توثيق البرنامج Documenting the Program

- في هذه المرحلة تتم كتابة وصف تفصيلي لصياغة البرنامج، ويشمل هذا التوثيق أصل المشكلة وخطوات الحل وخرائط الحل وتعليمات التشغيل ومتطلبات التشغيل والمدخلات والمخرجات وكيفية التحكم في البرنامج في المواقف المختلفة.

لغة البرمجة C++

- تعتبر C++ من لغات البرمجة القوية التي تمّكن المبرمج من كتابة مجموعة الأوامر و الإيعازات لحل مشكلة معينة على شكل برامج يتم تنفيذها بسرعة عالية. وتتميز هذه اللغة بأنها تمّكن المبرمج من كتابة البرامج الموّجهة المرّكبة الأفضل كفاءة . حيث تقسم لغات البرمجة إلى قسمين :

١- **لغات البرمجة ذات المستوى العالي High Level Languages** :

وهي اللغات التي يكتب بها البرنامج بطريقة تكون اقرب إلى لغة الإنسان وبالتالي ابعد من فهم الحاسبة (أعلى من فهم الحاسبة) . لذلك فأن البرنامج المكتوب بهذه اللغة لا يفهم بصورة مباشرة من قبل الحاسبة . ومن هذه اللغات لغة Pascal, Fortran ,ADA ,Algol,... . إن برنامج لغات المستوى العالي يمر بمرحلتين قبل التنفيذ :

١- **مرحلة الترجمة Compilation Stage**: يتم في هذه المرحلة تصحيح الأخطاء القواعدية والإملائية للبرنامج .

٢- **مرحلة التفسير Interpretation Stage**: حيث يتم في هذه المرحلة تحويل البرنامج الصحيح قواعدياً إلى صيغة أخرى مفهومة من قبل الحاسبة والتي تسمى لغة الآلة Machine Language

٢-**لغات البرمجة ذات المستوى المنخفض: Low Level Languages**

وهي اللغات التي يكون فيها البرنامج مكتوب بطريقة تفهم من قبل الحاسبة بصورة مباشرة . هذه اللغة تسمى بلغة الآلة مثل لغة Assembly . لذلك فأن البرنامج المكتوب بهذه اللغة ينفذ بشكل أسرع لأنه لا يمر بالمرحلتين السابقتين .

- لغة C++ هي لغة وسطية Middle Level Language مابين لغات المستوى العالي ولغات المستوى المنخفض، لذلك فأن البرنامج المكتوب بهذه اللغة يترجم وينفذ بشكل أسرع من لغات المستوى العالي .

الحاسوب و البرمجة

- الأوامر والتعليمات المكونة للبرنامج تنقسم إلى أربعة أنواع:
  - ✓ تعليمات لقراءة البيانات من وحدات الإدخال.
  - ✓ تعليمات لإجراء العمليات الحسابية والمنطقية على البيانات.
  - ✓ تعليمات لإخراج البيانات على وحدات الإخراج.
  - ✓ تعليمات لتخزين البيانات في الذاكرة الرئيسية أو الثانوية.

- **تحذير**: حساسية اللغة لحالة الحروف

لغة C++ حساسة لحالة الحروف بمعنى أنها تفرق بين الحروف الكبيرة Capital و الصغيرة Small. لذلك، إذا استخدمت متغيراً باسم sum وآخر باسم Sum فإن لغة C++ ستعتبرهما متغيرين مختلفين بسبب الحرف الأول. كذلك، إذا استخدمت الكلمتين int و INT، فإن لغة C++ ستعتبرهما كلمتين مختلفتين.

- لذا يجب الالتزام بحالة الحروف الصحيحة عند كتابة برامج بلغة C++.

