وزارة التعليم العالي والبحث العلمي
الجامعة التقنية الشمالية
المعهد التقني الموصل

الحقيبة التعليمية

القسم العلمي: تقنيات الشبكات وبرمجيات الحاسوب

اسم المقرر:البرمجة بلغة C++

المرحلة / المستوى: الاول

الفصل الدراسي: الأول

السنة الدراسية:2024 -2025

معلومات عامة:

اسم المقرر:	البرمجة بلغة C++ / programming in C++				
القسم:	تقنيات الشبكات وبرمجيات الحاسوب				
الكلية:	المعهد التقني الموصل				
المرحلة / المستوى	الأول				
الفصل الدراسي:	الأول				
عدد الساعات الاسبوعية:	نظري	2	عملي	3	
عدد الوحدات الدراسية:	5				
الرمز:	CPN100				
نوع المادة	نظري		عملي		كلهما	نعم
هل يتوفر نظير للمقرر في الاقسام الاخرى	لا				
اسم المقرر النظير (ان وجد)	/				
القسم	/				
رمز المقرر النظير	/				
معلومات تدريسي المادة					
اسم مدرس (مدرسي) المقرر:	م.م/ مروة نجم عبد				
اللقب العلمي:	مدرس مساعد				
سنة الحصول على اللقب	2023				
الشهادة :	ماجستير				
سنة الحصول على الشهادة	2018				
عدد سنوات الخبرة (تدريس)	6				

الوصف العام للمقرر

مادة C++ هي مادة أساسية تهدف إلى تزويد الطلاب بالمعرفة والمهارات اللازمة لتطوير البرمجيات باستخدام لغة C++ حيث تعتبر واحدة من أكثر لغات البرمجة شيوعًا واستخدامًا في مجالات متعددة، بما في ذلك تطوير الألعاب، نظم التشغيل، والبرمجيات التجارية, وهي منصة قوية لتطوير المهارات البرمجية الأساسية والمتقدمة، مما يساعد الطلاب على الاستعداد لسوق العمل أو لمزيد من الدراسات في مجالات البرمجة المختلفة.

الاهداف العامة

- سيتمكن الطالب من اتقان الأساسيات و فهم المبادئ الأساسية للبرمجة.
- تطوير وتنمية مهارات حل المشكلات البرمجية وتشجيع الطلاب على التفكير النقدي.
- تعزيز المهارات العملية وتوفير فرص للطلاب لتطبيق المهارات المكتسبة من خلال تطوير برامج صغيرة.
- سيتمكن الطالب من تعلم أساليب البرمجة الجيدة.

الأهداف الخاصة

- تعريف الطلاب ببنية لغة c++ ومكوناتها الأساسية، بما في ذلك القواعد النحوية.
- تمكين الطلاب من التعامل مع المتغيرات واستخدام أنواع البيانات المختلفة.
- تعليم الطلاب كيفية استخدام وتطبيق الجمل الشرطية والحلقات للتحكم بتدفق البرنامج.
- تعليم الطلاب كيفية تعريف وانشاء الدوال، مما يسهل تنظيم الكود البرمجي.
- تعزيز فهم الطلاب لأهمية كتابة كود نظيف وقابل للصيانة.
- تشجيع الطلاب على تطبيق المفاهيم من خلال مشاريع عملية ومسائل برمجية مختلفة.

الأهداف السلوكية او نواتج التعلم

أهداف تدريسية:

بعد الانتهاء من الدرس سيكون الطالب قادرا على ان:

- كتابة برنامج بسيط يستخدم المتغيرات، الأنواع، والجمل الشرطية.
- ان يقوم الطالب بإنشاء برنامج يستخدم الحلقات for ، while لأداء عمليات متكررة .
- أن يكتب الطالب دالة تأخذ مدخلات وتعيد مخرجات، ويستخدمها في برنامج متكامل.
- أن يكتب الطالب كودًا نظيفًا ومنظمًا، مع تعليقات توضيحية لشرح السطور في البرمجة.
- أن ينجح الطالب في حل مجموعة من المسائل البرمجية باستخدام التقنيات التي تم تعلمها.
- ان يكون الطالب قادرا على تعزيز مهارات العمل الجماعي من خلال مشاريع جماعية.

المتطلبات السابقة

- **لا يوجد.**

الأهداف السلوكية او مخرجات التعليم الأساسية		
ت	تفصيل الهدف السلوكي او مخرج التعليم	آلية التقييم
1	**تطبيق أساسيات البرمجة**: أن يكون الطالب قادرًا على كتابة برنامج بسيط يستخدم المتغيرات، الأنواع، والجمل الشرطية.	اختبار عملي + واجب منزلي
2	**استخدام الحلقات**: أن يقوم الطالب بإنشاء برنامج يستخدم الحلقات **(for، while)**	اختبار قصير + واجب منزلي
3	**كتابة دوال فعّالة :**أن يكتب الطالب دالة تأخذ مدخلات وتعيد مخرجات.	اختبار قصير + واجب منزلي
4	**حل مسائل برمجية :** أن ينجح الطالب في حل مسائل برمجية باستخدام التقنيات المكتسبة.	اختبار منتصف الفصل + تحديات برمجية+ واجبات منزلية
5	**كتابة كود نظيف**: أن يكتب الطالب كودًا نظيفًا ومنظمًا مع تعليقات توضيحية.	تقييم الكود+ واجبات منزلية
6	**التعاون في المشاريع** : ان يشارك الطالب بفعالية في مشروع جماعي.	تقييم جماعي + ملاحظات من الزملاء

أساليب التدريس (حدد مجموعة متنوعة من أساليب التدريس لتناسب احتياجات الطلاب ومحتوى المقرر)

الاسلوب او الطريقة	مبررات الاختيار
1. يقوم المعلم بشرح المفاهيم الأساسية بشكل مباشر، مع تقديم أمثلة عملية وتطبيقها.	يتيح للطلاب الحصول على المعلومات بشكل منظم ومباشر، مما يسهل فهم الأساسيات، خاصة للمبتدئين
2. تنفيذ مشاريع عملية تتطلب تطبيق المفاهيم المكتسبة	يساعد في تعزيز الفهم العملي ويسمح للطلاب بتجربة الحلول البرمجية في سياقات حقيقية
3. استخدام المناقشات الجماعية والأسئلة التفاعلية.	يعزز المشاركة الفعالة، مما يساعد على تحفيز التفكير النقدي ويشجع على تبادل الأفكار.
4. استخدام الألعاب التعليمية لتعزيز المفاهيم.	يضيف عنصر المرح ويحفز الطلاب على التعلم بطريقة تفاعلية، مما يزيد من التفاعل والانتباه.
5. تشجيع الطلاب على تقديم دروس لبعضهم البعض.	يعزز الفهم العميق ويساعد الطلاب على تطوير مهارات التواصل والعرض.
6. استخدام الفيديوهات التعليمية والمحتوى الرقمي.	يوفر موارد مرئية تساعد في تعزيز الفهم، ويمكن للطلاب الوصول إليها في أي وقت
7. تقديم أمثلة واقعية لشرح كيفية استخدام المفاهيم.	يساعد الطلاب على رؤية تطبيقات المفاهيم في الحياة الواقعية، مما يزيد من ارتباطهم بالمادة.

الفصل الاول من المحتوى العلمي							
عنوان الفصل	الوقت						
التوزيع الزمني	النظري	العملي	العنوان الرئيسي	العناوين الفرعية	طريقة التدريس	التقنيات	طرق القياس
الأسبوع الأول	2ساعة	3ساعة	مقدمة عن لغة C++	شرح عام عن لغات البرمجة وتطورها	محاضرة	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة +واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		ماهي لغة البرمجة C++		عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة +واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		لماذا لغة C++ هي لغة برمجية		عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة +واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		كيفية تنصيب لغة c++		عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة +واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
الأسبوع الثاني	2ساعة	3ساعة	اساسيات لغة c++ وأنواع المتغيرات	شرح المتغيرات واسناد القيم	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		التدريب على كتابة برامج بسيطة	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة

الاسبوع الثالث	2ساعة	3ساعة	عمليات التشغيل في لغة البرمجة c++ C++operators	العمليات الحسابية Arithmetic Operators	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		الاسبقية في تنفيذ العمليات الحسابية	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		عمليات الاحلال Assignment Operators	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		Comparison and Logical Operators المقارنة والعمليات المنطقية	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
الأسبوع الرابع	2ساعة	3ساعة	الجمل البرمجية بلغة C++ C++ statements	الهيكل العام لكتابة برنامج c++ بلغة++ c	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		ايعازات الإدخال / الإخراج في C++	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة، أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		اشكال الدالة الرئيسية main	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		المكتبات في لغة c++	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		شرح امثلة لكتابة برامج بسيطة	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة

الأسبوع الخامس	2ساعة	3ساعة	المعادلات الرياضية والبرمجية+ جمل الاختيار Selection Statements	امثلة على حل المعادلات الرياضية بشكل برمجي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		ترتيب تقييم العوامل حسب الأولوية	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		شرح الفرق بين عوامل الزيادة والنقصان prefix notation and postfix notation	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		The Single if statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		If -else statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		Else-if statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة

الفصل الثاني							
عنوان الفصل	الوقت						
التوزيع الزمني	النظري	العملي	العنوان الرئيسي	العناوين الفرعية	طريقة التدريس	التقنيات	طرق القياس
الأسبوع السادس	2ساعة	3ساعة	جمل الاختيار Selection Statements (تكملة)	عبارة IF المتداخلة	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		الجمل الشرطية Conditional Statements	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		جملة switch – case	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة	اختبارات قصيرة +أسئلة في نهاية كل محاضرة

الفصل الثالث (من المحتوى العلمي)							
عنوان الفصل	الوقت						
التوزيع الزمني	نظري	عملي	العنوان الرئيسي	العناوين الفرعية	طريقة التدريس	التقنيات	طرق القياس
الأسبوع السابع	2ساعة	3ساعة	جمل الدوران Loop statements	جملة while	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
الأسبوع الثامن	2ساعة	3ساعة	مراجعة شاملة للمواضيع السابقة+ امثلة وتطبيق	مراجعة عامة	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
الأسبوع التاسع	2ساعة	3ساعة	جمل الدوران Loop statements (تكملة)	جملة الدوران for statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة على جملة الدوران for	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		تطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة

الأسبوع العاشر	2ساعة	3ساعة		جملة الدوران المتداخلة Nested loops	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة متنوعة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
الأسبوع الحادي عشر	2ساعة	3ساعة		جملة الدوران Do/while statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
الأسبوع الثاني عشر	2ساعة	3ساعة	جمل شرط التحكم Breaking Control Statements	جملة شرط التوقف Break control statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		جملة شرط الاستمرار Continue control statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		جملة Goto Statement	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة

	2ساعة	3ساعة		امثلة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة

الفصل الرابع								
عنوان الفصل	الوقت							
التوزيع الزمني	النظري	العملي	العنوان الرئيسي	العناوين الفرعية	طريقة التدريس	التقنيات	طرق القياس
الأسبوع الثالث عشر	2ساعة	3ساعة	المصفوفات Arrays ,المصفوفات الأحادية	تعريف المصفوفة في لغة c++ وفوائدها	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		المصفوفة الأحادية One Dimensional Array	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		الوصول لعناصر المصفوفة في C++	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		طريقة وضع قيم أولية لعناصر المصفوفة في C++	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		التعامل مع المصفوفة بواسطة حلقة التكرار في C++	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
الأسبوع الرابع عشر	2ساعة	3ساعة	المصفوفات الثنائية Two Dimensional array	المصفوفات الثنائية Two Dimensional array	عرض تقديمي، شرح، أسئلة وأجوبة, مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة	المصفوفات الثنائية Two Dimensional array
الأسبوع الخامس عشر	2ساعة	3ساعة	الدوال functions	أنواع الدوال	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلةوأجوبة,مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة
	2ساعة	3ساعة		امثلة وتطبيق عملي	محاضرة+ مختبر عملي	عرض تقديمي، شرح، أسئلةوأجوبة,مناقشة+واجبات	اختبارات قصيرة +أسئلة في نهاية كل محاضرة

المحتوى العلمي

خارطة القياس المعتمدة

المحتوى التعليمي	عناوين الفصول	الأهمية النسبية	الأهداف السلوكية					عدد الفقرات
				المعرفة	الفهم	التطبيق	التحليل	التقييم	
			النسبة المئوية						
الفصل الاول	**مقدمة واساسيات لغة C++**	%100	%20		%30	30%	10%	10%	15
الفصل الثاني	**جمل الاختيار Selection statements**	%100	%20		%30	30%	10%	10%	11
الفصل الثالث	**جمل الدوران Loop statements**	%100	%20		%30	30%	10%	10%	13
الفصل الرابع	**المصفوفات Arrays والدوال functions**	%100	%20		%30	30%	10%	10%	11
المجموع	**4**								**50**

المحتويات (الفصل الاول)

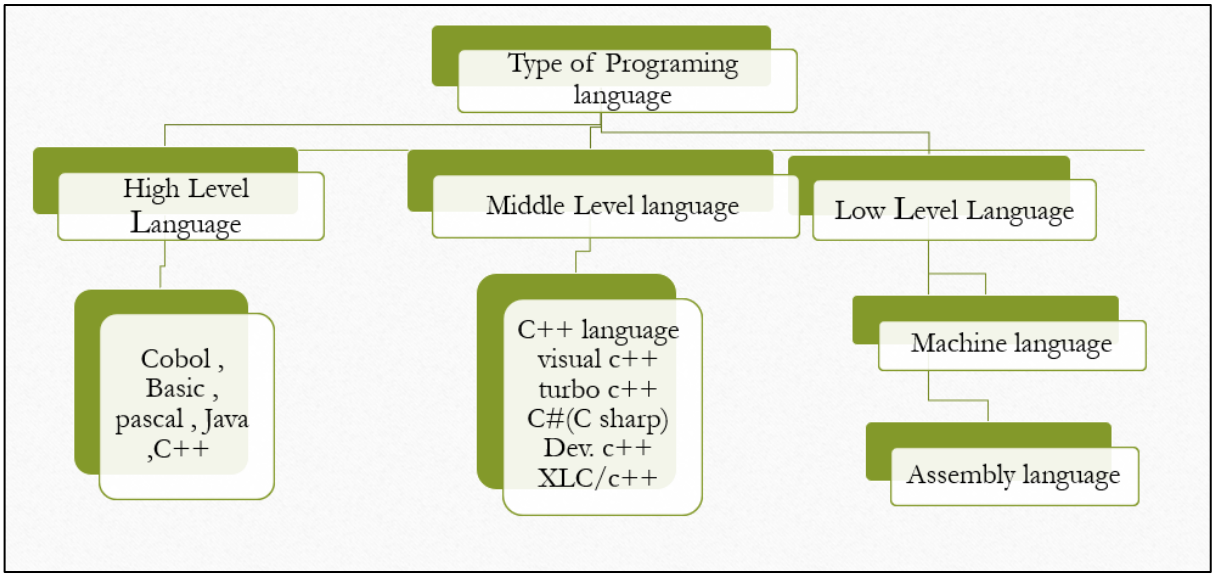
رقم المحاضرة: 1,2,3,4	
عنوان المحاضرة:	مقدمة واساسيات لغة C++
اســـم المدرس:	م.م/مروة نجم عبد
الفئة المستهدفة :	طلبة المرحلة الاولى
الهدف العام من المحاضرة :	تهدف هذه المحاضرة إلى تقديم لمحة شاملة عن لغات البرمجة بشكل عام، مع التركيز بشكل خاص على لغة C++، ومساعدتك على فهم أساسيات هذه اللغة. ستمكنك المحاضرة من التعرف على المفاهيم الأساسية للغة c++ وكذلك التعرف على أنواع المتغيرات فيها ،و تعريف الطلاب بالعمليات المختلفة مثل العمليات الحسابية (الجمع، الطرح، الضرب، القسمة) والعمليات المنطقية (مثل &&، ||، ! ،وعمليات الاحلال وكيفية استخدامها مما يتيح لك بناء قاعدة معرفية قوية تساعدك في تطوير مهاراتك البرمجية وفتح آفاق جديدة في مجالات البرمجة المختلفة.
الأهداف السلوكية او مخرجات التعلم:	1- تعريف الطلاب بمفهوم لغات البرمجة وأهميتها في تطوير البرمجيات والتطبيقات. 2- التعرف على الفروق بين لغات البرمجة عالية المستوى ومنخفضة المستوى. 3- التعرف على أنواع البيانات المختلفة في C++ مثل : int، float، char، و string. 4- ان يشرح الطالب مفهوم عمليات التشغيل في لغة C++ (الحسابية، المنطقية، والمقارنة). 5- ان يطبق الطالب العمليات الحسابية والمنطقية في برامج بسيطة بلغة C++. 6- توضيح مفهوم الجمل البرمجية (Statements) وكيفية تعلم كتابتها واستخدامها لتنفيذ تعليمات محددة 7- القدرة على كتابة برامج بسيطة باستخدام C++
استراتيجيات التيسير المستخدمة	التعلم التفاعلي- العرض المرئي(الشرائح التقديمية)- التطبيق العملي- التشجيع على التعاون-تعزيز التفكير النقدي- توفير موارد إضافية(فيديوهات, مواقع تعليمية...).
المهارات المكتسبة	• فهم أساسيات البرمجة • القدرة على كتابة شيفرات بسيطة واستخدام المكونات الأساسية للغة C++ . • تطبيق أنواع المتغيرات. • تطوير مهارات التفكير النقدي وحل المشكلات البرمجية من خلال التمارين العملية.
طرق القياس المعتمدة	الاختبارات القصيرة - المشاركة في الأنشطة الصفية- المشاريع العملية- التقييم الذاتي-المناقشات الصفية-التمارين العملية-واجبات منزلية.

1. **الاسئلة القبلية:**
   - ❖ لماذا تعتبر لغة C++ من اللغات المهيكلة والوسطية؟
   - ❖ ماذا تعرف عن الأغراض العامة التي تستخدم فيها لغة C++ ؟
   - ❖ ما هي الاسبقيات في تنفيذ العمليات الحسابية ؟
   - ❖ هل هنالك فرق ما بين العمليتين X++ و ++X
   - ❖ Write a c++ program to read 2 numbers and compute arithmetic operators (+,-,*,/).

2. **محتويات الفصل :**
   - ❖ مقدمة عن لغة c++
   - ❖ اساسيات لغة c++ وأنواع المتغيرات.
   - ❖ العمليات الحسابية Arithmetic Operators
   - ❖ الاسبقية في تنفيذ العمليات الحسابيةArithmatic operators as per precedence
   - ❖ الجمل البرمجية statements
   - ❖ شرح امثلة لكتابة برامج بسيطة
   - ❖ امثلة على حل المعادلات الرياضية بشكل برمجي
   - ❖ ترتيب تقييم العوامل حسب الأولوية
   - ❖ عوامل الزيادة والنقصان prefix notation and postfix notation
   - ❖ امثلة وتطبيق عملي

3. **المحتوى العلمي :**

مقدمة عن اللغات البرمجية بشكل عام



شرح عن المخطط السابق

- في بداية الحاسوب كانت البرمجة تتم على لغة بدائية منخفضة (واطئة) المستوى low level language تدعى لغة الالة machine language حيث كانت تفهمها هذه الالة مباشرة وتتم البرمجة عليها بأوامر مكونة من 0 و 1 ويسمى بالنظام الثنائي Binary number وكانت البرمجة عليها صعبة ومعقدة حتى تم تطوير لغة التجميع Assembly language وهي من اللغات المنخفضة المستوى أيضا حيث كانت سهلة بالنسبة للغة الالة. مع مرور الوقت تم تطوير لغات برمجية أخرى مثل Cobol و Basic و C و FoxPro وهذه تعتبر من اللغات العالية المستوى High level language وكان التعامل مع هذه اللغات بالكلمات والنصوص مما جعل هذه اللغات مقروءة.
- وتعتبر لغة C++ من اللغات الوسطية ما بين اللغات الواطئة المستوى واللغات العالية المستوى وتتميز بالسرعة في وقت التنفيذ بالنسبة للغات الاخرى.

صفات اللغات البرمجية

- **من صفات اللغات الواطئة المستوى :- Low level language**

يمكن الوصول الى أي موقع في الذاكرة عن طريق ذكر العنوان لذلك فان مستخدم اللغات الواطئة المستوى يحتاج الى ان يعرف تفاصيل عن ال micro processor وكيفية التعامل معه.

- **من صفات اللغات العالية المستوى :- High level language**

المستخدم لا يحتاج سوى الى معرفة عمل الايعازات التي يكتبها وتأثيراتها على البرنامج

- **من صفات لغة البرمجة C++ انها لغة وسطية :-**

- لأنه بإمكان هذه اللغة التعامل مع الـ register التابعة للـ micro processor
- ممكن التعامل مع البيانات (data) بصورة الـ bit و byte.
- ممكن التعامل مع عناوين الذاكرة عن طريق استخدام الـ pointer.

تعتبر لغة C++ من اللغات المهيكلة وذلك لانها تتعامل مع مايأتي:

- ✓ Subroutine (function)
- ✓ Loop and control statement such as :- for , do_while , while.
- ✓ Conditional statement such as : switch , case , if

تعتبر لغة C++ من لغات الأغراض العامة وتستعمل بكثرة في:

1- برمجة النظم system programming:- مثل نظام الـ Visual FoxPro احدى أنظمة قواعد البيانات.
2- أنظمة التشغيل operating system:- والذي يعتبر العقل المدبر للحاسبة مثل نظام التشغيل window 2010.
3- كما انها تدعم البرمجة الشيئية او الكائنية (oop).
4- استخدمت في تطبيقات الذكاء الاصطناعي مثلا العاب الشطرنج.
5- استخدمت في تصميم الألعاب (game development)
6- برمجة متصفحات الانترنت مثل جوجل كروم وفايرفوكس.

ملاحظة:- لا تفهم جميع الكومبيوترات الـ C++, أي نقصد عند جلب computer جديد فأنه لا يحتوي على أي لغة بهذا لم يفهم لغة الـ C++ اذن يجب تنصيب اللغة على الـ computer يعني يتم تنزيل برنامج المترجم (compiler) للغة الـ C++ وهذا البرنامج يحتوي على جميع الايعازات البرمجية الخاصة بهذه اللغة كأن تكون لغة C++ او turbo c++ او developer c++.

الصيغة العامة لكتابة برنامج الـ C++

```
#include<iosteam.h>
void main()
{
    statement 1;
    statement 2;
     .
     .
    statement n
}
```

1- المكتبات

- **مكتبة عامة لاوامر الادخال والاخراج:-** **#include<iostream.h>**
- **مكتبة الدوال الرياضية :-** **#include<math.h>**
- **مكتبة دوال معالجة النصوص:-** **#include<string.h>**
- **مكتبة دوال او امر الشاشة :-** **#include<conio.h>**
- **مكتبة عامة "اقدم مكتبة" لاوامر الادخال والإخراج :-** **#include<stdio.h>**

في هذا المكان يتم كتابة جميع المكاتب التي نحتاج اليها داخل البرنامج وكذلك المتغيرات التي تعرف بشكل عام لكل البرنامج والسجلات والدوال مثلا ايعاز الـ cin و cout والمستدعاة في مكتبة <iostream.h>

في أسماء المكتبات مثل (iostream.h , stdio.h) نجد انها تحتوي على حرفين:

- **i : يعني أوامر الادخال (input)**
- **o : يعني أوامر الإخراج (output)**

2- دالة الـ void main()

هذه الدالة هي الدالة الرئيسية في البرنامج ويتم فيها وضع الايعازات الرئيسية للبرنامج.

3- Block assignment using({) to beginning functions and using (}) to end function

هي المنطقة التي يتم بداخلها كتابة الايعازات البرمجية وتعريفاتها وغيرها:

ملاحظة 1 :- يجب وضع فارزة منقوطة (;) في نهاية أي جملة برمجية ويدل على ان الجملة البرمجية قد انتهت مثلا cout<<"hello student";

ملاحظة 2 :- يجب كتابة جميع الايعازات البرمجية بالحروف الصغيرة (small letter).

اساسيات لغة البرمجة C++:-

1- مجموعة الاحرف 1 - Character Set:-

▶ **تحتوي ال C++ على الحروف والارقام كما موضح ادناه:**

- **Upper case :- A,B,C,……Z**
- **Lower case :- a,b,c, ……..z**
- **Digits :- 0,1,2,……..9**

2- الرموز الخاصة 2- Special Characters

▶ **جميع الأحرف الأخرى الغير مسرودة تعالج كأحرف خاصة على سبيل المثال**

+	-	*	/	^
(	)	[	]	=
{	}	<	>	,
" (Double conations)	. (Dot)	: (Colon)	; (Semicolon)	(Blank Space)

▶ **لغة ال C++ قادرة على أن تميز الأحرف الكبيرة عن الأحرف الصغيرة ولذلك يصبح عدد الاحرف لها 52 حرف مثلا كلمة bag تختلف عن كلمة ل Bag وأيضا تختلف عن كلمة BAG.**

3- Key words (reserved words): الكلمات المحجوزة -3

هي ايضا متغيرات ولكن لايمكن استخدامها من قبل المستخدم كأسماء للمتغيرات او الثوابت نظرا لانها كلمات محجوزة وجميع الكلمات المحجوزة يجب ان تكتب بالحروف (lower case) الصغيرة

امثلة على الكلمات المحجوزة:-

cin	cout	for	case	char
short	eles	if	float	int
break	void	switch	main	long
delete	case	short	goto	double
public	sizeof	long	false	true

Data Type انواع البيانات:

متغيرات(variable) , ثوابت(constant) , مصفوفات(array) , الوظيفة (function) , أصناف (class) , السجلات(structure).

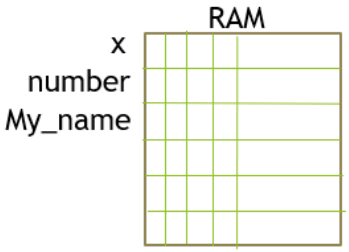
INTEGER	Bytes	REAL	Bytes	STRING	Bytes	LOGIC	Bytes	unsigned	bytes
short	2	float	4	char	1	bool	1	char	1
int	4	double	8	string	8			int	4
Long	4							short	2
								long	4

- **هذه المسميات موجودة في الذاكرة العشوائية (RAM) لكل نوع حجم معين**
- **1Kilobyte=1,000 byte 1Byte=8 bit 1Bit=0,1**

1Gigabyte=1,000,000,000 byte 1Megabyte=1,000,000byte

4- Variables 4- المتغيرات

▶ هي مواقع في الذاكرة تخزن فيها قيم معينة قد تتغير أثناء البرنامج او قد تبقى ثابتة. وأسماء المتغيرات يصطلحها المبرمج بغرض تخزين قيم في ذاكرة الـ RAM لتمكين المبرمج من اجراء العمليات المختلفة على المتغيرات, واذا أراد المبرمج ان يزيد او ينقص قيمة المتغير فيستطيع من خلال اسم المتغير.



▶ **شروط المتغير**

1. يبدأ بحرف مثلا my_name , number , x
2. لا يحتوي على فراغ مثلا my name , n umber غير مقبول برمجيا
3. لا يبدأ برقم مثلا 6number , 7x غير مقبول برمجيا
4. لا يحتوي على الرموز الخاصة **special characters** مثلا my-id , x-y غير مقبول برمجيا
5. لا يحتوي على كلمات محجوزة **reserved words** مثلا cout , cin , if , case

▶ مثال على المتغيرات المقبولة (valid variable): My_name , i , B

▶ مثال على المتغيرات الغير مقبولة (invalid variable): ros sal a()fest 3ab

أنواع المتغيرات:

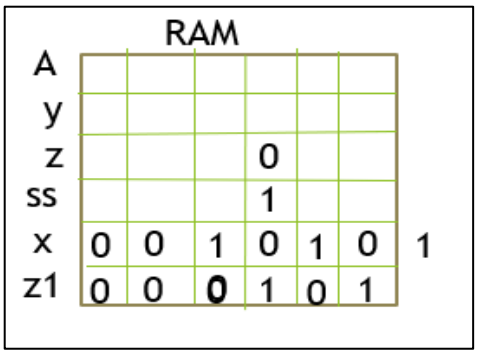
نوع المتغير	المتغير	استخدامه
Integer	int	يتم استخدامه في تعريف متغير من نوع رقم صحيح فقط مثل 5
float	float	يتم استخدامه في تعريف متغير من نوع عشري مثل 2.20
charcter	char	يتم استخدامه في تعريف متغير من نوع رمزي
double	double	يتم استخدامه في تعريف متغير من نوع حقيقي أي دقة مضاعفة
constant	const	يتم استخدامه في تعريف متغير من نوع ثابت لايمكن تغييره في البرنامج

الصيغة العامة لتعريف المتغيرات variable type variable name;

الصيغة العامة بتعريق القيمة الخاصة بالمتغير variable type variable name=value;

Example:-

- char A;
- int y;
- float z;
- double ss;
- char x='Y'; Y=84 =0010101
- short int z1=40; z1=40=000101
- unsigned short int X;
- int a,b,c,d; = int a; int b; int c; int d;
- int system=100;
- float zz=7.1234567; Double ss=12.123456789123456;



اسناد قيم للمتغيرات :- تكون طريقة اسناد قيمة او تعبير رياضي الى متغير يكون بالشكل التالي.

الصيغة العامة :-

variable name = value or mathematical expression;

- مثال :- الاسناد بطريقة مقبولة برمجيا

int y,x=5;

Y=x+6;

- مثال :- الاسناد بطريقة غير مقبول برمجيا

int y,x=5;

Y+3=x+6;

- المتغيرات الثابتة:- هي متغيرات تبقى ثابتة اثناء تنفيذ البرنامج ولا تتغير ابدا.
- الصيغة العامة:-

const data type variable = value;

const float pi=3.14;

فائدة الـ const حماية القيم التي نحتاجها كما هي ولا نريد ان يتم تغيير قيمتها بالخطأ.

5- constants — 5 - الثوابت:-

هناك ثلاثة أنواع من الثوابت:-

1- string constant — 1 -ثابت حرفي (سلسلة)
2- numeric constant — 2 -ثابت رقمي
3- character constant — 3 -ثابت حرفي

String constant: هي سلسلة من الاحرف الابجدية الرقمية محاطة بعلامات الاقتباس المزدوجة والحد الأقصى لطول السلسلة هو 255 حرف

مثلا "RE765.654" , "input integer number" , "the result"

Numeric constant: ثابت رقمي للارقام الموجبة او السالبة الصحيحة التي لا تحتوي على العلامة العشرية مثال long int x,y; short int x,y; int x,y;

Character constant:- الحرف يتمثل بعلامة الاقتباس المنفردة مثلا '#' , ':' , 'a' , 'A'
امثلة متنوعة عن الثوابت الحرفية

char x,y,z; char x='a'; char x;

استخدامات الشرطة المائلة للخلف (\)

▶ تستخدم الشرطة المائلة للخلف (\) مع الكلمة المحجوزة cout فقط وتكون ضمن علامتي اقتباس مزدوجة او منفردة :

الشرطة المائلة مع الحرف	المعنى	توضيح
\n	New line	طباعة سطر جديد
\t	8 Spaces (Tab)	طباعة 8 مسافات فارغة
\b	Backspace	الرجوع الى الخلف
\a	Sound "beep"	اصدار صوت من الجهاز
endl	New line	طباعة سطر جديد
ends	8 Spaces (Tab)	طباعة 8 مسافات فارغة
//	command	كتابة التعليقات. التعليقات مهمة جدا بجانب ا لاسطر البرمجية حتى فيما بعد اذا اردنا فهم البرنامج الذي كتبناه في وقت مضى ومجرد ان نقرأ التعليقات يكون فهم البرنامج لنا سهل
/* تعليق */	Multy line commands	نستطيع ان نكتب اكثر من سطر للتعليق multy line

C++ operators عمليات التشغيل فى لغة البرمجة++C

Operators: are symbols used in C++ to carry out operations. Operators are applied on operands to perform various operations.

عمليات التشغيل: هي رموز مستخدمة في c++ لتنفيذ العمليات ويتم تطبيق الـ operators على المعاملات لإداء عمليات مختلفة.

The different types of operators used in C++ are the following.

الأنواع المختلفة من عمليات التشغيل المستخدمة في الـ C++ هي كما يلي:

- Arithmetic Operators
- Assignment Operators
- Comparison and Logical Operators
- Relational Operators

1- Arithmetic Operators العمليات الحسابية

العمليات الحسابية: هذه العمليات تتطلب متغيرين ليتم تنفيذها.

Let us take an example of operands a, b with values 10 and 5 respectively.

لنأخذ مثالاً على المعاملات a و b مع القيمتين 10 و5 على التوالي.

Arithmetic operators in C++		
Operator	Description	Example
+	Addition of two operands	a + b = 15
-	Subtraction of right operand from the left operand	a - b = 5
*	Multiplication of two operands	a * b = 50
/	Division of left operand by the right operand	a / b = 2
%	Modulus – the remainder of the division of left operand by the right	a % b = 0
++	Increment Operator which increases the value of the operand by 1.	++b= 6
- -	Decrement Operator which decreases the value of the operand by 1.	--b= 4

The division result are: ناتج القسمة حسب نوع المتغيرات

Integer / integer = integer → 39 / 7 = 5

Integer / float = float → 39/7.0 = 5.57

float / integer = float → 39.0/7 = 5.57

float / float = float → 39.0/7.0 = 5.57

While 39%5 = 4 since 39 = 7 * 5 + 4

Arithmatic operators as per precedence الاسبقية في تنفيذ العمليات الحسابية

() for grouping the variables.

- Unary for negative number.
* / multiplication & division.
+ - addition and subtraction.

5 + 6 * 5 – 8 → 5 + (6 * 5) – 8 → (5+ 30) – 8 → 35 – 8 → 27

Example: X + Y * X – Z, Where X=5, Y=6, and Z=8.

2 -Assignment Operators in C++ عمليات الاحلال

عمليات الاحلال لديها الصيغة العامة التالية:

Variable = variable operator expression

Example: X=X+5; Y=Y*10;

يمكن كتابة عملية عمليات الاحلال (Assignment operator) بالشكل التالي (الصيغة العامة) :-

ملاحظة :-(كتابة عملية عمليات الاحلال بصيغة ثانية تعطي نفس المعنى):

Variable operator = expression;

Example X+=5; Y*=10;

عمليات الاحلال:- يتم إعادة استخدام الاحلال إلى متغير ، القيمة المعدلة تكون للمعالجة الحالية:

Assignment Operators in C++		
Operator	Description	Example
=	Assign right hand side (RHS) value to the left-hand side (LHS).	X=Y → X = Y يعمل إحلال منY الى X
+=	Value of LHS var. will be added to the value of RHS and assign it back to the var. in LHS.	X+=Y → X=X+Y
-=	Value of RHS var. will be subtracted to the value of LHS and assign it back to the var. in LHS.	X-=Y → X=X-Y
*=	Value of LHS var. will be multiplied to the value of RHS and assign it back to the var. in LHS.	X*=Y → X=X*Y
/=	Value of LHS var. will be divided to the value of RHS and assign it back 10 the var. in LHS.	X/=Y → X=X/Y
%=	The remainder will be stored back to the LHS after integer division is carried out between the LHS var. and the RHS var.	X%=Y → X=X%Y

This is a valid statements: هذه الجملة مقبولة برمجيا :

A=b=c+4;

C=3*(d=12.0/x);

Exercise:

Rewrite the equivalent statements for the following examples, and find it results

Assume: X=2, Y=3, Z=4, V=12, C=8.

أعد كتابة العبارات المكافئة للأمثلة التالية ، وإيجاد نتيجة كل واحدة منها ؟

افرض: X=2, Y=3, Z=4, V=12, C=8

Example	Equivalent Statement (الجملة المكافئة)	Result (النتيجة)
X+ = 5	X = X + 5	X ← 7
Y- = 8	Y = Y - 8	Y ← -5
Z* = 5	Z = Z * 5	Z ←
V/ = 4		V ←
C% = 3		C ←

3- Comparison and Logical Operators المقارنة والعمليات المنطقية

المقارنة والعمليات المنطقية : لديها ثلاثة أنواع من
1- العمليات العلائقية(المقارنة) (Relational operators)
2- عمليات المساواة (Equality operators)
3- العمليات المنطقية (Logical operators)

(a) (Relational operators):- العمليات العلائقية(المقارنة)

تتكون من الرموز التالية =< ,=> ,< ,> وسوف ترجع قيمة واحدة اما true او false او 0 او 1

الرمز	التوضيح
>	اكبر من
<	اصغر من
>=	اكبرمن او يساوي
<=	اصغرمن او يساوي

Example: 3 > 4 → false , 6<=2 → false , 10>-32 → true, (23*7)>=(-67+89) → true

(b) Equality operators: عمليات المساواة :تتكون من الرموز التالية:-

الرمز	التوضيح
==	يساوي
=!	لا يساوي

Example: a=4, b=6, c=8. a==b false, (a*b) != c true, s' == 'y' false

(c) Logical operators :- العمليات المنطقية

يتم إنشاء التعبير المنطقي من التعابير العلائقية عن طريق استخدام العمليات المنطقية (!) NOT , && AND , (||) OR.

الرمز	التوضيح
&&	AND "و"
!	NOT "النفي"
||	OR "او" ,

And (&&) Tables		
A	B	A&&B
T	T	T
T	F	F
F	T	F
F	F	F

And (&&) Tables		
A	B	A&&B
1	1	1
1	0	0
0	1	0
0	0	0

OR (||) Table		
A	B	A||B
T	T	T
T	F	T
F	T	T
F	F	F

OR (||) Table		
A	B	A||B
1	1	1
1	0	1
0	1	1
0	0	0

Not (!) Table	
A	B
T	F
F	T

Not (!) Table	
A	B
1	0
0	1

Example 1:- a=4, b=5, c=6

(a<b)&&(b<c)	(a<b) || (b>c)	!(a<b) || (c>b)	(a<b) || (b>c) && (a>b) || (a>c)
T && T	T || F	!(T) || T	T || F && F || F
T	T	F || T	T || F || F
		T	T || F
			T

Example 2:- Assume: X=0, Y=1, Z=1. Find the following expression:

M = ++X || ++Y && ++Z

Solution:- M = ++X || ++Y && ++Z

= T || (T && T)

= T || T

= T

(e) Special Operators: العمليات الخاصة:

1- Unary Operator: -1 عمليات أحادية

-	Negative value (minus sign).	قيمة سالبة (علامة الناقص)
++	Increment.	الزيادة بمقدار 1
--	Decrement.	النقصان بمقدار 1

ملاحظة :- هذه العمليات تطلب متغير واحد لتنفيذ العمليّة.