مكونات لغة البرمجة C++

١- الكلمات المحجوزة Reserved Words

هذه الكلمات يكون لها صيغ إملائية محدد يجب الالتزام بها عند كتابة البرنامج وإلا سيكون البرنامج محتوياً على خطأ و لا يمكن تنفيذه. الصيغة الإملائية للكلمات المحجوزة للغة C++ تتكون من مجموعة من الحروف الصغيرة Letters Lower Case حيث **تحتوي لغة C++ على ٣٢ كلمة قياسية** وهذه الكلمات لها مكان معين تستخدم فيه للأداء غرض معين مثل :

for , static , if , do , double ,int,.....

٢- المعَرف Identifier :

يشمل كل الأسماء التي تعرف من قبل المبرمج في البرنامج مثل أسماء المتغيرات والثوابت والأنواع البيانية واسم البرنامج ويتكون المعرف من حرف واحد على الأقل صغير أو كبير أو underscore متبوع بسلسلة من الحروف أو الأرقام أو underscore وغير ذلك يعتبر غير مقبول من قبل اللغة، مثال

Sum , text1 , a22 ,_ss , ft_2.... مقبول

3a_a , 2Max, m1*, d12$,..... غير مقبول

الكلمات الأساسية المحجوزة في لغة C++

asm	auto	bool	Break	case
catch	char	class	Const	const_cast
continue	default	delete	Do	double
dynamic_cast	else	enum	explicit	export
extern	false	float	For	friend
goto	if	inline	Int	long
mutable	namespace	new	operator	private
protected	public	register	reinterpret_cast	return
short	signed	sizeof	Static	static_cast
struct	switch	template	This	throw
true	try	typedef	Typeid	typename
union	unsigned	using	Virtual	void
volatile	wchar_t	while		

مكونات لغة البرمجة C++

٣-الدوال المكتبية Library functions :

- تمثل مجموعة من الدوال التي يتم استدعائها من المكتبة الخاصة بها .
- كل مكتبة في لغة C++ لها مجموعة من الدوال الخاصة بها لأداء شيء معين مثل:
  - مكتبة دوال القراءة والكتابة iostream.h
  - مكتبة الدوال الرياضية math.h
  - مكتبة معالجة السلاسل الرمزية string.h ،الخ.
- تقسم مكتبة الدوال إلى قسمين :

١- مكتبة الدوال القياسية Standard library functions

٢- مكتبة الدوال الخاصة بالصنف Class library functions

الشكل العام لبرنامج C++

```
1. #include <library name.h>
2. Public Declaration
3. Subprograms
4.  Main ()
5.   {
6.      Private Declaration
7.              Statements..
                Statements..
                Statements..
8.  Return  <value>;
9.    }
```

[Header Files]

[Program Body]

١. استدعاء المكتبات

٢. منطقة التصاريح العامة

٣. البرامج الفرعية

٤. الدالة الرئيسية

٥. بداية الدالة الرئيسية

٦. منطقة التصاريح الخاصة

٧. جمل برمجية

.....

.....

٨. القيمة التي ترجعها الدالة

٩. نهاية الدالة الرئيسية

شرح أجزاء برنامج C++

١- جزء استدعاء الدوال المكتبية Library Functions Call :

يتم في هذا الجزء استدعاء الدوال المطلوب التعامل معها في البرنامج . حيث يوجد هنالك أكثر من نوع من هذه الدوال مقسمة كل واحدة حسب عملها مثل مكتبة دوال القراءة والطباعة ومكتبة الدوال الرياضية ومكتبة الدوال الخاصة بالسلاسل الرمزية ،الخ . حيث يتم استدعاء مكتبة الدوال المطلوبة عن طريق الإيعاز #include

مثلاً لو أردنا استدعاء الدالة الخاصة بإيعازات القراءة والطباعة فأن الصيغة تكون كالتالي :

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

٢- جزء تعريف المتغيرات العامة (العالمية) Global Variables Declarations :