1- إشارة الـ - : المتغير يحمل إشارة سالبة مثلا x– او -30

2- ++ : الزيادة بمقدار 1 .

++x(المتغير يسبق الـ ++)	x ++(++ تسبق المتغير)
في حالة المتغير يسبق الـ ++ الزيادة بـ 1 يعني خرج القيمة ثم قم بالزيادة مثلا x++ الـ x=5 تخرج قيمة الـ x=5 ثم بعدها نقوم بالزيادة تصبح x=6	في حالة الـ ++ تسبق المتغير يعني زيادة بـ 1 ثم اخراج القيمة مثلا ++x الـ x=5 تصبح قيمة الـ x=6

3- وبنفس الطريقة في حالة استخدام إشارة -- .

2- Ternary Operator: -2 عمليات ثلاثية :

يطلق عليها بالعمليات الشرطية مثل if else

3- comma operator -3 الفارزة

تستخدم الفارزة للفصل بين المتغيرات عند تحديد نوع المتغير مثلا int a,b,c

4- Type Convertor 4 - انواع التحويلات:لدينا نوعين من تحويلات النوع

(A) Converting by assignment:	(B) Cast operator
Int x; float y; x=y;	Result = (int)(19.2/4); or Result = int (19.2/4);

1- Statements الجمل البرمجية

الجمل البرمجية في الحاسبة تقوم بتنفيذ بعض الفعاليات. وتحتوي على ثلاثة أنواع من الجمل هي تعابير رياضية وجمل مركبة وجمل شرطية.

Expression statement (تعابير رياضية)	Compound statement (جمل مركبة)	Control statement (جمل شرطية)

x=y;
sum=x+y;
	{
	a=b+c;
	x=x*x;
	y=a+x;
	}
		if(a>b)
		{
		a=1;
		k=a+1;
		}

2- Getting started with c++ البدء باستخدام c++

- الهيكل العام لكتابة برنامج بلغة c ++ النموذجي:
- رأس البرنامج program heading
- البداية Begin
- تعريف نوع المتغير type or variable declaration
- الجمل البرمجية statement of operation
- النتيجة Result
- نهاية البرنامج end

The keyboard and screen I/O instructions in C++ are :

ايعازات الإدخال / الإخراج الخاصة بلوحة المفاتيح والشاشة في C ++ هي:

(a) cout : جملة الإخراج

الصيغة العامة:- cout<<variable1<<variable2<< .....<<variable n

تلفظ (سي أوت cout)هو اختصار للجملة course output وهو كائن يقوم بإخراج ماياتي بعده ويظهره على وحدة الإخراج القياسية (الشاشة) وهي إحدى الخدمات التي تقدمها مكتبة .iostream.h

<< يسمى معامل الإخراج او يسمى بمعامل الوضع Put to operator يجبر على إرسال الأشياء التي على يمينه ان يظهر على يساره أي ما تكتبه بعد هذا القوس يقوم بإخراجه.

علامة التخصيص " ": كل ما سوف يكتب بداخله سوف يتم إظهاره على وحدة الإخراج القياسية (الشاشة)

(;) الفارزة المنقوطة تخبر أن الأمر انتهى وعليه أن يذهب للأمر التالي .وهي مهمة جدا في نهاية كل سطر في ما بين قوسين الدالة الرئيسية { } .(الدالة الرئيسية هي main)

EX: cout<<"C++ language"<<x++<<endl;

(b) cin : هذا الكائن(العبارة) تخزن الرقم الذي يكتبه المستخدم من لوحة المفاتيح في متغير يدعى integer والذي يلفظ كـ (سي ان c in) لوحة المفاتيح.

الصيغة العامة cin>>var.1>>var.2>>.....var.n

<< : ويأخذ عامل الحصول get from (>>) الأشياء الموضوعة على يساره ويضعها في المتغير الموجود على يمينه، عند تنفيذ هذه العبارة ينتظر البرنامج ان يكتب المستخدم رقما من نوع integer ويضغط على مفتاح Enter . ويتم تعيين القيمة التي أدخلها المستخدم إلى المتغير integer ويمكن استعمال عامل الحصول عدة مرات في نفس العبارة: cin >>

integer1>>integer2;

ويضغط المستخدم هنا Enter ، أو مفتاح المسافة space او مفتاح الـ tab بعد كل قيمة, قبل أن يكتب القيمة التالية، ولكنه من الأفضل عادة إدخال قيمة واحدة في كل مرة لتجنب الخطأ

cin>>x>>y;

Example 1

```
#include<iostream.h>
void main()
{
 //     A program to print welcome
   cout<<"welcome";
}
```

output

Welcome press any key to continue

#include<iostream.h>

بمعنى موجه.

include:بمعنى تضمين.

#include: هو عبارة عن موجه للمترجم لكي يقوم بإدراج ملف رئيسي ضمن الملف الأساسي ويجب أن يوضع الملف الرئيسي بين < >.

العلامتين **<>** التي تحصر اسم الملف الدليلي تعني ان المترجم يبدأ البحث عن الملف الدليلي في الفهرسة او المجلد (folder) الأساسي الذي يحوي كل الملفات الدليلية ويسمى include.

io : مختصر ل inout/output أي ايعازات الادخال والإخراج.

stream : مكتبة قياسية خاصة بالدخال والإخراج و ..... الخ

.h : أي الملف الدليل header file.

#include<iostream.h> : يسمى هذا بالمرشد المهيئ preprocessor directive وهو عبارة عن تعليمة للمصرف ان يدرج كل النص الموجود في الملف iostream.h في البرنامج. وهو ملف يجب تضمينه مع أي برنامج يحتوي على عبارات تطبع بيانات على الشاشة او تستقبل بيانات من لوحة المفاتيح ويسمى iostream ملف ترويسة header file.
تعتبر الدوال من أهم مقومات البرنامج في لغة C++ ونجد أن البرنامج يمكن ان يتألف من دالة واحدة او اكثر وانه لابد لكل دالة من اسم يدل عليها عند استخدامها واستدعائها.

void main() :- ال **void** وتعني ان هذه الدالة لا ترجع اي قيمة .

main() هي اللب الأساسي لاي برنامج ولن يعمل الا بدونها وهي بمعنى دالة وهي دالة مستقلة ينقل نظام تشغيل المتحكم إليهما. وهي جزء أساسي في برنامج ال C++. فان المترجم يبحث عن هذه الدالة أولا لتنفيذها اما اذا لم تكن موجودة فستظهر رسالة خطأ Error Massage توضح ذلك.
() : القوسين مهمة جدا فهي تعني أن هذا السطر عبارة عن دالة وليست متغير ،بمعنى آخر أنها تعني أن هذا الاسم هو اسم لدالة فبدون هذه الأقواس يترجمها المترجم على أساس انها اسم لمتغير فلهذا تعتبر هذه الأقواس مهمة جدا.
{} : هما جسم الدالة الرئيسية وتكتب بداخله الكود المراد ولا يصح كتابة كود خارجهما وإلا سيظهر خطأ.

هناك ثلاث اشكال للدالة الرئيسية main() وهي:

void main()	int main()	main()
{	{	{
.....	.....	.....
.....	.....	.....
}	return 0	}
	}	

المكتبة (math.h) : تحتوي على دوال رياضية كثيرة مثل:

توضيح	الرمز الرياضي	الدالة
القيمة الطلقة للاعداد الحقيقة	|x|	abs(x)
جا س		sin(x)
جتا س		cos(x)
ضا س		tan(x)
جا$^{-1}$ س		sinh(x)
جتا$^{-1}$ س		cosh(x)
ضا$^{-1}$س		tanh(x)
الاس	x^y	pow(x,y)
عكس اللوغاريتم	e^x	exp(x)
الجذر التربيعي	$\sqrt{x}$	sqrt(x)
اللوغاريتم	Log x	Log(x)
تقريب الكسور للاعلى		Ceil(x)
حذف الكسور		Floor(x)
اللوغاريتم الطبيعي	Ln(x)	Log(x)

Examples امثلة

Example 1 : **C++ program to print welcome.**

```
#include<iostream.h>
void main()
{
    // A program to print welcome
   cout<<"welcome";
}
```

Output:

```
Welcome
```

****

The endl is used in C++ to represent a new line , as shown in the following example:

Example 2 :

```
#include<iostream.h>
void main()
{
   cout<< " hallow"  << endl ;
   cout<< " students"  ;
}
```

Output:

```
hallow
students
```

****

The \n is a special escape code , also used in C++ to represent a new line , as shown in the following example:

Example 3 :

```
#include<iostream.h>
void main()
{
   cout<< " hallow  \n " ;
   cout<< " students" ;
}
```

Output:

```
hallow
students
```

<u>Example 4</u> : The following program reads three different inputs and outputs it.

الطريقة الأولى لقراءة المتغيرات:

```
#include<iostream.h>
void main()
{
   int num = 3 ;
cout<<" number = "<< num << "\n" ;
  char ch =  'a'  ;
cout<< " character = "<< ch<< "\n" ;
 float fa = 34.45 ;
cout<< " real number = "<< fa<< "\n" ;

}
```

```
Output:
Number=3
Character=a
Real number=34.45
```

<u>Example 5</u> : The following program reads three different inputs and outputs it.

الطريقة الثانية لقراءة المتغيرات:

```
#include<iostream.h>
void main()
{
   int n ;  float f ;   char c ;
   cout<< " input integer number: "<< "\n" ;
   cin >> n ;
   cout<< endl ;
   cout<< " input decimal number: "<< "\n" ;
   cin >> f;
   cout<< endl ;
   cout<< " input character "<< "\n" ;
   cin >> c ;
   cout<< endl ;

        }
```

```
Output:
input integer number: 5
input decimal number: 4.2
input character: A
```

Constants Example مثال على الثوابت

Example 6 : Write a C++ program that reads the radius of a circle then computes and outputs its area .

اكتب برنامج لقراءة نصف قطر دائرة ثم حساب وطباعة مساحة هذه الدائرة.

```
#include<iostream.h>
void main()
{
   const float  pi = 3.14 ;
   int r ; float c;
   cout<< " input the radius of circle : ";
   cin >> r ;
   cout<< endl ;
   c=r * r * pi ;
   cout<< " the area of circle :  "<< c ;
}
```

Output:

```
enter the radius of circle: 5

the area of circle: 78.5
```

Example on arithmetic operators مثال على العمليات الحسابية

Example 7 : the following program computes arithmetic operators.

```
#include<iostream.h>
void main()
{
   int  a , b , sum , sub , mul , div ;
  cout<< " Enter any two numbers : "<< endl ;
   cin >> a >> b  ;

   sum= a+b ;
  sub= a-b ;
   mul= a*b ;
   div= a/b ;

   cout<< " a= "<< a << " b= "<< b  << " sum = "<< sum << endl ;
   cout<< " a= "<< a << " b= "<< b  << " sub= "<< sub << endl ;
   cout<< " a= "<< a << " b= "<< b  << " mul= "<< mul << endl ;
   cout<< " a= "<< a << " b= "<< b  << " div= " << div << endl ;
 }
```

```
Output:
Enter any two numbers
10   20
A=10  b=20  sum=30
Sub=-10
Mul=200
Div=0
```

Example 1:

Write the following equation as a C++ expression: $f = \frac{a+b+c+d+e}{10}$

Solution: **الحل**

الخطوة الأولى: تحويل المعادلة الرياضية الى معادلة بلغة C++

F=(a+b+c+d+e)/10

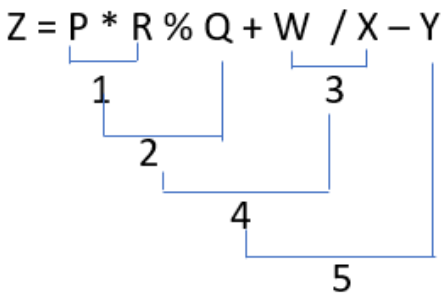
Example 2:

State the order of evaluation for the following expression:

Z=P * R % Q + W / X – y

Solution:

1- *
2- %
3- /
4- +
5- -



<u>Example 1</u> :Write C++ program to perform the above equation:

```
#include<iostream.h>
void main()
{
    int Z,P,R,Q,W,X,Y;
    cout<<"enter P:"; cin>>P;
    cout<<"enter R:"; cin>>R;
    cout<<"enter Q:"; cin>>Q;
    cout<<"enter W:"; cin>>W;
    cout<<"enter X:"; cin>>X;
    Z=P*R%Q+W/X-Y;
    cout<<"the result="<<Z<<endl;
}
```

OUTPUT ناتج التنفيذ

```
enter P:3
enter R:4
enter Q:6
enter W:5
enter X:7
the result=858993460
Press any key to continue
```

مكتبة الـ (math.h) تحتوي على أوامر الخاصة بالدوال رياضية تستخدم كما لافي المعادلات الرياضية

توضيح	الرمز الرياضي	الدالة
القيمة الطلقة للاعداد الحقيقة	|x|	abs(x)
جا س		sin(x)
جتا س		cos(x)
ضا س		tan(x)
جا-١ س		sinh(x)
جتا-١ س		cosh(x)
ضا-١س		tanh(x)
الاس	x^y	pow(x,y)
عكس اللوغاريتم	e^x	exp(x)
الجذر التربيعي	$\sqrt{x}$	sqrt(x)
اللوغاريتم	Log x	Log(x)
اللوغاريتم الطبيعي	Ln(x)	Log(x)

Example: Write the following equation as a C++ expression and state the order of evaluation of the binary operators:-

$$f=\sqrt{\frac{\sin(x)-x^5}{\ln(x)+\frac{x}{4}}}$$

اكتب المعادلة التالية كتعبير C ++ وحدد ترتيب تقييم العوامل الثنائي

solution: الحل

الخطوة الاولى :تحويل المعادلة من الصيغة الرياضيات الى صيغة البرمجة بلغة C++

f=sqrt((sine(x) – pow(x,5)) / (log(x) + x/4))

الخطوة الثانية : ترتيب التقييم حسب الأولوية order of evaluation

f=sqrt((sin(x) – pow(x,5)) / (log(x) + x/4))

1 2 4 5

3 6

7

8

س/ اكتب برنامج بلغة C++ لتنفيذ المعادلة أعلاه

Q/ Write C++ program to execute the above equation

برنامج بلغة C++

```
#include<iostream.h>
#include<math.h>
void main()
{
    int x;
    double f;
    cin>>x;
    f=sqrt((sin(x)-pow(x,5))/(log(x)+x/4));
    cout<<"f="<<f<<endl;
}
```

```
5
f=-1.#IND
Press any key to continue
```

Homework : **Write the following equation as a C++ expression and state the order of evaluation of the binary operators**

واجب :- اكتب المعادلة التالية كتعبير C ++ وحدد ترتيب تقييم العوامل الثنائي

$$Z = \sqrt{\frac{x2 - 3\sin(x)}{\tan x^3 + x^3/y}}$$

---

1. يمكن كتابة عاملي ++ و -- إما قبل المتغير (prefix notation) أو بعد المتغير (postfix notation) (تدوين كما في التالي:

- **تتم زيادة x قبل (أي يتم زيادة قيمة X ومن ثم تطبع على الشاشة)**

Prefix notation ++X

- **يتم زيادة x بعد (أي تطبع قيمة X على الشاشة ومن ثم يتم زيادة قيمةX)**

postfix notation X++

The difference between the Prefix and Postfix

الاختلاف ما بين زيادة قيمةX قبل وبعد

prefix

```
#include<iostream.h>
void main()
{
      int y;
      int x=7;
      cout<<++x<<endl;
      y=x;
      cout<<y<<endl;
}
```

output

```
8
8
Press any key to continue
```

Postfix

```
#include<iostream.h>
void main()
{
      int y;
      int x=7;
      cout<<x++<<endl;
      y=x;
      cout<<y<<endl;
}
```

output

```
7
8
Press any key to continue
```

4. **الأسئلة البعدية في نهاية الفصل:**

- ❖ ماهي الأسباب التي جعلت لغة C++ من اللغات المهيكلة والوسطية؟
- ❖ عدد الأغراض العامة التي تستخدم فيها لغة C++ ؟
- ❖ عدد الاسبقيات عند تنفيذ العمليات الحسابية .
- ❖ هل تختلف العملية X++ عن العملية ++X مع التوضيح.
- ❖ Write a c++ program to compute arithmetic operators (+,-,*,/) of two numbers and print the result.

- **المصادر الاساسية :**

1cpp.pdf (uotechnology.edu.iq)

م.م/ بشارسعدون مهدي, البرمجة المهيكلة c++ , الجامعة التكنولوجية قسم علوم الحاسوب

- **المصادر المقترحة: لايوجد**

- **روابط مقترحة ذات صلة: مجموعة فيديوهات للمحاضرات المشروحة في هذا الكتاب.**

https://www.youtube.com/playlist?list=PLnI1fRHezj5F8y024ulIKxV-DqdIyR4E6

المحتويات (الفصل الثانى)

رقم المحاضرة:5,6	
عنوان المحاضرة:	جمل الاختيار Selection statements
اســـم المدرس:	م.م/مروة نجم عبد
الفئة المستهدفة :	طلبة المرحلة الاولى
الهدف العام من المحاضرة :	
الأهداف السلوكية او مخرجات التعلم:	• أن يصمم الطالب حلولًا لمشاكل بسيطة باستخدام العمليات الحسابية والجمل البرمجية بلغة.C++ • أن يشرح الطالب مفهوم جمل الاختيار (مثل if، else if، switchودورها في التحكم في تدفق البرنامج • أن يطبق الطالب جمل الاختيار لحل مشكلات تتطلب اتخاذ قرارات داخل البرنامج.
استراتيجيات التيسير المستخدمة	التعلم التفاعلي- العرض المرئي(الشرائح التقديمية)- التطبيق العملي- التشجيع على التعاون-تعزيز التفكير النقدي- توفير موارد إضافية(فيديوهات, مواقع تعليمية...).
المهارات المكتسبة	• مهارات البرمجة الأساسية(تمكين الطلاب من فهم واستخدام الجمل البرمجية في لغة++C • مهارات حل المشكلات. • مهارات كتابة الأكواد. • مهارات التصحيح والتدقيق(القدرة على اكتشاف الأخطاء المنطقية واختبار الكود المكتوب والتحقق من عمل العمليات الحسابية والمنطقية بشكل صحيح.)
طرق القياس المعتمدة	الاختبارات القصيرة - المشاركة في الأنشطة الصفية- المشاريع العملية- التقييم الذاتي-المناقشات الصفية-التمارين العملية-واجبات منزلية.

1. **الاسئلة القبلية:**
   - اكتب برنامج بلغة ++C لقراءة رقم صحيح وفحص هذا الرقم هل هو زوجي ام فردي
   - اكتب برنامج بلغة ++C لقراءة عدد صحيح ، وطباعة رقم اليوم في الأسبوع

2. **محتويات الفصل :**

▶ **جمل الاختيار Selection Statements**

- Ifالشرطية وانواعها المختلفة
- Switch-case
- امثلة وتطبيق عملي

3. **المحتوى العلمي:**

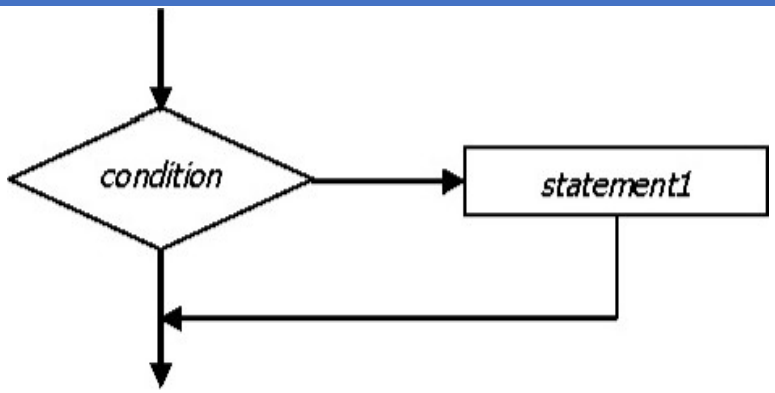
1. Selection Statements: جمل الاختيار

- تستخدم التعبيرات الشرطية بشكل أساسي في اتخاذ القرار. يوفر ++ C هياكل تحديد متعددة: if, if/else, else if, nested if and switch.

2. The Single if statement structure:

يتم استخدام عبارة IF للتعبير عن تعبير شرطي. إذا كان الشرط المعطى صحيحًا ، فسيتم تنفيذ الجمل مابعد الـ if وإلا فإنه سيتم تنفيذ الجمل الاختيارية.

General Form of single-selection If statement:
if (expression or condition) statementl ;



Example 1:	if (avrg >= 3.5) cout<< "good"
Example 2:	if (x > 0.0) sum += x;

Exmple 3:

```
cin>> num;
if ( num==0 )
zcount=zcount + l;
```

Example 1

Write a C++ program to read any two numbers and print the largest value of it:

```
#include<iostream.h>
void main( )
{
 Float x,y;
cout<<"Enter any two numbers \ n";
cin>>x>>y
If (x>y)
cout << "largest value is"<<x<<endl;
}
```

1. the single Block if Statement structure:

يتم تضمين عبارة block IF في ({) و (}) لتجميع التعريفات والعبارات في مكان واحدة وتعتبر هذه بمثابة بيان واحد. أو كتلة. الهيكل هو:

General Form of single block selection If statement:

```
if (expression or condition)
  {
     Statement1;
     statement2;
     statement3;
  }
```

Example2: -

Write a C++ program to read a number and check if it's positive, if it's so prints it, add it to a total, and decrement it by 2:

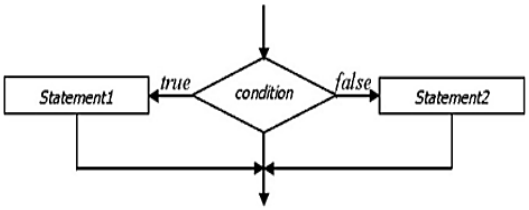
```
#include<iostream.h>
void main()
{
        int num, total=O;
        cin>>num;
        if (num>=0)
        {       cout<<num <<" is a positive";
                Total+=num;
                 num = num - 2;
        }
}
```

General Form of If/else statement:	

If(expression)
statement1;
else
statement2;
	if (expression)
	{statements;}
	else
	{statements;}

4. The If/else Statement Structure:

The IF structure is




```
If (i>j) {  If (a>b)
Temp=a;
        }
else
temp=b;
```

Example 1:

```
cin value;
if ( value >=0 )
        cout << "positive";
else
        cout << "negative";
```

Example 2:

```
cin >> numl >> num2;
if ( numl > num2 )
        cout << num 1 ;
else
        cout << num2;
```

مثال 3: اكتب برنامج بلغة C++ لقراءة درجة طالب ثم فحص هذه الدرجة اذا كانت اكبر او يساوي 50 اطبع كلمة ناجح (pass) والا اطبع كلمة راسب (fail)

Example 3

Write a C++ program to read a student degree, and check if it's degree greater than or equal to 50, then print pass, otherwise print fail:

```
#include<iostream.h>
void main()
{
        int degree;
        cin>>degree;
        if (degree>=50)
                cout<<"pass";
        else
                cout<<"fail";
        }
```

output

```
50
pass
Press any key to continue
```

```
49
fail
Press any key to continue
```

مثال 4: اكتب برنامج بلغة C++ لقراءة رقم صحيح وفحص هذا الرقم هل هو زوجي او فردي

Example 4

Write a C++ program to read a number, and check if it's even or odd:

```
#include<iostream.h>
void main( )
{
        int num; cin num;
        if ( num % 2==0)
            cout<<"even";
        else
            cout "odd";
}
```

output

```
10
even
Press any key to continue
```

```
7
odd
Press any key to continue
```

جملة else if

Else if Statements:

General Form of else if statement:

```
if ( expression or condition 1 ) statementl ;
else if ( expression or condition 2) statement2;
else if ( expression or condition 3) statement3 ;
else if ( expression or condition n ) statement-n ; else
statement-e ;
```

مثال حول عبارة else if:

Example I:

```
if ( value==0)        cout << "grade is A";
else if ( value==1 ) cout << "grade is B";
else if ( value==2 ) cout << "grade is C";
else cout << "grade is X";
```

مثال 5: اكتب برنامج بلغة C++ لقراءة رقم صحيح وطباعة اليوم من الأسبوع

Example 5

Write a C++ program to read a number, and print the day of the week.

```
#include<iostream.h>
void main( )
{
int day;
cin>> day;
if (day==1)          cout << "Sunday";
else if (day== 2)    cout<< "Monday";
else if (day==3)     cout<< "Tuesday";
else if(day==4)      cout<<"Wednesday";
else if (day==5)     cout<< "Thursday";
else if (day==6)     cout<<"Friday";
else if (day==7)     cout << "Saturday";
else cout<<"Invalid day number";
}
```

مثال 6:

Example 6

Write C++ program to compute the value of Z according to the following equations:

$$Z = \begin{cases} x+5 & ; x<0 \\ \cos(x)+4 & ; x=0 \\ \sqrt{x} & ; x>0 \end{cases}$$

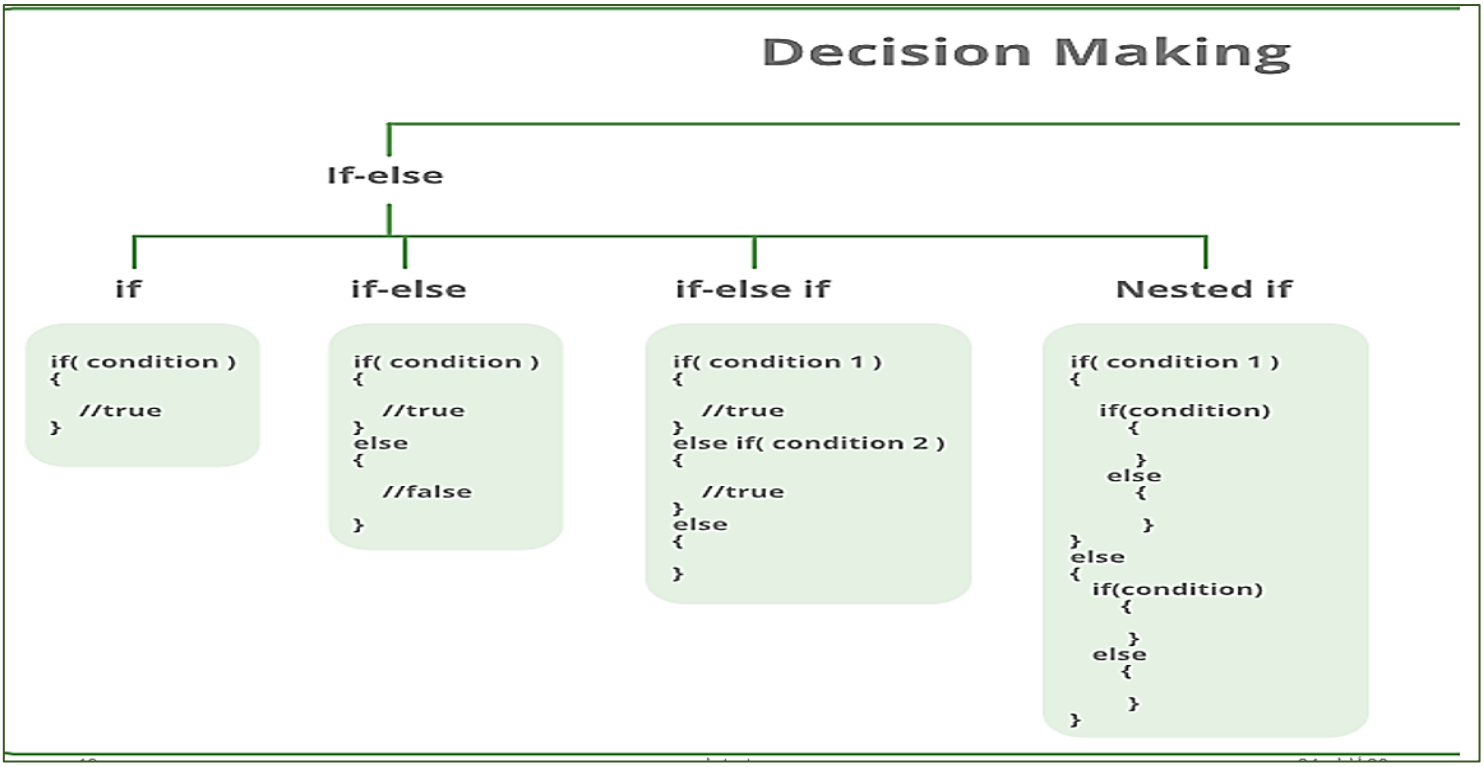
```
#include<iostream.h>
#include<math.h>
void main()
{
      int z,x;
      cout<<"Enter X value\n";
      cin>>x;
      if(x<0)
            z=x+5;
      else if(x==0)
            z=cos(x)+4;
      else
            z=sqrt(x);
      cout<<"Z is \t"<<z<<endl;
}
```

output

```
Enter X value
-1
z is  4
Press any key to continue
```

```
Enter X value
0
z is  5
Press any key to continue
```

```
Enter X value
10
z is  3
Press any key to continue
```



Nested if statement — عبارة IF المتداخلة

- بعض من عينات عبارة الـ if-else المتداخلة الموضحة أدناه

form (1)	form (2)	form (3)

if (exp.)
{statements}
else
{statements}
 	
ملاحظة:
1- في حالة تنفيذ جملة واحدة سواء كانت مابعد ال if او ال else لايحتاج الى وجود علامتي { }.
2- اما اذا كان اكثر من جملة يجب وضع علامتي الـ { }.
	if (exp.)
	{
	if (exp.)
	{statements}
	else
	{statements}
	}
	else
	{statements}
		if (exp.)
		{
		if (exp.)
		{statements}
		else
		{statements}
		}
		else
		{
		if (exp.)
		{statements}
		else
		{statements}
		}

Example (form 1)

```
int num;
cin>>num;
if(num%2==0)
    cout<<"the num is even= "<<num<<endl;
else
    cout<<"the num is odd = "<<num<<endl;
```

```
5
the num is odd=  5
Press any key to continue
```

Example (form 2)

```
int a,b,z;
cin>>a>>b>>z;
if(a>b)
{
   if(a>z)
       cout<<"the a is largest= "<<a<<endl;
   else
       cout<<"the num is z = "<<z<<endl;
}
else
    if(b>z)
        cout<<"the b is largest="<<b<<endl;
    else
        cout<<"the z is largest="<<z<<endl;
```

```
6
8
10
the z is largest= 10
Press any key to continue
```

```
10
6
8
the a is largest= 10
Press any key to continue
```

```
6
10
8
the b is largest= 10
Press any key to continue
```

Example (form 3)

```
int a,b,c,av;
cin>>a>>b>>c;
av=(a+b+c)/3;
if((av>=50)&&(av<=59))
    cout<<"pass\n";
else  if((av>=60)&&(av<=69))
    cout<<"medium\n";
      else  if((av>=70)&&(av<=79))
          cout<<"Good\n";
            else if((av>=80)&&(av>=89))
                cout<<"Very Good\n";
                 else if((av>=90)&&(av<=100))
                     cout<<"Exellent\n";
                      else  cout<<"failed\n";
```

```
60
66
77
medium
Press any key to continue
```

Conditional Statement: الجمل الشرطية

الصيغة العامة (condition ? True : False);

Example 1:

```
#include<iostream.h>
void main()
{
int value;
cin>>value;
cout<<(value>=0? "positive" : "negative")<<endl;
}
```

```
5
positive
Press any key to continue
```

```
-6
negative
Press any key to continue
```

Example 2

```
#include<iostream.h>
void main()
{
int x,y;
cin>>x>>y;
cout<<(x<y ? -1:(x==y ? 0:1));
}
```

```
4
7
-1
Press any key to continue
```

```
5
5
0
Press any key to continue
```

```
7
4
1
Press any key to continue
```

Example4

Write C++ program to read integer number, and print if its even or odd:

اكتب برنامج بلغة C ++ لقراءة عدد صحيح ، واطبع إذا كان زوجيًا أو فرديًا:

```
#include<iostream.h>
void main()
{
int value;
cout<<"Enter the number\n";
cin>>value;
cout<<(value%2==0 ? "even" : "odd")<<endl;
}
```

```
Enter the number
9
odd
Press any key to continue
```

```
Enter the number
10
even
Press any key to continue
```

استخدام الايعاز(جملة) switch – case

جملة الـ switch – case تعتبر من الجمل الشرطية

switch – case :-

هو صانع قرار خاص متعدد الاتجاهات يختبر ما إذا كان التعبير يطابق أحد عدد القيم الثابتة والأقواس وفقًا لذلك

الصيغة العامة :-

```
Switch (selector)
{
      case label1 : statement1; break;

      case label2 : statement2; break;

      case label3 : statement3; break;

        :

     case label-n : statement-n; break;

     default  :   statement-e; break;

}
```

مثال حول جملة الـ switch - case

```
#include<iostream.h>
void main()
{
int value;
cin>>value;
switch(value)
{
    case 0: cout<<"grade is A"; break;
    case 1: cout<<"grade is B"; break;
    case 2: cout<<"grade is C"; break;
    default: cout<<"grade is X"; break;
}
}
```

```
0
grade is A
Press any key to continue
```

```
1
grade is B
Press any key to continue
```

```
3
grade is X
Press any key to continue
```

Example 1

Write C++ program to read integer number, and print the number of the day in a week:

اكتب برنامج بلغة C ++ لقراءة عدد صحيح ، وطباعة رقم اليوم في الأسبوع

```
#include<iostream.h>
void main()
{
int day;
cin>>day;
switch(day)
{
      case 1: cout<<"Sunday\n"; break;
      case 2: cout<<"Monday\n"; break;
      case 3: cout<<"Tuesday\n"; break;
      case 4: cout<<"Wednesday\n"; break;
      case 5: cout<<"Thursday\n"; break;
      case 6: cout<<"Friday\n"; break;
      case 7: cout<<"Saturday\n"; break;
      default:cout<<"Invalide day number\n";break;
}
}
```

```
1
Sunday
Press any key to continue
```

```
2
Monday
Press any key to continue
```

```
5
Thursday
Press any key to continue
```

```
9
Invalide day number
Press any key to continue
```

Example 2: Write C++ program to read two integer numbers, and to perform on these numbers :

اكتب برنامج بلغة C ++ لقراءة رقمين صحيحين ، ولانجاز (العمليات الحسابية) على هذه الأرقام:

```
#include<iostream.h>
void main()
{
int a,b;
char x;
  cout<<"Enter two numbers\n";
cin>>a>>b>>x;
switch(x)
{
     case'+': cout<<a+b<<endl; break;
     case'-': cout<<a-b<<endl; break;
     case'*': cout<<a*b<<endl; break;
     case'/': cout<<a/b<<endl; break;
     default:break;
}
}
```

```
Enter two numbers
6
9
+
15
Press any key to continue

Enter two numbers
100
200
*
20000
Press any key to continue

Enter two numbers
40
30
%
Press any key to continue
```

Nested Switch Selecation Statement: **تحديد جملة الـ Switch المتداخلة**

- الصيغة العامة :-

```
Switch (selector 1)
{
    case label1 :  statement1; break;
    case label2 :  statement2; break;
    case label3 :  switch(selector 2)
                   {
                      case label1 :  statement1; break;
                      case label2 :  statement2; break;
                        :
                   }
    case label-n :  statement-n; break;
     default : statement-e; break;
}
```

Example 3 :Write C++ program to read integer number, and print the name of the computerized department:

اكتب برنامج بلغة ++ C لقراءة عدد صحيح ، وطباعة اسم القسم الحاسوب وفروعه:

```
#include<iostream.h>
void main()
{
  int a,b;
      cout<<"Enter the number of department "<<endl;
       cin>>a;
       switch(a)
       {
          case 1:cout<<"Software Engineering Department"<<endl;break;
          case 2:cout<<"Control and Computer Department"<<endl;break;
          case 3:cout<<"Computer Sciences Department"<<endl;
              cout<<"Enter the no. of Branch ";
              cin>>b;
              switch(b)
              {
                case 1:cout<<"Software"<<endl;break;
                case 2:cout<<"Information System"<<endl;break;
                case 3:cout<<"Security"<<endl;
                case 4:cout<<"AI"<<endl;
              } break;
       default :cout<<"invaild the number"<<endl;break;
       }
}
```

```
Enter the number for department name
1
Software Engineering Department
Press any key to continue
```

```
Enter the number for department name
2
Control and computers Department
Press any key to continue
```

```
Enter the number for department name
3
Computer sciences Depatment
Enter the no. of branch  4
AI
Press any key to continue
```

```
Enter the number of department
4
invaild the number
Press any key to continue
```

4. **الأسئلة البعدية في نهاية الفصل:**

- ❖ اكتب برنامج بلغة C++ لمعرفة هل العدد المدخل هو زوجي ام فردي
- ❖ اكتب برنامج بلغة C++ لطباعة ارقام ايام الأسبوع

- **المصادر الاساسية :**

1cpp.pdf (uotechnology.edu.iq)

م.م/ بشارسعدون مهدي, البرمجة المهيكلة c++ , الجامعة التكنولوجية قسم علوم الحاسوب

- **المصادر المقترحة:** لايوجد

- **روابط مقترحة ذات صلة:** مجموعة فيديوهات للمحاضرات المشروحة في هذا الكتاب.

https://www.youtube.com/playlist?list=PLnI1fRHezj5F8y024ulIKxV-DqdIyR4E6

المحتويات (الفصل الثالث)

رقم المحاضرة: 7,8,9,10,11,12	
عنوان المحاضرة:	جمل الدوران Loop statements
اســـم المدرس:	م.م/مروة نجم عبد
الفئة المستهدفة :	طلبة المرحلة الاولى
الهدف العام من المحاضرة :	تمكين الطلاب من كتابة برامج تتضمن منطقًا تفاعليًا ، حيث يمكن للبرنامج اتخاذ قرارات وتنفيذ عمليات متكررة بناءً على شروط وبيانات معينةمن خلال فهم واستخدام جمل الاختيار والتكرار.
الأهداف السلوكية او مخرجات التعلم:	• أن يشرح الطالب مفهوم جمل التكرار مثل while do while, for ,وأهميتها في تنفيذ الكود بشكل متكرر. • أن يطبق الطالب جمل الاختيار لحل مشكلات تتطلب اتخاذ قرارات داخل البرنامج. • أن يكتب الطالب برامج تتضمن جمل تكرار لتنفيذ عمليات متكررة • ان يكون الطالب قادرا على فهم دور جملة التحكم break و continue و goto للتحكم في التدفق. • تطبيق جملة break بفعالية لإنهاء الحلقات مثل for و while و-do while عند تحقق شرط معين. • التعرف على الفرق بين break وجمل التحكم الأخرى مثل continue
استراتيجيات التيسير المستخدمة	التعلم التفاعلي- العرض المرئي(الشرائح التقديمية)- التطبيق العملي- التشجيع على التعاون-تعزيز التفكير النقدي- توفير موارد إضافية(فيديوهات, مواقع تعليمية...).
المهارات المكتسبة	• مهارات التحكم في تدفق البرنامج • مهارات استخدام جمل التكرار وحل المشكلات البرمجية • مهارات البرمجة المنظمة والتصحيح والتدقيق
طرق القياس المعتمدة	الاختبارات القصيرة - المشاركة في الأنشطة الصفية- المشاريع العملية- التقييم الذاتي-المناقشات الصفية-التمارين العملية-واجبات منزلية.