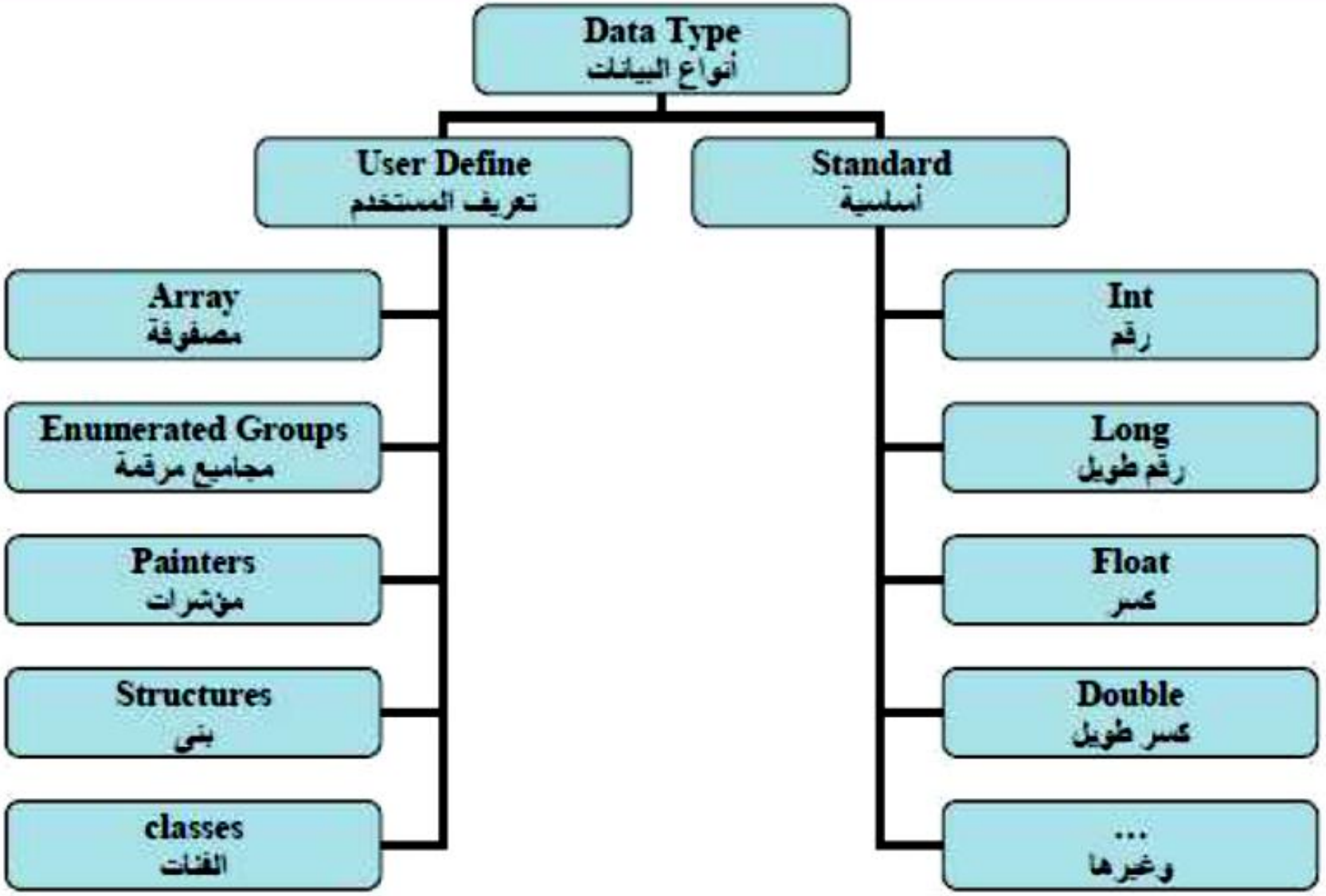
يمكن تعريف المتغير Variable بأنه عنوان موقع ذاكرة يستعمل لخزن قيمة معينة بشكل مؤقت طول فترة تنفيذ البرنامج ، وهذه القيمة المخزونة فيه يمكن أن تكون عدد صحيح سالب أو موجب ، عدد كسري سالب أو موجب أو أي رمز مثل الحروف أو الأرقام . أي يجب تحديد النوع البياني للمتغير لتحديد ماهية القيمة المخزونة ،حيث توضع كلمة تدل على النوع البياني قبل اسم المتغير للتصريح عن نوع المتغير .كل جملة تعريف متغير يجب أن تنتهي بفارزة منقوطة لدلالة على انتهاء جملة التعريف مثلاً :

```
int  count ;        ----> here count is Global variable
```