1. **الاسئلة القبلية:**

- **Write program c++ to print the number from 0 to 5**
- **Write a c++ program to print the following figure:**

```
*
* *
* * *
* * * *
```

- **Write C++ program to find e from the following series:**
  e=1+(1/1!) + (1/2!) + (1/3!) + …… + (1/n!)
- **write C++ program to check if zero or negative value found**

2. **محتويات الفصل :**

- ✓ **جملة الدوران while**
- ✓ **مراجعة شاملة للمواضيع السابقة مع امثلة متنوعة**
- ✓ **جملة الدوران Do-while**
- ✓ **جملة الدوران for statement**
- ✓ **جملة الدوران المتداخلة Nested loop**
- ✓ **Breaking Control Statements جمل شرط التحكم**

3. **المحتوى العلمي:**

Loop statements جمل الدوران

تحتاج جمل الدوران الى:

- عداد (رقم تبدأ منه الدوران ورقم تنتهي اليه) لكي تنفذ الدوران
- شرط التوقف
- مقدار الزيادة

يوفر C ++ ثلاثة هياكل لجمل للتكرار هي :

1- while statement

2- do while statement

3- for statement

While Repetition Structure: الصيغة العامة لجملة الWhile

تحتاج جملة الـWhile الى شرط يحدد استمرارها او توقفها فهي ستستمر بلا توقف طالما الشرط متحقق .

```
1- while(condition)
      statement1;
2- while(condition)
{
     statement1;
     statement2;
        :
     statement-n;
}
```

Example1: print the number from 0 to 10 اطبع الأرقام من 0 الى 10

Example 1:

```
#include<iostream.h>
void main()
{
int i=0;
while(i<10)
{
  cout<<i;
   i++;
}
}
```

```
0 1 2 3 4 5 6 7 8 9
Press any key to continue
```

Example 2:

```
#include<iostream.h>
void main()
{
int w=0;
while(w<=3)
{
cout<<"value is : "<<w<<endl;
w++;
}
}
```

```
value is : 0
value is : 1
value is : 2
value is : 3
Press any key to continue
```

ملاحظة :يمكن الاستغناء عن الاقواس{ } اذا كانت الجملة التي تنفذها تتكون من سطر واحد

Example2: Printing even numbers only from 10 digits

طباعة الاعداد الزوجية فقط من بين 10 ارقام

```
i=0;
while(i<10)
{
    cout<<i;
    i+=2;
}
```

output: 0 2 4 6 8

i=10 even number

```
i=1;
while(i<10)
{
    cout<<i;
    i+=2;
}
```

output: 1 3 5 7 9

i=10 odd number

<u>Example3</u>: Write C++ program to find the summation of the following series:

sum=1 + 3 + 5 + 7 + …….+ 99

اكتب برنامج بلغة++C لايجاد مجموع السلسلة التالية

<u>Sol:</u>

```
#include<iostream.h>
void main()
{
int count=1;
int sum=0;
while(count<=99)
{
sum=sum+count;
count=count + 2;
}
cout<<"sum is :  "<<sum<<endl;
}
```

Example4 :Write C++ program to find the cube of a number, while it is positive:

اكتب برنامج بلغة لايجاد مكعب رقم عندما يكون موجب

Sol:

```
#include<iostream.h>
void main()
{
int num, cubenum;
cout<<"Enter positive number\n";
cin>>num;
while(num>0)
{
cubnum=num*num*num;
cout<<"cub number is  :  "<< cubnum <<endl;
cin>>num;
}
cout<<"error";
}
```

<u>Example5</u>: Write C++ program to find the summation of the following series.

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2$$

<u>Sol:</u>

```
#include<iostream.h>
void main()
{
int i=1, n, sum=0;
cout<<"Enter positive number\n";
cin>>n;
while(i<=n)
{
   sum+=i*i;
   i++;
}
Cout<<"sum is : " <<sum<<endl;
}
```

<u>Example6</u>: Write c++ program to find the summation of students marks and its average, assume the student have 8 marks;

<u>Sol:</u>

```
#include<iostream.h>
void main() {
int mark, i, sum=0;
float av=0; i=1;
While(i<=8)
{
cout<<"Enter mark";
cin>>mark;
sum+=mark;
i++;
}
cout<<"sum is "<<sum<<endl;
av=sum/8;
cout<<"average is "<<av<<endl;
}
```

مراجعة شاملة للمواضيع السابقة واسئلة متنوعة

سؤال تطبيقي عن استخدام جملة ال if الشرطية

Q1: write C++ program to read on integer number and check if it is positive or negative, even or odd, and write a suitable message in each case.

اكتب برنامج بلغة ++ C لقراءة عدد صحيح ومعرفة ما إذا كان موجبًا أم سالبا ، زوجيًا أم فرديًا ، واكتب رسالة مناسبة في كل حالة

```
#include<iostream.h>
void main()
{
int num;
cin>>num;
if(num>=0)
{
  cout<<"positive"<<endl;
      if(num%2==0)
          cout<<"even"<<endl;
      else
          cout<<"odd"<<endl;
}
else
{

  cout<<"negative"<<endl;
      if(num%2==0)
          cout<<"even"<<endl;
      else
          cout<<"odd"<<endl;
}
}
```

```
"D:\MSDev98\MyProjects\ii1\Debug\ii1.exe"
16
positive
even
Press any key to continue
```

```
"D:\MSDev98\MyProjects\ii1\Debug\ii1.exe"
-15
negative
odd
Press any key to continue
```

ملاحظات مهمة عن جملة الـ switch:

1. لا يجوز بعد جملة الـ case استخدام تعبيرات حسابية مثلا: case x+y
2. جملة ال switch لايجوز معها استخدام المتغيرات من النوع الحقيقي float
3. جملة الـ switch لايجوز معها استخدام المتغير من نوع string
4. لايجوز بعد جملة الـ case استخدام أدوات المقارنة مثلا case >5=50
5. لايجوز بعد جملة الـ case استخدام الفارزة المنقوطة مثلا case b;

المقارنة ما بين جملة الـ if و switch

switch	if
1- لا يمكن استخدام ارقام حقيقية 2- لايمكن استخدام أدوات المقارنة 3- يجب ان يكون قيمة المتغير قيمة ثابتة مثل 1 و 45 و 7 وهكذا 4- لا يمكن استخدام متغير من النوع string 5- لا يمكن استخدام اكثر من شرط في جملة واحدة 6- يكون اسرع خلال التنفيذ	1- يمكن استخدام ارقام حقيقية او اسم او حرف 2- ممكن استخدام أدوات المقارنة 3- ممكن استخدام قيمة ثابتة او مدى معين من الأرقام مثل 90> وهكذا 4- يمكن استخدام متغير من النوع string 5- يمكن استخدام اكثر من شرط في جملة واحدة

Q2/ Write a C++ program to read integer number and check if it is even or odd use statement switch

```
#include<iostream.h>
void main()
{
int x;
cout<<"Enter a number ";
cin>>x;
switch(x%2)
{
   case 0:cout<<"even"<<endl;break;
   default: cout<<"odd"<<endl;break;
}
}
```

```
"D:\MSDev98\MyProjects\ii1\Debug\ii1.exe"
Enter a number 7
odd
Press any key to continue
```

```
"D:\MSDev98\MyProjects\ii1\Debug\ii1.exe"
Enter a number 8
even
Press any key to continue
```

<u>ملاحظة مهمة</u>: وضع عبارة شرطية داخل جملة الـ switch – case أي ممكن وضع عبارة (if) الشرطية بمختلف أنواعها داخل الـ case حالها حال أي جملة برمجية

<u>على سبيل المثال</u>: برنامج ندخل الرقم من شاشة التنفيذ ثم يطلب من الطالب ادخال رقم العملية التي يريد ان يؤديها على هذا الرقم اذا ادخل الطالب رقم 1 سوف يبين له هل الرقم موجب ام سالب واذا ادخل رقم 2 يبين له هل الرقم زوجي ام فردي.

```
#include<iostream.h>
void main()
{
int mark,num;
cin>>mark>>num;
switch(num)
{
case 1:
{
    if(mark>=0)
     cout<<"positive"<<endl;
  else
     cout<<"negative"<<endl;
}
break;
case 2:
    {
        if(mark%2==0)
        cout<<"even"<<endl;
    else
        cout<<"odd"<<endl;
    }
    break;

}
}
```

```
"D:\MSDev98\MyProjects\p\Debug\p.exe"
7
1
positive
Press any key to continue
```

```
"D:\MSDev98\MyProjects\p\Debug\p.exe"
8
2
even
Press any key to continue
```

<u>Q3:</u> write C++ program to read on integer number and check if it is positive or negative, even or odd, and write a suitable message in each case use statement nested switch

```
#include<iostream.h>
void main()
{
int x;
cout<<"Enter a number ";
cin>>x;
switch(x%2)
{
    case 0:cout<<"even"<<endl;
        switch(x>0)
        {
        case 1:cout<<"posative"<<endl;break;
        default:cout<<"negative"<<endl;
        }break;
    default: cout<<"odd"<<endl;
        switch(x>0)
        {
        case 1:cout<<"posative"<<endl;break;
        default:cout<<"negative"<<endl;
        }
}
}
```

Q4/ Write a C++ program to read 10 integer number and find the summation of positive number only

```
#include<iostream.h>
void main()
{
    int i=1, mark , sum=0;
    while(i<=10)
    {
        cin>>mark;
        if(mark>0)
            sum+=mark;
        i++;
    }
    cout<<sum<<endl;
}
```

```
"D:\MSDev98\MyProjects\b\Debug\b.exe"
8
7
5
-9
-5
-7
10
8
7
6
51
Press any key to continue
```

Q5/ Write a C++ program to read 10 integer number and find the summation of even number only.

```
#include<iostream.h>
void main()
{
    int i=1, mark , sum=0;
    while(i<=10)
    {
        cin>>mark;
        if(mark%2==0)
            sum+=mark;
        i++;
    }
    cout<<"sum="<<sum<<endl;
}
```

```
"D:\MSDev98\MyProjects\b\Debug\b.exe"
8
2
9
7
5
1
10
5
7
9
sum=20
Press any key to continue
```

Q6: Write C++ program to print the following: 10 9 8 7 6 5 4 3 2 1

```
#include<iostream.h>
void main()
{
    int i=10;
    while(i>=1)
    {
      cout<<i;
      i--;

    }
    cout<<endl;

}
```

ناتج التنفيذ

```
"D:\MSDev98\MyProjects\b\Debug\b.exe"
10987654321
Press any key to continue
```

Q7: Write C++ program to print the following: 1 2 4 8 16 32 64

```
#include<iostream.h>
void main()
{
    int i=1;
    while(i<=64)
    {
      cout<<i<<ends;
      i*=2;
    }
    cout<<endl;

}
```

```
"D:\MSDev98\MyProjects\b\Debug\b.exe"
1 2 4 8 16 32 64
Press any key to continue
```

Q8: Write C++ program to print the following:

```
1  10
2  9
3  8
4  7
5  6
```

```
#include<iostream.h>
void main()
{
    int i=1;
    while(i<=6)
    {
      cout<<i<<ends<<11-i<<endl;
      i++;
    }
    cout<<endl;

}
```

"D:\MSDev98\MyProjects\b\Debug\b.exe"

```
1 10
2 9
3 8
4 7
5 6
6 5

Press any key to continue
```

<u>Q9:</u> Write c++ program to find x power n

```
#include<iostream.h>
void main()
{
    int x,n,po,i=1;
    cin>>x>>n;
    po=1;
    while(i<=n)
    {
        po=po*x;
        cout<<"po="<<ends<<po<<endl;
        i++;
    }
    cout<<endl;

}
```

"D:\MSDev98\MyProjects\b\Debug\b.exe"

```
2
3
po= 2
po= 4
po= 8

Press any key to continue
```

Q10: Write c++ program to find the factorial of n!=n * n-1* n-2 * n-3 * …….. * 2 * 1

6!=6 * 5 * 4 * 3 * 2 * 1

ناتج التنفيذ

```
#include<iostream.h>
void main()
{
    int f=1,n;
    cin>>n;
    while(n>1)
    {
        f=f*n;
        n--;
    }
    cout<<f<<endl;

}
```

"D:\MSDev98\MyProjects\b\Debug\b.exe"

```
4
24
Press any key to continue
```

Q11: Write C++ program to print the following: 0 1 1 2 3 5 8 13 21 ……. N

```
#include<iostream.h>        //0 1 1 2 3 5 8 13 21 ..... n
void main()                 //x y z
{
int x=0, y=1, z, sum=0, n;
cin>>n;
cout<<x<<ends<<y<<ends;
sum=x+y;
while(x+y<=n)
{
    z=x+y;
    cout<<z<<ends;
    x=y;
    y=z;
    sum+=z;
}
cout<<endl<<sum<<endl;
}
```

"D:\MSDev98\MyProjects\ii1\Debug\ii1.exe"

```
8
0 1 1 2 3 5 8
20
Press any key to continue
```

sum	x	y	z
1	0	1	1
2	1	1	2
4	1	2	3
7	2	3	5
12	3	5	8
20			

Loop statements جمل الدوران

2- for statement

الصيغة العامة لجملة الدوران for statement هي:-

1- for(initialization ; continuation condition ; update statement;)

for (القيمة البدائية ; شرط التوقف ; مقدار الزيادة)

2- for(initialization ; continuation condition ; update)

```
 {
     statement 1;
     statemenr 2;
         :
}
```

امثلة حول جملة الدوران for statement

Example 1: for(i=0 ; i<10 ; i++) output 0 1 2 3 4 5 6 7 8 9

cout<<i;

Example 2: for(i=0 ; i<10 ; i+=2) output 0 2 4 6 8

cout<<i;

Example 3: for(i=1 ; i<10 ; i+=2) output 1 3 5 7 9

cout<<i;

<u>Example 4</u>: write program c++ to print the number from 0 to 10

اكتب برنامج لطباعة الأرقام من 0 الى 10

لطباعة الأرقام من 0 الى 10

```
#include<iostream.h>
void main()
{
for(int i=0 ; i<10 ; i++)
  cout<<i;
}
```

```
0 1 2 3 4 5 6 7 8 9
Press any key to continue
```

لطباعة الأرقام من 0 الى 4

```
#include<iostream.h>
void main()
{
int i;
for(i=0 ; i<=3 ; i++)
cout<<"value is : "<<i<<endl;
}
```

```
value is : 0
value is : 1
value is : 2
value is : 3
Press any key to continue
```

ملاحظة :يمكن الاستغناء عن الاقواس{ } اذا كانت الجملة التي تنفذها تتكون من سطر واحد

<u>Example 5</u>: write C++ program to add numbers between 1 and 100

اكتب برنامج لجمع (لإضافة) الاعداد من 1 الى 100:

```
#include<iostream.h>
void main()
{
int i , sum=0;
for(i=1;i<=100;i++)
sum=sum+i;
cout<<"sum is = "<<sum;
}
```

<u>Example6</u>: Write C++ program to find the factorial of n (using for statement)

n!=n * n-1 * n-2 * n-3 * …… *2 * 1

اكتب برنامج لايجاد مفكوك n باستخدام جملة for

```
#include<iostream.h>
void main()
{
int n , f=1;
cout<<"enter positive number  ";
cin>>n;
for(int i=1 ; i<=n ; i++)          // or for(i=n; i>=1 ; i--)
f=f*i;
cout<<"factorial is: "<<f;   }
```

Example7: Write C++ program to the result of the following :

$$\sum_{i=1}^{20} i^2$$

```
#include<iostream.h>
void main()
{
int i , sum=0;
for(i=1 ; i<=20 ; i++)
sum=sum+(i * i);
cout<<"the sum is:  "<<sum<<endl;}
```

<u>Example8</u>: Write C++ program to read 10 integer number, and find the sum of positive number only:

اكتب برنامج لقراءة 10 ارقام صحيحة وايجاد مجموع الارقام الموجبة فقط

```
#include<iostream.h>
void main()
{
int num, sum=0;
for(int i=1 ; i<=10 ; i++)
{
   cout<<"enter your number: ";
   cin>>num;
   if(num>0)
   sum=sum+num;
}
cout<<"the sum is: "<<sum;
}
```

<u>Example9</u>: Write C++ program to print the following series:

1 , 2 , 4 , 8 , 16 , 32 , 64

```
#include<iostream.h>
void main()
{
int i;
for(i=1 ; i<=64 ; i*=2)
  cout<<i <<"  ";
  cout<<endl;
}
```

```
1
1*2=2
2*2=4
4*2=8
8*2=16
16*2=32
32*2=64
```

Example10: Write c++ program to print the following

```
1   10
2   9
3   8
4   7
5   6
6   5
```

```
#include<iostream.h>
void main() {
{
int x;
for(x=1 ; x<7 ; ++x)
   cout<<x<<"\t"<<11-x<<endl;
}
```

Nested loops

يمكننا وضع الحلقات واحدة داخل الأخرى لحل مشاكل برمجية معينة . قد تتداخل الحلقات على النحو التالي

- loop1
  - loop2
- loop1
  - loop2
  - loop3
- loop1
  - loop2
    - Loop3
-

Example11: Write a C++ program to evaluate the following series: $\sum_{i=1}^{5} \sum_{j=1}^{10} i+2j$

12345678910 12345678910 12345678910 12345678910 12345678910

```
#include<iostream.h>
void main() {
int i,j,sum=0;
for(i=1;i<=5;i++)
for(j=1;j<=10;j++)
sum=sum+(i+2*j);
cout<<"sum is: "<<sum; }
```

Example12: Write a c++ program to print the following figure:

```
*
* *
* * *
* * * *
```

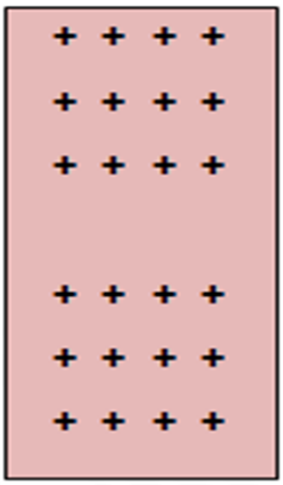
```
#include<iostream.h>
void main()
{
int i,j;
for(i=1;i<=4;i++)
{
   for(j=1;j<=i;j++)
      cout<<"*";
       cout<<"\n";}}
```

Example13: Write a C++ program to read a line using for loop

```
#include<iostream.h>
void main()
{
cout<<"Explaining the nested for loop\n";
for(int i=0;i<=2;i++){
cout<<i;
for(int j=0;j<=2;j++) {
cout<<"computer sciences department\n";
}}}
```

<u>Example14</u>: What is the output of the following C++ program ?

```
#include<iostream.h>
void  ()
{
int i,j,k;
for(i=1;i<=2;i++) //للجزء الأول والثاني
{
   for(j=1;j<=3;j++) // للاسطر
   {
      for(k=1;k<=4;k++) // للاعمدة//
       cout<<“  * ”;
       cout<<“\n”;}
       cout<<“\n”;
}}
```



مثال يوضح عمل الـ for الخارجية والـ for الداخلية حيث المتغير i يمثل الـ for الخارجية والمتغير j يمثل الـ for الداخلية:

```
#include<iostream.h>
void main()
{
    int i,j;
    for(i=1;i<=10;i++)
    {
        cout<<"i= "<<i<<"   "<<ends;
        for(j=1;j<=4;j++)
            cout<<j<<ends;
            cout<<endl;
    }
}
```

```
i= 1   1 2 3 4
i= 2   1 2 3 4
i= 3   1 2 3 4
i= 4   1 2 3 4
i= 5   1 2 3 4
i= 6   1 2 3 4
i= 7   1 2 3 4
i= 8   1 2 3 4
i= 9   1 2 3 4
i= 10   1 2 3 4
Press any key to continue
```

مثال يوضح عمل الـ for الخارجية والتي تضم فورين كل for مستقلة عن الاخرى حيث المتغير i يمثل الـ for الخارجية والتي تحتوي على 2 for المستقلة ويمثل بالمتغير j والمتغير k

```
#include<iostream.h>
void main()
{
    int i,j,k;
    for(i=1;i<=10;i++)
    {
        cout<<i<<endl;
    for(j=1;j<=10;j++)
        cout<<j;
    cout<<endl;
        for(k=1;k<=4;++k)
            cout<<k;
        cout<<endl;
    }
}
```

```
1
12345678910
1234
2
12345678910
1234
3
12345678910
1234
4
12345678910
1234
5
12345678910
1234
6
12345678910
1234
7
12345678910
1234
8
12345678910
1234
9
12345678910
1234
10
12345678910
1234
```

مثال يوضح عمل الـ for المتداخلة:

```
#include<iostream.h>
void main()
{
    int i,j,k;
    for(i=1;i<=10;i++)
    {
        cout<<"i= "<<i<<"  "<<ends;
    for(j=1;j<=4;j++)
    {
        cout<<"j= "<<j<<ends;
        for(k=1;k<=4;++k)
            cout<<k<<ends;
    }
    cout<<endl;
    }
}
```

```
i= 1   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 2   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 3   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 4   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 5   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 6   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 7   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 8   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 9   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
i= 10   j= 1 1 2 3 4 j= 2 1 2 3 4 j= 3 1 2 3 4 j= 4 1 2 3 4
Press any key to continue
```

Example1: Write a c++ program to print the following figure:

```
*********
********
*******
*****
****
**
*
```

```
#include<iostream.h>
void main()
{
    int i,j;
    for(i=1;i<=10;i++)
    {
        for(j=10;j>=i;j--)
            cout<<"*";
        cout<<endl;
    }
}
```

```
**********
*********
********
*******
******
*****
****
***
**
*
Press any key to continue
```

<u>Example2</u>: Write a c++ program to print the following figure:

```
1
333
55555
7777777
999999999
7777777
55555
333
1
```

```cpp
#include<iostream.h>
void main()
{
        int i,j,k,x=4,y=1;
        for(i=1;i<10;i++)
        {
                for(j=1;j<=x;j++)
                        cout<<"  ";
                for(k=1;k<=y;k++)
                        cout<<y;
                cout<<endl;
                if(i<5)
                {
                        x--;
                        y=y+2;
                }
                else
                {
                        x++;
                        y=y-2;
                }
        }
}
```

```
    1
   333
  55555
 7777777
999999999
 7777777
  55555
   333
    1
Press any key to continue
```

<u>Example3</u> :Write C++ program to find e from the following series:
e=1+(1/1!) + (1/2!) + (1/3!) + …… + (1/n!)

```cpp
#include<iostream.h>
void main()
{
    int i,k,n,x;
    float e=1.0,f=1;
    cout<<"Enter the number of terms: ";
    cin>>n;
    for(i=1;i<=n;i++)
    {
        f=1;
        for(k=1;k<=i;k++)
            f=f*k;
        e=e+(1/f);
    }
    cout<<"e= "<<e<<endl;
}
```

```
Enter the number of terms: 4
e= 2.70833
Press any key to continue
```

<u>Example4</u> :Write C++ program to find e from the following series:
Z=x1/1! + x2/2! + x3/3! + x4/4! + …… + xn/n!

```
#include<iostream.h>
void main()
{
    int i,j,k,x,n,p=1,f=1;
    float z=0;
    cout<<"Enter the number of terms: ";
    cin>>n;
    cout<<"Enter the value of x: ";
    cin>>x;
    for(i=1;i<=n;i+=2)
    {
        p=1;
        for(j=1;j<=i;j++)
            p=p*x;
            cout<<"p= "<<p<<endl;
        f=1;
        for(k=2;k<=(i+1);k++)
            f=f*k;
            cout<<"f= "<<f<<endl;
            z=z + (p/f);
            cout<<"z= "<<z<<endl;
    }
}
```

```
Enter the number of terms: 5
Enter the value of x: 2
p= 2
f= 2
z= 1
p= 8
f= 24
z= 1
p= 32
f= 720
z= 1
Press any key to continue
```

<u>Example5</u> :Write C++ program to find e from the following series:
Z=x1/2! + x3/4! + x5/6! + x7/8! + …… + xn/m!

```
#include<iostream.h>
void main()
{
    int i,j,k,x,n,p=1,f=1;
    float z=0;
    cout<<"Enter the number of terms: ";
    cin>>n;
    cout<<"Enter the value of x: ";
    cin>>x;
    for(i=1;i<=n;i++)
    {
        p=1;
        for(j=1;j<=i;j++)
            p=p*x;
            cout<<"p= "<<p<<endl;
        f=1;
        for(k=1;k<=i;k++)
            f=f*k;
            cout<<"f= "<<f<<endl;
            z=z+(p/f);
            cout<<"z= "<<z<<endl;
    }
}
```

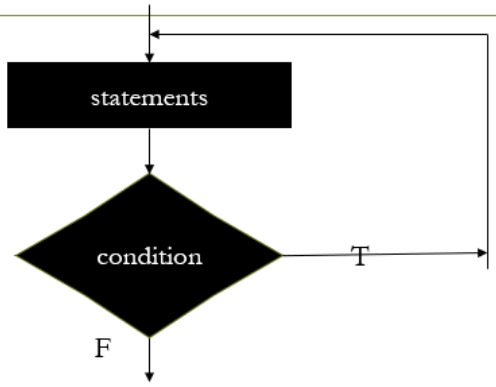
```
Enter the number of terms: 3
Enter the value of x: 2
p= 2
f= 1
z= 2
p= 4
f= 2
z= 4
p= 8
f= 6
z= 5.33333
Press any key to continue
```

Do/while statement

الصيغة العامة لجملة الـ do/while

```
do
statement;
while(condition);
```

```
do
{
statement1;
statement2;
:
statement-n;
}
while(condition);
```



Example1:

```
i=0;
do
    {
      cout<<i;
       i++;
    }
  while(i<10);
```

output

0 1 2 3 4 5 6 7 8 9

Example2: print the even number only

```
i=0;
do
  {
    cout<<i;
    i+=2;
  }
while(i<10);
```

output: 0 2 4 6 8

Example 3:- write c++ program to valid input checking that accept the numbers between 50 ..70 only

اكتب برنامج بلغة ++c لفحص الإدخال الصحيح الذي يقبل ما بين 50 .... 70 فقط

```
#include<iostream.h>
void main()
{
    int accept=1;
    int x,low=50,high=70;
    do
    {
        cout<<"enter number: ";
        cin>>x;
        if(x>=low && x<=high)
            accept=1;
        else
            accept=0;
    }
    while(!accept);
}
```

while (accept == 1) or
while (accept != 0)

```
enter number: 30
enter number: 44
enter number: 77
enter number: 88
enter number: 90
enter number: 66
Press any key to continue
```

Example 4:- write c++ program to find the summation of students marks, and its average, assume the student have 8 marks:

```
#include<iostream.h>
void main()
{
    int mark,i,sum=0;
    float av=0;
    i=1;
    do
    {
        cout<<"enter mark: ";
        cin>>mark;
        sum=sum+mark;
        i++;
    }
    while(i<=8);
    cout<<"sum is: "<<sum<<endl;
    av=sum/8;
    cout<<"avearge is: "<<av<<endl;

}
```

```
"D:\MSDev98\MyProjects\mmmm
enter mark: 70
enter mark: 60
enter mark: 77
enter mark: 88
enter mark: 78
enter mark: 87
enter mark: 76
enter mark: 74
sum is: 610
avearge is: 76
Press any key to continue
```

Example 5:- write c++ program to find the factorial of n :

n! = n * n-1 * n-2 * n-3 * …… * 2 * 1

```
#include<iostream.h>
void main()
{
    int n , f=1;
    cout<<"enter positive number: ";
    cin>>n;
    do
    {
        f=f*n;
        n--;
    }
    while(n>1);
    cout<<"factorial is: "<<f<<endl;
}
```

```
"D:\MSDev98\MyProjects\mmmmm
enter positive number: 6
factorial is: 720
Press any key to continue
```

Example 6:- write c++ program to find the summation of even numbers

```
#include<iostream.h>
void main()
{
    int num=2,sum=0,max;
    cout<<"enter a number: ";
    cin>>max;
    do
    {
        sum=sum+num;
        num+=2;
    }
    while(num<=max);
    cout<<"max is: "<<max<<endl;
    cout<<"sum="<<sum<<endl;
}
```

```
"D:\MSDev98\MyProjects\mmmm
enter a number: 10
max is: 10
sum=30
Press any key to continue
```

Example 7:- write c++ program to compute the sum of even number from some numbers until (“0”)

اكتب برنامج لايجاد مجموع الاعداد الزوجية طالما الرقم لايساوي صفر أي عند الرقم صفر يتوقف

```
#include<iostream.h>
void main()
{
    int x,sum=0;
    do
    {
        cin>>x;
        if(x%2==0)
            sum=sum+x;
    }
    while(x!=0);
        cout<<"sum= "<<sum<<endl;
}
```

```
7
8
2
10
5
7
9
6
3
4
20
0
sum= 50
Press any key to continue
```

Example 8:- write c++ program to compute the sum of the integer numbers that divisible by 4 from 10 different integer number. Using (do/while)

```
#include<iostream.h>
void main()
{
    int x,sum=0,i=1;
    do
    {
        cin>>x;
        if(x%4==0)
            sum+=x;
        i++;
    }
    while(i<=10);
        cout<<"sum= "<<sum<<endl;
}
```

```
4
8
12
16
9
3
2
20
40
34
sum= 100
Press any key to continue
```

Breaking Control Statements:

جمل شرط التحكم

من أجل المعالجة الفعالة لعبارات الحلقة ، C++ يسمح باستخدام الأنواع التالية من عبارات التحكم:

(a) Break control statement: جملة شرط التوقف

استخدام جملة break لإنهاء التحكم او الخروج من جملة الحلقة التكرارية. عادةً ما يتم استخدام جملة الـ break في جملة الايعاز البرمجي switch-case وفي كل حالة شرط يجب استخدام جملة الـ break . إذا لم يكن الأمر كذلك ، فسيتم نقل التحكم إلى الجملة اللاحقة أيضًا. الصيغة العامة لجملة الـ break هي :

break;

```
#include<iostream.h>
void main()
{
        int i;
        for(i=1;i<=100;i++)
        {
                cout<<i<<"  ";
        if(i==10) break;
        }
        cout<<endl;
}
```

```
1 2 3 4 5 6 7 8 9 10
Press any key to continue
```

Example 2:

```
#include<iostream.h>
void main()
{
    int i,j;
    for(i=1;i<=10;i++)
        for(j=1;j<=20;j++)
    {
        cout<<i<<" "<<j<<"  "<<i*j<<endl;
        if(i==j) break;
    }
    cout<<endl;
}
```

```
1 1  1
2 1  2
2 2  4
3 1  3
3 2  6
3 3  9
4 1  4
4 2  8
4 3  12
4 4  16
5 1  5
5 2  10
5 3  15
5 4  20
5 5  25
6 1  6
6 2  12
6 3  18
6 4  24
6 5  30
6 6  36
7 1  7
7 2  14
7 3  21
7 4  28
7 5  35
7 6  42
7 7  49
8 1  8
8 2  16
8 1  8
8 2  16
8 3  24
8 4  32
8 5  40
8 6  48
8 7  56
8 8  64
9 1  9
9 2  18
9 3  27
9 4  36
9 5  45
9 6  54
9 7  63
9 8  72
9 9  81
10 1  10
10 2  20
10 3  30
10 4  40
10 5  50
10 6  60
10 7  70
10 8  80
10 9  90
10 10  100
```

Example 3:- write C++ program to check if zero or negative value found:

اكتب برنامج C ++ للتحقق مما إذا تم العثور على قيمة صفرية أو قيمة سالبة

```
#include<iostream.h>
void main()
{
    int value,i=0;
    while(i<=10)
    {
        cout<<"enter a number \n";
        cin>>value;
        if(value<=0)
        {
            cout<<"Zero or negative value found\n";
            break;
        }
        i++;
    }
}
```

```
enter a number
5
enter a number
8
enter a number
0
Zero or negative value found
Press any key to continue
```

(b) Continue control statement جملة شرط الاستمرار

يتم استخدام continue لتكرار نفس العملية مرة أخرى حتى يتحقق من الخطأ. تركيبته العامة هي:

continue;

```
#include<iostream.h>
void main()
{
    int value;
    cout<<"enter a number \n";
    cin>>value;
    while(value<=100)
    {

      if(value<=0)
      {
      cout<<"Zero or negative value found\n";
      continue;
      }
    cin>>value;
    }
}
```

```
enter a number
5
8
0
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
Zero or negative value found
```

Example2

```
#include<iostream.h>
void main()
{
      int x,n;
      do
      {
      cin>>x;
      cin>>n;
      if(n<1)
         continue;
      cout<<x<<endl;
      --n;
      }
      while(n<1);
}
```

```
5
-5
9
-9
3
-3
6
-5
8
2
8
Press any key to continue
```

Example 2:

```
#include<iostream.h>
void main()
{
start:
cout<<"welcom\n";
goto start;
cout<<"Hellow\n";
cout<<"Hi";
}
```

```
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
```

Goto Statement

هي جملة الدوران او نوع من جمل الـ loop لكن دوران غير منتظم

الصيغة العامة : **goto label;**

Example 1

```
#include<iostream.h>
void main()
{
start:
cout<<"****"<<endl;
goto start;
}
```

- طريقة استخدام الـ goto
- انشاء عنوان (كتابة العنوان حسب رغبة المستخدم)
- كتابة المقطع البرمجي
- كتابة goto ← العنوان

ملاحظة: لا يجوز كتابة عنوان الـ goto قبل تعريف المتغير

Example 2:

```
#include<iostream.h>
void main()
{
start:
cout<<"welcom\n";
goto start;
cout<<"Hellow\n";
cout<<"Hi";
}
```

```
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
welcom
```

Example 3: Write C++ program to check if zero or negative value found:

```
#include<iostream.h>
void main()
{
int value,i=0;
while(i<=10)
{
    cout<<"enter number ";
    cin>>value;
    cout<<"zero or nrgative found\n";
    goto error;
    i++;
}
error:
if(value>0)
cout<<"input data error\n";
}
```

```
enter number 9
zero or nrgative found
input data error
Press any key to continue
```

```
enter number -9
zero or nrgative found
Press any key to continue
```

Example 4

```
#include<iostream.h>
void main()
{
    int x=0;
start:
cout<<++x<<endl;
if(x<=10) goto start;
}
```

```
1
2
3
4
5
6
7
8
9
10
11
Press any key to continue
```

Example 5:

```
#include<iostream.h>
void main()
{
    int x=10;
start:
cout<<--x<<endl;
if(x>0) goto start;
}
```

```
9
8
7
6
5
4
3
2
1
0
Press any key to continue
```

4. **الأسئلة البعدية في نهاية الفصل:**

- **Write program c++ to print the number from 5 to 0**
- **Write a c++ program to print the following figure:**

```
****
* **
* *
*
```

- **Write C++ program to find e from the following series:**
  e=2+(1/1!) + (1/2!) + (1/3!) + …… + (1/n!)
- **write C++ program to check if zero or posetive value found**

- **المصادر الاساسية :**

1cpp.pdf (uotechnology.edu.iq)

م.م/ بشارسعدون مهدي, البرمجة المهيكلة ++c , الجامعة التكنولوجية قسم علوم الحاسوب

- **المصادر المقترحة:** لايوجد
- **روابط مقترحة ذات صلة:** مجموعة فيديوهات للمحاضرات المشروحة في هذا الكتاب.

https://www.youtube.com/playlist?list=PLnI1fRHezj5F8y024ulIKxV-DqdIyR4E6

المحتويات (الفصل الرابع)

رقم المحاضرة 13,14,15	
عنوان المحاضرة:	المصفوفات Arrays والدوال functions
اســـم المدرس:	م.م/مروة نجم عبد
الفئة المستهدفة :	طلبة المرحلة الاولى
الهدف العام من المحاضرة :	❖ تمكين الطالب من فهم كيفية تعريف المصفوفة في.C++ ❖ بإمكان الطالب التعرف على أنواع المصفوفات (مصفوفة أحادية البُعد، ثنائية البُعد، ❖ ان يتمكن الطالب من التفاعل مع المصفوفات من خلال عمليات الوصول والتعديل ❖ ان يتمكن الطالب من تعريف الدوال، تمرير المعاملات إليها، واسترجاع القيم من خلالها. ❖ ان يتمكن الطالب من معرفة الفرق بين الدوال التي ترجع قيمة وتلك التي لا ترجع قيمة) الدوال.(void
الأهداف السلوكية او مخرجات التعلم:	❖ فهم طريقة تعريف المصفوفات والدوال في.C++ ❖ القدرة على كتابة برامج تتضمن عمليات على المصفوفات. ❖ استخدام الدوال لتقسيم البرنامج إلى وحدات صغيرة تسهل الفهم والتطوير. ❖ تطبيق المفاهيم لتطوير برامج أكثر تنظيمًا وكفاءة.
استراتيجيات التيسير المستخدمة	التعلم التفاعلي- العرض المرئي(الشرائح التقديمية)- التطبيق العملي- التشجيع على التعاون-تعزيز التفكير النقدي- توفير موارد إضافية(فيديوهات, مواقع تعليمية...).
المهارات المكتسبة	• القدرة على تعريف وإنشاء مصفوفات من أنواع بيانات مختلفة في C++ • التمكن من الوصول إلى العناصر داخل المصفوفة باستخدام الفهارس • استخدام الحلقات مثل for وwhileللتكرار عبر المصفوفات. • القدرة على تعريف دالة بأسمائها ومعاملاتها والقيم التي ترجعها. • تعلم كيفية تمرير المعاملات إلى الدوال بالمرجعية أو بالقيمة. • فهم كيفية إنشاء واستخدام الدوال من النوع voidالتي لا ترجع قيمة. • تعلم كيفية إنشاء دوال ترجع قيمة وتحقيق الاستفادة منها في البرمجيات.
طرق القياس المعتمدة	الاختبارات القصيرة - المشاركة في الأنشطة الصفية- المشاريع العملية- التقييم الذاتي-المناقشات الصفية-التمارين العملية-واجبات منزلية.