لذلك فإن المتغيرات تنقسم بشكل رئيسي إلى قسمين المتغيرات العامة والمتغيرات المحلية .

- المتغيرات العامة: هي المتغيرات التي تكون معروفة في أي جزء من أجزاء البرنامج ، والتي لو أردنا التصريح عنها فيتم ذلك في هذا الجزء .
- المتغيرات المحلية: هي المتغيرات التي يمكن استخدامها فقط ضمن الدالة المعرفة ضمنها تلك المتغيرات .

أنواع البيانات في لغة C++



الأنواع البيانية القياسية في لغة C++

- يوجد خمس أنواع بيانية والتي تعتبر أساسية في لغة C++ :

١- **int :** يستعمل لتعريف المتغيرات التي تكون قيمها أعداد صحيحة سالبة أو موجبة مثل :

22 , 30, 1000, 0, -3 , -89 ,...

٢- **float** : يستعمل لتعريف المتغيرات التي تكون قيمها أعداد حقيقية (كسرية) سالبة أو موجبة مثل :

0.66 , -1.276 , 99.78 , ...

٣- **double** : لتعريف المتغيرات التي تكون قيمها أعداد حقيقية (كسرية) سالبة أو موجبة وذات عدد مراتب بعد الفارزة أكثر من ستة أرقام . أي يستعمل للأعداد الكسرية الكبيرة .

٤- **char :** يستعمل لتعريف المتغيرات التي تكون قيمها رموز مثل الحروف الصغيرة والكبيرة والعمليات الرياضية ....الخ .

٥- **void:** يستخدم لتعريف المتغيرات عديمة القيمة . وهو غالباً ما يستخدم مع الدوال الفرعية كما تم توضيحه مع الدالة main .

الأنواع البيانية غير القياسية في لغة C++

يوجد نوع واحد من البيانات الغير القياسية وهو السلاسل الرمزية String . هذا النوع يستعمل لتعريف المتغيرات التي تكون قيمها سلسلة من الرموز (ماعدا رمز الفراغ) مثل أسماء الأشخاص وغيرها من الأسماء أو العبارات .مثل: Programmer , computer , Mohammad ,My_name, . . .

هذا النوع يعرف على شكل مصفوفة من الرموز كالتالي : char str[10] ;

أي أن المتغير str يمكن أن يخزن سلسلة رمزية طولها عشرة رموز .

- **الثوابت Constants :**

الثوابت تشير إلى قيم ثابتة التي لا يمكن تغيرها في البرنامج . الثوابت يمكن أن تكون أي نوع من الأنواع البيانية القياسية . يوجد في لغة C++ طريقتين لتعريف الثوابت والتي هي :

1-using the word const: Ex :

```
Const char ch='A';
Const int x=100;
```

2-using user defined data type: Ex:

```
#define PI  3.14          /*  no need for semicolon " ; "   */
 #define t  -180
```

شرح أجزاء برنامج C++

٣-جزء كتابة البرامج الفرعية Subprograms :

حيث يتم في هذا الجزء كتابة البرامج الفرعية (الدوال) التي سوف نقوم بدراستها بشكل موسع فيما بعد .