1. الاسئلة القبلية:

- Write C++ program to read (4*4) 2D array, then find the summation of the array elements
- Write C++ program to read 3 numbers and print it in reverse order
- Write C++ program to convert 2D-array into 1D-array.
- write a program using a function to calculate the average of two numbers entered by the user in the main program

اكتب برنامج باستخدام دالة لحساب معدل رقمين أدخلهما المستخدم في البرنامج الرئيسي

2. محتويات الفصل :

- تعريف المصفوفة في لغة c++ وفوائدها
- المصفوفة الأحادية
- One Dimensional Array
- الوصول لعناصر المصفوفة في C++
- طريقة وضع قيم أولية لعناصر المصفوفة في C++
- التعامل مع المصفوفة بواسطة حلقة التكرار في C++
- المصفوفات الثنائية
- Two Dimensional array
- أنواع الدوال
- امثلة وتطبيق عملي

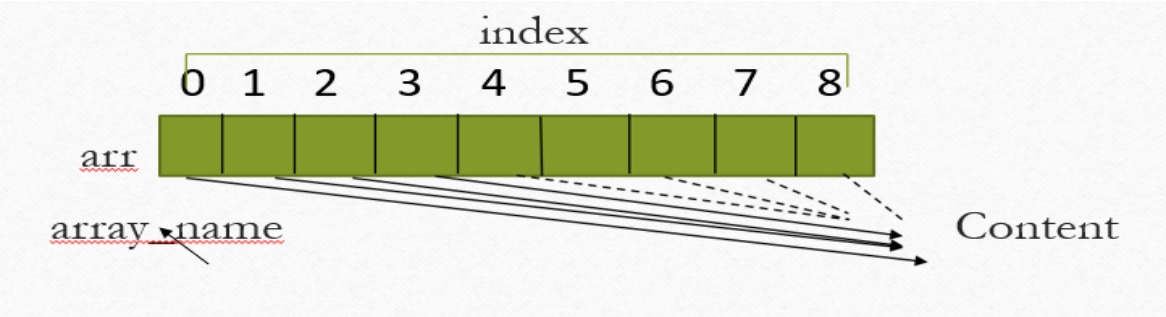
3. المحتوى العلمي

مفهوم المصفوفات في البرمجة C++

- المصفوفة (Array) عبارة عن متغير واحد يتألف من عدة عناصر (Elements) من نفس النوع.

  و كل عنصر في المصفوفة يمكن تخزين قيمة واحدة فيه.
- عناصر المصفوفة تتميز عن بعضها من خلال رقم محدد يعطى لكل عنصر يسمى index.

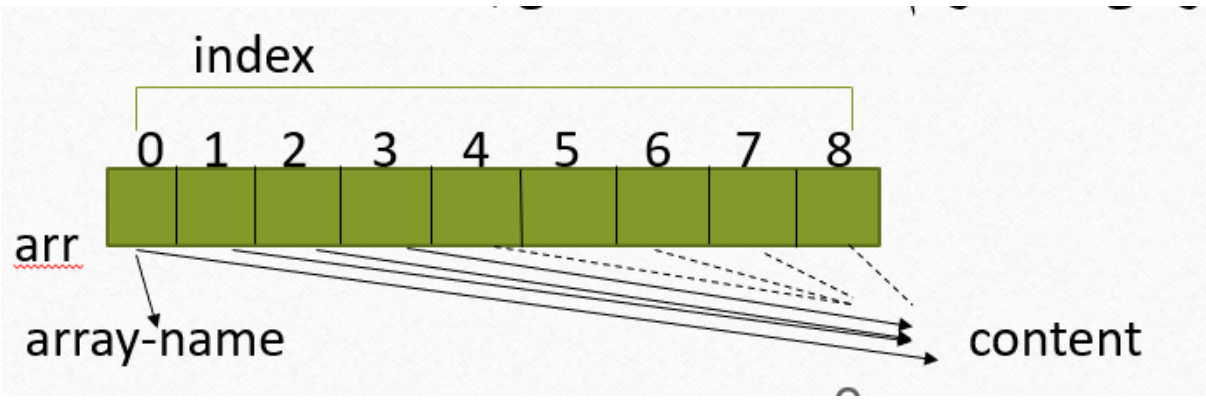
  أول عنصر في المصفوفة دائماً يكون رقمه 0.
- ملاحظة: أن عدد عناصر المصفوفة ثابت, أي بمجرد تحديد عناصر المصفوفة لا يمكن تغييره من جديد, علما اننا نستطيع تغيير قيم هذه العناصر متى شئنا.



فوائد المصفوفات

- تقليل عدد المتغيرات المتشابهة, فمثلاً إذا كنا نريد تعريف 10 متغيرات نوعهم int نقوم بتعريف مصفوفة واحدة تتألف من 10 عناصرمن نوع int.
- التعامل مع الكود يصبح أسهل, لأنك إذا قمت بتخزين المعلومات داخل مصفوفة, تستطيع تعديلهم, مقارنتهم أو جلبهم كلهم دفعة واحدة بكود صغير جداً باستخدام حلقة.

  تستطيع الوصول لأي عنصر من خلال رقم الـ index الخاص به.



المصفوفة فى C++

هناك ثلاث طرق يمكننا اتباعها لتعريف مصفوفة (Declare Array) جديدة سنتعرف عليها تباعاً:

- **الأسلوب الأول: يستخدم لتعريف مصفوفة مع تحديد عدد عناصرها**

```
datatype arrayName[size];
```

- **الأسلوب الثاني يستخدم لتعريف مصفوفة مع تحديد قيمها الأولية**

```
datatype arrayName[] = {value1, value2, ..};
```

- **الأسلوب الثالث يستخدم لتعريف مصفوفة مع تحديد عدد عناصرها و قيمة بعض عناصرها**

```
datatype arrayName[size] = {value1, value2, ..};
```

- datatype: هو نوع القيم التي يمكن تخزينها في عناصر المصفوفة.
- size: هو عدد عناصر المصفوفة.
- arrayName: هو إسم المصفوفة .
- [ ]: هذا الرمز يمثل من كم عدد تتألف المصفوفة.
- **أمثلة حول طريقة تعريف مصفوفة أحادية(One Dimensional Array)**
- هنا قمنا بتعريف مصفوفة ذات بعد واحد إسمها x نوعها int تتألف من 5 عناصر. int x[5];
- هنا قمنا بتعريف مصفوفة ذات بعد واحد إسمها x نوعها int ووضعنا فيها 6 عناصر, و هذا يعني أن عدد عناصرها أصبح 6 لأننا لم نحدد عدد عناصرها.

```
int x[] = {1, 2, 3, 4, 5, 6}
```

- هنا قمنا بتعريف مصفوفة ذات بعد واحد اسمها x ونوعها int تتألف من 5 عناصر, و قمنا بوضع قيم أولية في أول 3 عناصر فيها .

```
int x[5] = {1, 2, 3};
```

فوائد المصفوفات

- لنفترض الآن أننا قمنا بتعريف مصفوفة نوعها **int**
- إسمها **a** وتتألف من **5** عناصر.
- `int a[] = { 10, 20, 30, 40, 50 };`
- يمكنك تصور شكل المصفوفة **a** في الذاكرة كالتالي.

عدد عناصر المصفوفة هو 5

قيمة العنصر (Element Value)	10	20	30	40	50
رقم العنصر (Element Index)	a[0]	a[1]	a[2]	a[3]	a[4]

- بما أن المصفوفة تتألف من **5** عناصر, تم إعطاء العناصر أرقام **indexes** بالترتيب من **0** إلى **4**.
- اذاً هنا أصبح عدد عناصر المصفوفة يساوي **5** و هو ثابت لا يمكن تغييره لاحقاً في الكود.
و للوصول لقيمة أي عنصر نستخدم **index** العنصر الذي تم إعطاؤه له

Example 1:

في المثال التالي, قمنا بتعريف مصفوفة, ثم غيرنا قيمة العنصر الأول والعنصر الخير فيها, و من ثم عرضنا قيمة جميع العناصر.

```
#include<iostream.h>
void main()
{
    //قمنا بتعريف مصفوفة تتالف من ٥ عناصر
    int arr[]={10,20,30,40,50};
    // قمنا بتغيير قيمة العنصر الاول والعنصر الاخير في المصفوفة
    arr[0]=1;
    arr[4]=7;
    // قمنا بعرض قيم جميع عناصر المصفوفة
    cout<<"arr[0]="<<arr[0]<<endl;
    cout<<"arr[1]="<<arr[1]<<endl;
    cout<<"arr[2]="<<arr[2]<<endl;
    cout<<"arr[3]="<<arr[3]<<endl;
    cout<<"arr[4]="<<arr[4]<<endl;
}
```

```
arr[0]=1
arr[1]=20
arr[2]=30
arr[3]=40
arr[4]=7
Press any key to continue
```

Example 2:-

في المثال التالي قمنا بوضع القيمة **0** كقيمة أولية لجميع عناصر المصفوفة, ثم قمنا بعرض قيمها.

```
#include<iostream.h>
void main()
{
    int a[5]={0};
    cout<<"a[0]="<<a[0]<<endl;
    cout<<"a[1]="<<a[1]<<endl;
    cout<<"a[2]="<<a[2]<<endl;
    cout<<"a[3]="<<a[3]<<endl;
    cout<<"a[4]="<<a[4]<<endl;
}
```

```
a[0]=0
a[1]=0
a[2]=0
a[3]=0
a[4]=0
Press any key to continue
```

طريقة وضع قيم أولية لعناصر المصفوفة في C++

- عند إنشاء مصفوفة جديدة وتحديد عدد عناصرها وبدون إعطائها قيم أولية, سوف نجد قيم غريبة في بعض عناصرها (قيم عشوائية).
  سبب ذلك أن هذه القيم كانت موجودة مسبقاً في الذاكرة.
- لذا في حال إنشاء مصفوفة جديدة مع حذف أي قيم عشوائية قد تكون موجودة فيها, يجب أن نقوم بإدخال القيمة الأولية التي نريد وضعها لعناصر المصفوفة لحظة إنشائها.

في المثال التالي, قمنا بتعريف مصفوفة تتألف من 5 عناصر و لم نعطها قيم أولية, ثم قمنا بعرض القيم الافتراضية الموجودة فيها.

```
#include<iostream.h>
void main()
{
    int a[5];
    cout<<"a[0]="<<a[0]<<endl;
    cout<<"a[1]="<<a[1]<<endl;
    cout<<"a[2]="<<a[2]<<endl;
    cout<<"a[3]="<<a[3]<<endl;
    cout<<"a[4]="<<a[4]<<endl;
}
```

```
a[0]=-858993460
a[1]=-858993460
a[2]=-858993460
a[3]=-858993460
a[4]=-858993460
Press any key to continue
```

طريقة معرفة عدد عناصر المصفوفة فى C++

- إذا أردت معرفة عدد عناصر أي مصفوفة, يمكنك الحصول عليه من خلال قسمة حجم المصفوفة, على حجم اول عنصر فيها.

```
#include<iostream.h>
void main()
{
    int a[]={7,4,5,6,7,3,2,1,9,7};
    cout<<"size of array a:"<<sizeof(a)<<endl;
    cout<<"size of a[0]:"<<sizeof(a[0])<<endl;
    int n = sizeof(a) / sizeof(a[0]);
    cout<<"number of element array="<<n<<endl;
}
```

```
size of array a:40
size of a[0]:4
number of element array=10
Press any key to continue
```

التعامل مع المصفوفة بواسطة حلقة التكرار في C++

- قراءة عناصر مصفوفة من لوحة المفاتيح:

```
#include<iostream.h>
void main()
{
    int a[4];
    for(int i=0;i<4;i++)
    cin>>a[i];
}
```

```
7
8
9
5
Press any key to continue
```

```
#include<iostream.h>
void main()
{
    int num[5];
    for(int i=0;i<=4;i++)
    cin>>num[i];
}
```

```
5
7
6
9
3
Press any key to continue
```

- طباعة عناصر مصفوفة على الشاشة

```
#include<iostream.h>
void main()
{
    int a[4];
    for(int i=0;i<4;i++)
    cout<<"a[i]="<<a[i]<<endl;
}
```

```
#include<iostream.h>
void main()
{
    int a[4];
    for(int i=0;i<4;i++)
    cin>>a[i];
    for(i=0;i<4;i++)
    cout<<"a[i]="<<a[i]<<endl;
}
```

```
6
7
8
9
a[i]=6
a[i]=7
a[i]=8
a[i]=9
Press any key to continue
```

طريقة ثانية لطباعة قيم المصفوفة باستخدام الايعاز for loop

```
#include<iostream.h>
void main()
{
//قمنا بتعريف مصفوفة تتكون من ٤ عناصر
    int arr[4]={10,20,30,40};
//قمنا بتغيير قيمة العنصر الاول والعنصر الاخير من المصفوفة
    arr[0]=1;
    arr[3]=7;
//قمنا بطباعة عناصر المصفوفة
    for(int i=0;i<4;i++)
    cout<<"arr[i]="<<arr[i]<<endl;
}
```

```
arr[i]=1
arr[i]=20
arr[i]=30
arr[i]=7
Press any key to continue
```

Example 1: write c++ program to display 2nd and 5th elements of array distance

اكتب برنامج بلغة C++ لعرض قيمة العنصر الثاني وقيمة العنصر الخامس من المصفوفة

```
#include<iostream.h>
void main()
{
    double distance[]={23.14,70.52,104.08,468.78,6.28};
    cout<<"2nd element is:"<<distance[1]<<endl;
    cout<<"5th element is:"<<distance[4]<<endl;
}
```

```
2nd element is:70.52
5th element is:6.28
Press any key to continue
```

Example 2: write c++ program to read 5 numbers and print it in reverse order:

اكتب برنامج بلغة C++ لقراءة 5 ارقام وطباعتها بترتيب معكوس

```
#include<iostream.h>
void main()
{
    int a[5];
    cout<<"Enter 5 numbers \n";
    for(int i=0;i<5;i++)
    {
        cout<<i<<":";
        cin>>a[i];
    }
    cout<<"The reverse order is: \n";
    for(i=4;i>=0;i--)
    cout<<i<<":"<<a[i]<<endl;
}
```

```
Enter 5 numbers
0:8
1:5
2:9
3:4
4:7
The reverse order is:
4:7
3:4
2:9
1:5
0:8
Press any key to continue
```

Example 3: write c++ program to find the summation of array elements:
اكتب برنامج بلغة C++ لايجاد مجموع عناصر مصفوفة

```
#include<iostream.h>
void main()
{
    int const l=10;
    int a[l];
    int sum=0;
    cout<<"Enter 10 numbers \n";
    for(int i=0;i<10;i++)
    {
        cout<<"Enter value "<<i<<": ";
        cin>>a[i];
        sum=sum+a[i];
    }
    cout<<"sum is  "<<sum<<endl;
}
```

```
Enter 10 numbers
Enter value 0: 10
Enter value 1: 8
Enter value 2: 9
Enter value 3: 3
Enter value 4: 2
Enter value 5: 1
Enter value 6: 7
Enter value 7: 6
Enter value 8: 5
Enter value 9: 4
sum is  55
Press any key to continue
```

Example 4: write c++ program to find the minimum value in array of 8 numbers:
اكتب برنامج بلغة C++ لايجاد اقل قيمة في مصفوفة عدد عناصرها 8

```
#include<iostream.h>
void main()
{
    int n=8; int a[]={18,25,36,44,12,60,75,89};
    int min=a[0];
    for(int i=0;i<n;i++)
    if(a[i]<min)
        min=a[i];
    cout<<"The minimum number in array is "<<min<<endl;
}
```

```
The minimum number in array is 12
Press any key to continue
```

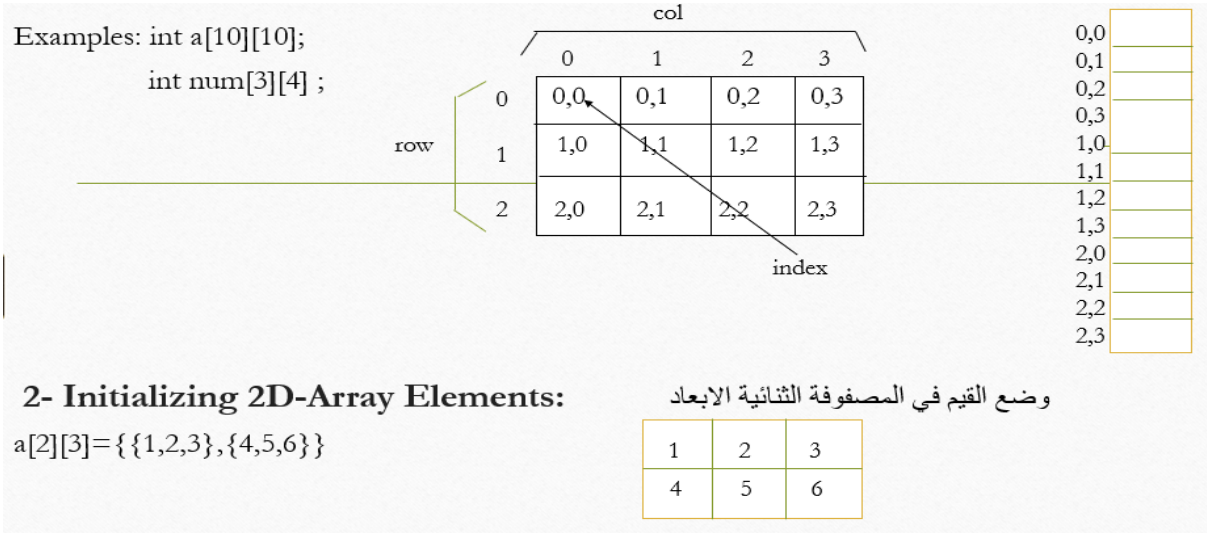
المصفوفات الثنائية الابعادarray of two dimension

- المصفوفة ثنائية البعد : هي عبارة متغير واحد يتألف من عدة عناصر من نفس النوع وكل عنصر في المصفوفة يمكن تخزين قيمة واحدة من نفس النوع وعناصر المصفوفة تتميز عن بعضها من خلال رقم محدد يعطى لكل عنصر يسمى index اول عنصر في المصفوفة تبدأ بالرقم 0,0 .
- طريقة تعريف مصفوفة ثنائية
- Data type Array-name [row-size][col-size];
- Data type: نوع المصفوفة الثنائية

Array-name: اسم المصفوفة الثنائية

Row-size : حجم اوعدد عناصر الصف

Col-size: حجم اوعدد عناصر العمود



Read\Write\Process Array Elements

Example 1: Write c++ program to read 15 number, 5 numbers per row, then print them:

اكتب برنامج C ++ لقراءة 15 رقمًا ، 5 أرقام في كل صف ، ثم اطبعها:

```
#include<iostream.h>
void main()
{  int a[3][5];
int i,j;
for( i=0; i<3; i++ )
for( j=0; j<5; j++)
cin>>a[i][j];
for( i=0; i<3; i++ )
{
for( j=0; j<5; j++)
cout<<a[i][j];
cout<<endl;
}   }
```

```
2
3
4
6
7
6
8
0
4
5
9
3
2
7
8
23467
68045
93278
Press any key to continue
```

Example 2: Write C++ program to read (4*4) 2D array, then find the summation of the array elements, finally print these element.

اكتب برنامج ++ C لقراءة مصفوفة ثنائية الأبعاد (4 * 4) ، ثم اوجد مجموع عناصر المصفوفة،وأخيراً اطبع هذه العناصر.

```
#include<iostream.h>
void main()
{
int a[4][4],i,j,sum=0;
for(i=0;i<4;i++)
for(j=0;j<4;j++)
cin>>a[i][j];
for(i=0;i<4;i++ )
for(j=0;j<4;j++)
sum+=a[i][j];
cout<<"sum="<<sum<<endl;
for(i=0;i<4;i++)
{
for(j=0;j<4;j++)
cout<<a[i][j];
cout<<endl;
}
}
}
```

```
1
2
3
4
1
2
3
4
1
2
3
4
1
2
3
4
sum=40
1234
1234
1234
1234
Press any key to continue
```

Example 3: Write C++ program to read 3*4 2D array then find the summation of each row.

اكتب برنامج بلغة ++C لقراءة مصفوفة ثنائية بأبعاد 3*4 ثم اوجد مجموع كل صف

```
#include<iostream.h>
void main()
{   int a[3][4],i,j,sum;
for( i=0; i<3; i++ )
for( j=0; j<4; j++)
cin>>a[i][j];
for( i=0; i<3; i++ )
{
sum=0;
for( j=0; j<4; j++)
sum+=a[i][j];
cout<<"sum="<<sum<<endl;
}
for( i=0; i<3; i++ )
{
for( j=0; j<4; j++)
 cout<<a[i][j];
 cout<<endl;
}
}
```

```
1
2
3
4
5
6
7
8
3
5
2
6
sum=10
sum=26
sum=16
1234
5678
3526
Press any key to continue
```

Example 4: write C++ program to read 3*4 2D array, then replace each value equal 5 with 0

اكتب برنامج بلغة C++ لقراءة مصفوفة ثنائية الابعاد (3*4) ثم استبدل كل قيمة تساوي 5 بـ 0

```
#include<iostream.h>
void main()
{   int a[3][4],i,j;
for( i=0; i<3; i++ )
for( j=0; j<4; j++)
cin>>a[i][j];
for( i=0; i<3; i++ )
for( j=0; j<4; j++)
if(a[i][j]==5)
a[i][j]=0;
for( i=0; i<3; i++ )
{
for( j=0; j<4; j++)
cout<<a[i][j];
cout<<endl;
}
}
```

```
1
2
3
4
5
3
2
1
5
3
4
5
1234
0321
0340
Press any key to continue
```

Example 5: Write C++ program to addition two 3*4 arrays.

اكتب برنامج بلغة C++ لاضافة (مجموع) مصفوفتين بالابعاد (3*4)

```
#include<iostream.h>
void main()
{
int a[3][4], b[3][4],c[3][4];
int i,j;
cout<<"enter element of array a \n";
for( i=0; i<3; i++ )
for( j=0; j<4; j++)
cin>>a[i][j];
cout<<"enter element of array b \n";
for( i=0; i<3; i++ )
for( j=0; j<4; j++)
cin>>b[i][j];
for( i=0; i<3; i++ )
for( j=0; j<4; j++)
c[i][j]=a[i][j]+b[i][j];
cout<<"array c \n";
for( i=0; i<3; i++ )
{
    for( j=0; j<4; j++)

    cout<<c[i][j]<<"   ";
    cout<<endl;
}   }
```

```
enter element of array a
1 2 3 4
1 2 3 4
1 2 3 4
enter element of array b
5 6 7 8
5 6 7 8
5 6 7 8
array c
6   8   10   12
6   8   10   12
6   8   10   12
Press any key to continue
```

Example 6 : Write C++ program to convert 2D-array into 1D-array.

اكتب برنامج بلغة C++ لتحويل مصفوفة ثنائية الابعاد الى مصفوفة أحادية الابعاد

```
#include<iostream.h>
void main()
{
    int a[3][4], b[12];
    int i,j,k=0;
//قراءة مصفوفة ثنائية الابعاد
    for(i=0;i<3;i++)
        for(j=0;j<4;j++)
            cin>>a[i][j];
// طباعة مصفوفة ثنائية الابعاد
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
            cout<<a[i][j]<<" ";
            cout<<endl;
    }
//تحويل مصفوفة ثنائية الابعاد الى مصفوفة احادية البعد
    for(i=0;i<3;i++)
        for(j=0;j<4;j++)
        {
            b[k]=a[i][j];
            k++;
        }
//طباعة مصفوفة احادية البعد
    for(i=0;i<k;i++)
        cout<<b[i]<<" ";
        cout<<endl;
}
```

```
1 2 3 4 5 6 7 8 9 1 2 3
1 2 3 4
5 6 7 8
9 1 2 3
1 2 3 4 5 6 7 8 9 1 2 3
Press any key to continue
```

Example 7: Write C++ program to replace each element in the main diameter (diagonal) with zero.

اكتب برنامج بلغة C++ ليحل محل كل قيمة في القطر الرئيسي بالقيمة 0

```
#include<iostream.h>
void main ( )
{
        int a [ 3 ] [ 3 ];
        int i , j;
        for ( i = 0 ; i < 3; i++ )
            for ( j = 0 ; j < 3; j++ )
                cin >> a [ i ] [ j ];
        for ( i = 0 ; i < 3; i++ )
            for ( j = 0 ; j < 3; j++ )
                if ( i == j )    a [ i ] [ j ] = 0;
        for ( i = 0 ; i < 3; i++ )
            {
                for ( j = 0 ; j < 3; j++ )
                    cout << a [ i ] [ j ];
                cout << endl;
            }
}
```

0,0		
	1,1	
		2,2

i = j

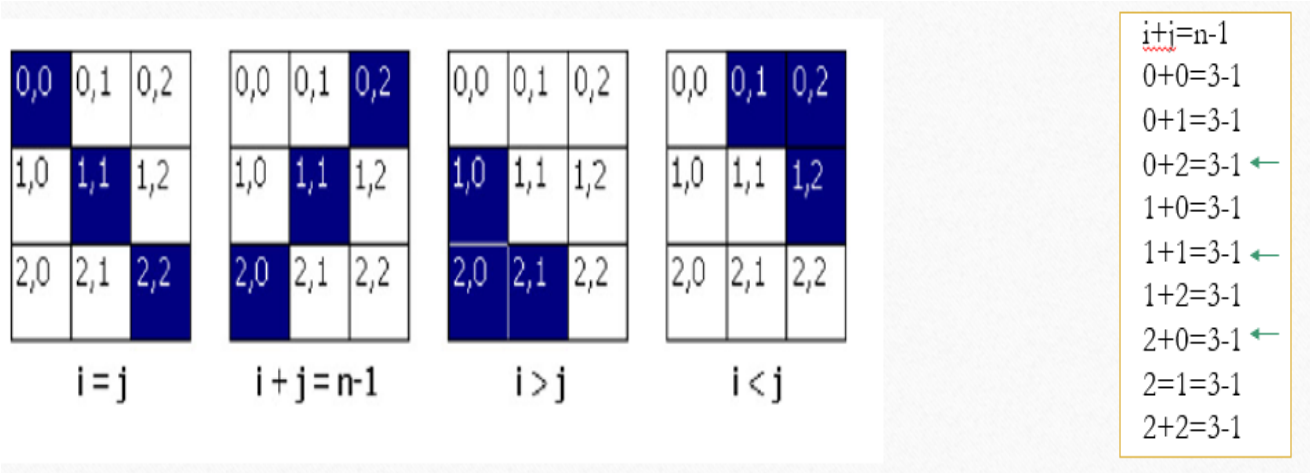
كيفية طباعة او اجراء عمليات حسابية على عناصرمصفوفة ثنائية مربعة (عدد الصفوف=عدد الاعمدة)

1- عناصر القطر الرئيسي i=j

2- عناصر القطر الثانوي i+j=n-1

3- عناصر فوق القطر الرئيسي i>j

4- عناصر تحت القطر الرئيسي i>j



Example 8: Write C++ program to print the square root of an array.

اكتب برنامج بلغة C++ لطباعة الجذر التربيعي لمصفوفة

```
#include<iostream.h>
#include<math.h>
void main()
{
    int a[3][3]; double b[3][3];
    int i,j;
//قراءة مصفوفة ثنائية الابعاد
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            cin>>a[i][j];
// حساب الجذر التربيعي لكل قيمة في المصفوفة
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            b[i][j]=sqrt(a[i][j]);
//طباعة المصفوفة الخاصة بناتج حساب الجذر التربيعي
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            cout<<b[i][j]<<"    ";
            cout<<endl;
    }
}
```

```
4 9 16 25 36 49 64 81 16
2   3   4
5   6   7
8   9   4
Press any key to continue
```

Example 9: Write C++ program to read (3*3) 2D array, then find the summation of the main diagonal and its secondary diagonal of the array elements, finally print these elements:

```
#include<iostream.h>
void main()
{
int a[3][3],i,j,x=0,y=0;
for( i=0; i<3; i++ )
for( j=0; j<3; j++)
cin>>a[i][j];

for( i=0; i<3; i++ )
for( j=0; j<3; j++)
if(i==j)
x=x+a[i][j];
cout<<"x="<<x<<endl;

for( i=0; i<3; i++ )
for( j=0; j<3; j++)
if(i+j==3-1)
y=y+a[i][j];
cout<<"y="<<y<<endl;
}
```

```
1 2 3
8 3 4
9 2 7
x=11
y=15
Press any key to continue
```

الدوال Functions

Functions (الوظيفة او دالة)

الدوال:- هي عبارة عن مجموعة جمل او تعليمات ذات كيان خاص للقيام بعمل معين او **هي عبارة عن برنامج فرعي يؤدي مهمة معينة مكملة للمهمة التي يؤديها البرنامج الرئيسي (main program)** .

أظهرت التجربة ان افضل طريقة لتطوير وصيانة برنامج الكتروني هو انشائه من أجزاء صغيرة او (وحدات) والوحدات في لغة C++ تسمى **function** او الدالة او الوظيفة وهذه الوظائف او الدوال مفيدة جدا لقراءة وكتابة وتصحيح وتعديل البرامج المعقدة. ويمكن أيضا دمجها بسهولة في البرنامج الرئيسي. في لغة C++ تعد الـ **main()** نفسها دالة تعني ان الدالة الرئيسية هي استدعاء دوال أخرى لاداء مهام مختلفة.

المزايا الرئيسية لاستخدام الدالة:

1- من السهل كتابة دالة صغيرة صحيحة.

2- من السهل قراءة وكتابة وتصحيح الدالة.

3- سهولة الحفاظ على هذه الدالة او تعديلها

4- تمثل الدوال الصغيرة الى انها ذاتية التوثيق وقابلة للقراءة بشكل كبير

5- يمكن استدعاء الدالة لعدد من المرات في أي مكان ذات متغيرات مختلفة

الصيغة العامة للدوال (Function)

```
Return-type  function-name (parameter1, parameter2……)
{
(Body of function)
Statement1;
Statement2;
:
Statement n
(Return some thing)
}
```

- **Return-type:-** نحدد بها نوع القيمة التي سوف ترجعها الدالة بعد ان يتم استدعائها من قبل البرنامج الرئيسي والايعاز الذي يرجع هذه القيمة هو return والذي يكون بالشكل التالي

- Return(return value);

- مع ملاحظة ان الدالة قد لاترجع شيء لذلك يمكن تعريف الـ return-type بشكل void (خالي) وبذلك الدالة لا تحتوي على ايعاز ال return.

- **Function-name:-** هو اسم الدالة التي بواسطتها يتم استدعاء الدالة داخل البرنامج الرئيسي على ان تكون بشروط الـ identifier name هو اسم عام مثل أسماء المتغيرات وأسماء الدوال.

- **Parameter-list:-** قائمة أسماء المتغيرات مع أنواعها هذه المتغيرات تجلب قيمها من البرنامج الرئيسي وتنقلها الى داخل الدالة ليتم التعامل معها مع ملاحظة ان يفصل بين المتغيرات فوارز وقد لا تعتمد الدالة على أي متغير عندها تكتب الاقواس فارغة

- قد يكون نوع الدالة هو int او float اوchar وما الى ذلك او قد يتم الإعلان عنها على انها نوع void مما يعلم المترجم بعدم ارجاع أي قيمة الى الدالة

Void function-name(….)

Int function-name(….)

- يقال ان أي متغير يتم تعريفه في جسم الدالة يكون محليا (local) لتلك الدالة .وتعتبر المتغيرات الأخرى التي لم يتم الإعلان عنها كوسيطات او في جسم الدالة تسمى عام (global) وتعريفها خارج الدالة . مثال

- Void square(int a, int b) ⟶ a,b are the formal argument
- float output(void) ⟶ function without formal arguments

أنواع الدوال
type of user-defined function

1- دالة ليس لديها متغيرات parameter ولم ترجع قيمة

Example: Write a function that print a massage "welcome, this is a function"

```
#include<iostream.h>
void fun_print()
{
    cout<<"welcom, this is a function"<<endl;
}
void main()
{
fun_print();
}
```

```
welcom, this is a function
Press any key to continue
```

2- دالة لها متغيرات parameter ولم ترجع قيمة

Example: write a function that take your name and print message "welcome", your name"

عمل دالة لها متغير واحد ومن ثم تطبع الجملة التالية

```
#include<iostream.h>
void print_name(char name)
{
    cout<<"welcom    "<<name<<endl;
}
void main()
{
    char a='s';
print_name(a);
}
```

```
welcom   s
Press any key to continue
```

3- دالة لها متغيرات وترجع قيم.

Example: Write a program using functions to find the largest of two values and print the largest value.

```
#include<iostream.h>
int max(int a, int b)
{
    int c;
    if(a>b)
        c=a;
    else
        c=b;
    return(c);
}
void main()
{
cout<< max(5,6)<<endl;
}
```

```
6
Press any key to continue
```

Example1:- write a program to calculate the squared value of a number passed from main function. use this function in a program to calculate the squares of numbers from 1 to 10

اكتاب برنامج لحساب القيمة التربيعية لرقم تم معالجته من الوظيفة الرئيسية. استخدم هذه الوظيفة في برنامج لحساب مربعات الأرقام من 1 إلى 10

```
#include<iostream.h>
int square(int y)
{
    int z;
    z=y*y;
    return(z);
}
void main()
{
int x;
for(x=1;x<=10;x++)
cout<<x<<"   "<<square(x)<<endl;
}
```

```
1  1
2  4
3  9
4  16
5  25
6  36
7  49
8  64
9  81
10  100
Press any key to continue
```

Example2: write a program using a function to calculate the average of two numbers entered by the user in the main program

اكتب برنامج باستخدام دالة لحساب معدل رقمين أدخلهما المستخدم في البرنامج الرئيسي

```
#include<iostream.h>
float aver(int x1 , int x2)
{
    float z;
    z=(x1+x2)/2.0;
    return(z);
}
void main()
{
float x;
int num1,num2;
cout<<"Enter 2 positive number\n";
cin>>num1>>num2;
x=aver(num1,num2);
cout<<x<<endl;
}
```

```
Enter 2 positive number
70
80
75
Press any key to continue
```

Example3: write a program using a function to find the summation of the following series:

$\sum_{i=1}^{n} i^2 = 1^2 + 22 + 32 + \cdots n^2$

اكتب برنامج باستخدام الدوال لإيجاد مجموع هذه السلسلة التالية

```
#include<iostream.h>
int summation(int x)
{
    int i=1, sum=0;
    while(i<=x)
    {
        sum+=i*i;
        i++;
    }
    return(sum);
}
void main()
{
int x,y;
cout<<"Enter positive number\n";
cin>>x;
y=summation(x);
cout<<"sum is "<<y<<endl;
}
```

```
Enter positive number
7
sum is 140
Press any key to continue
```

Example4: write a function to find the largest integer among three integers entered by the user in the main function.

اكتب دالة لايجاد اكبر قيمة من بين ثلاث قيم صحيحة ادخلهما المستخدم في البرنامج الرئيسي

```
#include<iostream.h>
int max(int x1 , int x2 , int x3)
{
    int big;
    big=x1;
    if(x2>big)
        big=x2;
    if(x3>big)
        big=x3;
    return(big);
}
void main()
{
int largest,x1,x2,x3;
cout<<"Enter 3 integer number\n";
cin>>x1>>x2>>x3;
largest=max(x1,x2,x3);
cout<<"the number is largest  "<<largest<<endl;
}
```

```
Enter 3 integer number
7
10
8
the number is largest  10
Press any key to continue
```

Example5: write a program using a function to find the equation of the following

$$Z=X^3 + Y^3/ K^3 +5/P^3$$

```
#include<iostream.h>
float power(int a)
{
    int p=1;
    for(int i=1;i<=3;i++)
        p=p*a;
    return(p);
}
void main()
{
int x,y,k,p;
float z=0;
cin>>x>>y>>k>>p;
z=((power(x) + power(y)) / power(k)) + (5/power(p));
cout<<z<<endl;
}
```

```
6
5
8
9
0.672874
Press any key to continue
```

Example6: write a program to compute Z by using a function to compute the factorial and pwer to the number $Z=\frac{1!}{X^1}-\frac{2!}{X^2}+\frac{3!}{X^3}-\frac{4!}{X^4}+\cdots\frac{n!}{X^n}$

```
#include<iostream.h>
int fact(int a)
{
    int f=1;
    for(int i=1;i<=a;i++)
        f=f*i;
    return(f);
}
int power(int x , int y)
{
    int p=1;
    for(int j=1;j<=y;j++)
        p=p*x;
    return(p);
}
void main()
{
int x,n,s=1;
float z=0;
cin>>x>>n;
for(int y=1;y<=n;y++)
{
    z=z+fact(y)/power(x,y)*s;
    s=s*(-1);
}
cout<<"z=  "<<z<<endl;
}
```

```
2
7
z= 30
Press any key to continue
```

4. الأسئلة البعدية في نهاية الفصل:

- Write C++ program to read (3*3) 2D array, then find the summation of the array elements
- Write C++ program to print the reverse order of 3 numbers
- Write C++ program to convert 2D-array into 1D-array.
- write a program using a function to calculate the sum and average of two numbers entered by the user in the main program

- **المصادر الاساسية :**

1cpp.pdf (uotechnology.edu.iq)

م.م/ بشارسعدون مهدي ,البرمجة المهيكلة ++c , الجامعة التكنولوجية قسم علوم الحاسوب

- **المصادر المقترحة:** لايوجد
- **روابط مقترحة ذات صلة:** مجموعة فيديوهات للمحاضرات المشروحة في هذا الكتاب.

https://www.youtube.com/playlist?list=PLnI1fRHezj5F8y024ulIKxV-DqdIyR4E6