٤- الدالة الرئيسية main() :

الأجزاء الثلاثة المذكورة أعلاه هي أجزاء اختيارية Optional أي يمكن أن تكتب في البرنامج أو لا تكتب و تكون موجودة في البرنامج عند الحاجة .

أما هذا الجزء فهو جزء رئيسي في البرنامج ،فلا يمكن كتابة برنامج لا يحتوي على الدالة main ،هذه الدالة يمكن أن ترجع قيمة، هذه القيمة يجب تحديد نوعها بكلمة تقع قبل كلمة main .مثل

```
int Main( )
```

شرح أجزاء برنامج C++

هيكل هذه الدالة مقسوم إلى ثلاثة أجزاء محصورة بين قوس بداية الدالة { وقوس نهاية الدالة } هذه الأجزاء هي :

أ- جزء تعريف المتغيرات المحلية Local variable Declaration :

في هذا الجزء يتم تعريف المتغيرات المحلية ،أي المتغيرات الخاصة بالدالة main .

- لقد تميزت لغة C++ عن باقي اللغات الأخرى بأن هذه المتغيرات يمكن أن تعرف في أي جزء داخل الدالة main ولكن بشرط قبل الجمل التي تستخدم هذه المتغيرات ،أي لا يشترط تعريف هذه المتغيرات في بداية الدالة . طريقة التعريف نفسها للمتغيرات العامة ولكن مكان المتغيرات مختلف مثل :

```
int x, y;
double t;
char g;
```

- يمكن إعطاء قيمة للمتغير أثناء تعريفه ، مثل :

```
int x=100;
char ch='A' ;
```

شرح أجزاء برنامج C++

ب- جزء كتابة جمل البرنامج Statements Code :

هذا الجزء يمثل كتابة الجمل البرمجية الخاصة بلغة C++ ، حيث يشترط وضع الفارزة المنقوطة في نهاية كل جملة للدلالة بأن هذه الجملة قد انتهت والذي بعدها جملة جديدة وهكذا إلى أخر جملة في البرنامج.

ج- الإيعاز return : وهو يشير إلى القيمة التي سوف ترجعها الدالة ،بحيث هذه القيمة يجب أن تكون من نفس النوع البياني (Data Type) المحدد قبل main .

في لغة C++ يمكن للدالة أن ترجع قيمة معينة أو لا ترجع شيء عن طريق وضع كلمة void قبل main والتي تعني لاشيء . في هذه الحالة تحذف كلمة return من البرنامج لأنه لا توجد قيمة راجعة .

العمليات الحسابيـة و المنطقية في لغة C++

العملية	الرمز
الجمع	+
الطرح	-
الضرب	*
القسمة	/
باقي القسمة (Modulus)	%
الزيادة بواحد	++
الإنقاص بواحد	--

Type	Operator
postfix	() ++ --
prefix	++ -- !
	* / %
	+ -
	< <= > >=
	== !=
	&&
	||
	= += -= *= /= %=

العمليات الحسابيـة و المنطقية في لغة C++

استخدامها	عمليات التساوي
التحقق من تساوي القيم	==
التحقق من عدم تساوي القيم	=!

استخدامها	العمليات المنطقية
عملية AND وتستخدم للربط بين جملتين	&&
عملية OR وتستخدم للربط بين جملتين	||
عملية NOT وتستخدم مع جملة واحدة فقط	!

استخدامها	عمليات المقارنة
التحقق من أن القيمة اليسرى أقل من القيمة اليمنى	<
التحقق من أن القيمة اليسرى أكبر من القيمة اليمنى	>
التحقق من أن القيمة اليسرى أكبر من أو تساوي القيمة اليمنى	>=
التحقق من أن القيمة اليسرى أصغر من أو تساوي القيمة اليمنى	<=

التعليقات Comments

- عبارة عن توضيحات يكتبها المبرمج و لا يتم تنفيذها أثناء ترجمة البرنامج:

١. تعليق السطر الواحد:

```
1. // This is a comment
2. // And this is another comment
```

٢. تعليق السطور المتعددة:

```
1. /*
2. This is a comment
3. In tow lines
4. */
```

```
1. /* This is
2. a comment
3. in three lines */
```

C++ Assignment statement عبارة الإسناد في لغة

- تستخدم عبارة الإسناد لخزن قيمة في متغير . والصيغة العامة لها هي :

<variable> = expression ;

Ex:

X= 5;

أي أن القيمة الموجودة في الطرف الأيمن تخزّن في الطرف الأيسر . هنا عبرّنا عن الطرف الأيمن بتعبير، وهذا التعبير يجب أن يؤدي إلى قيمة مثل قيمة عددية أو متغير يحتوي على قيمة أو عملية حسابية تؤدي إلى قيمة. فيما يلي بعض الملاحظات حول عبارة الإسناد :

١- إذا كان المتغير في الطرف الأيسر محتويا على قيمة سابقة فأن هذه القيمة سوف تحذف وتحل محلها القيمة الجديد .

٢- يمكن لعبارة الإسناد أن تخزن قيمة لأكثر من متغير في نفس الوقت ، مثلاً:

x = y = z = 50; // Multi Assignment

عبارات الطباعة في لغة C++

- إيعاز الطباعة في لغة البرمجة يمّكن المبرمج من عرض نتائج البرنامج على الشاشة Screen
- في لغة C++ يوجد عدة أنواع من الايعازات ، أغلب هذه الايعازات خاصة بلغة C العادية مثل printf و putc الخ.
- ولكن من أهم الايعازات الخاصة بلغة C++ هو إيعاز cout الموجود ضمن مكتبة iostream . الصيغة العامة لهذا الإيعاز هي :

```
cout << expression ;
```

- **ملاحظات** :

١. يجب الالتزام بالشكل العام لهذا الإيعاز من خلال كتابة الصيغة الإملائية للإيعاز بشكل صحيح متبوعة بعلامتي الرمز أصغر >> ومن ثم الشيء المراد طباعته مع الالتزام بوضع الفارزة المنقوطة في نهاية الجملة .

٢. علامتي الأصغر >> تسمى بعامل الحشر Insertion Operation والذي يقوم بحشر البيانات المراد طباعتها إلى الشاشة .

عبارات الطباعة في لغة C++

٣. لقد تم تسمية البيانات المراد طباعتها بـ Expression لدلالة على عمومية الشيء المطلوب طباعته . التعبير هنا يشمل كل شيء يؤدي إلى قيمة عددية أو قيمة رمزية مثل المتغيرات ،الثوابت ،القيم العددية (الصحيحة أو الكسرية) ، الرموز والسلاسل الرمزية . إمثلاً

cout << x;	طباعة قيمة متغير
cout <<-100 ;	طباعة عدد صحيح
cout <<0.025;	طباعة عدد كسري
cout << 2*x+y	طباعة ناتج دالة
cout << " Hello World " ;	طباعة سلسلة رمزية
cout << " A " ;	طباعة رمز

مثال على تعليمة الطباعة cout <<

برنامج يقوم بطباعة جملتين توضيحيتين على الشاشة

```
#include<iostream>
using namespace std ;
void main ()
{
char[25] line;
line = " Welcome in Hama University " ;
cout << line << endl ;
cout << "Faculty of Sciences" << endl;
}
```

تضمين تعليمات مكتبة iostream

استخدام تعليمات مكتبة std

استدعاء الدالة الرئيسية

بداية البرنامج

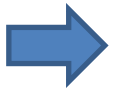
تعريف متغيّر

إسناد جملة إلى المتغير line

طباعة محتوى المتغير line والانتقال إلى سطر جديد

طباعة جملة توضيحية والانتقال لسطر جديد

نهاية البرنامج



Welcome in Hama University
Faculty of Sciences

الثوابت الرمزية ذات الشرطة المعكوسة في لغة C++

- بما أننا لا نستطيع استعمال بعض الرموز الموجودة في لوحة مفاتيح الحاسب، فقد استحدث لغة C++ شفرات رمزية خاصة تستعمل شرطة كثوابت رمزية معكوسة لها، وهذه الشفرات مدونة في الجدول التالي:

الشفرة	معناها	القيمة الصحيحة لها
"\b"	رجوع بمقدار خانة واحدة	8
"\n"	سطر جديد	13
"\t"	ترتيب أفقي	9
"\0"	للقيمة الخالية	0
"\r"	علامة رجوع	13
"\v"	ترتيب عمودي	11
"\\"	الشرطة المعكوسة \	92
"/f"	تقديم صفحة	12

عبارة إدخال القيم للمتغيرات المعرّفة (قراءة) cin>>

- تستخدم تعليمة cin >> لإدخال قيم للمتغيرات التي تم تعريفها في بداية البرنامج والبقاء على نفس السطر بانتظار إدخال قيمة.
- دائماً ينتهي السطر البرمجي بفاصلة منقوطة للدلالة على انتهاء تعليمة الإدخال.
- **مثال:** إدخال قيمة لمتغير واحد x cin>>x;
- يمكن استخدام تعليمة cin >> لإدخال قيم لمجموعة متغيرات على نفس السطر البرمجي.
- مثال: إدخال قيم لعدة متغيرات cin>>x>>y>>z;

مثال على تعليمة الإدخال cin >>

برنامج يطلب إدخال جملة إلى متغيّر معرّف في البرنامج، ثم يقوم بطباعة محتوى المتغيّر على الشاشة

```
#include<iostream>
#include<conio>
using namespace std ;
void main ()
{
char[20] w ;
cout << "enter your words: \n ";
cin >>w ;
clrsc();
cout << w ;
}
```

تضمين تعليمات مكتبة iostream

تضمين تعليمات مكتبة conio

استخدام تعليمات مكتبة std

استدعاء الدالة الرئيسية

بداية البرنامج

تعريف نوع المتغير w

طباعة جملة توضيحية

إدخال جملة إلى المتغير w

مسح شاشة الحاسب

طباعة محتوى المتغير w

نهاية البرنامج

مثال ٢ على تعليمة الإدخال cin >>

برنامج يطلب إدخال عددين صحيحين إلى ذاكرة الحاسب، ثم يقوم بطباعة ناتج جمعهما على الشاشة

```
#include<iostream>
using namespace std ;
int main ()
{
int Num1, Num2, Sum ;
cout << "enter Num1 & Num2: \n ";
cin >> Num1 >> Num2 ;
Sum = Num1 + Num2 ;
cout << "\n Sum = " << Sum ;
Return0 ;
}
```

تضمين تعليمات مكتبة iostream

استخدام تعليمات مكتبة std

استدعاء الدالة الرئيسية

بداية البرنامج

تعريف أنواع المتغيرات

طباعة جملة توضيحية

إدخال قيم إلى المتغيرات

جمع قيم المتغيرّين و وضع الناتج في متغير آخر

طباعة محتوى المتغير Sum

إعادة قيمة صفر للدالة main

نهاية البرنامج

اكتب برنامج لإدخال اسم وعمر ورقم هاتف ثم طباعة جملة توضيحية تتضمن البيانات المدخلة

```
#include<iostream>
using namespace std ;
void main ()
{
char[10] name;
int age;
long mobile;
cin >> name;      ← Ahmad
cin >> age;       ← 20
cin >> mobile;    ← 2112309
cout << "Hello, I am:" <<  name << “\t “ << age << “: Years old” << endl;
cout << “My phone number is:\n“ << mobile;
 }
```

النتيجة:

Hello, I am: Ahmad 20: Years old
My mobile number is:
2112309

أو

```
cin >> name >> age >> mobile ;
```