الدرس الثاني:

١- مقدمة عامة.

٢- انظمة التشغيل الأولى.

٣- لغات الآلة ، لغات التجميع ، اللغات عالية المستوى.

٤- تاريخ الـ c++ .

٥- مكتبة الـ c++ القياسية.

٦- مقدمة تمهيدية للبرمجة بتقنية الكائن الموجه.

بإسمه نبدا وعليه نتوكل وبه نستعين.

مقدمة

c++ لغة برمجة قوية لكل من لهم توجه إلى أن يكون مبرمجين سواء كانو مبتدئين أو من لديهم خبرة في البرمجة ، وهي لغة برمجة مناسبة جدا لبناء أنظمة معلومات ضخمة وقوية.

مسلسل هذه الدورة يركز على التبسيط والتدرج عن طريق الكثير من الأمثلة والتوضيحات والتمارين واستخدام تقنيات برمجية رائعة من أهم هذه التقنيات هي تقنية الكائن الموجه object-oriented ، لكي نصل لغايتنا من شرح هذه التقنية استخدمنا طريقة الكود المباشر (live-code approach) وهي طريقة التوضيح عن طريق كتابة كود وتنفيذه وعرض النتائج.

في الفصل الأول (هذا الفصل) مقدمة لتوضيح اساسيات الكمبيوتر ومقدمة للغة الـ c++ وتاريخها وكيف تطورت ، وفي الفصول الأولى التي تليها سوف يتم شرح جمل ضخمة كافيه لأن يكون لديك أساس صلب وقوي تنطلق منه إلى باقي الفصول

لغة الـ C++ هي اللغة الرائدة اليوم والمهيمنة على عرش البرمجة وتطوير البرمجيات، ومن خلال هذه الدورة (الكتاب) سوف نقدم لك البرمجة بلغة الـ C++ بنسختها العالمية والموحدة في الولايات المتحدة الأمريكية بواسطة مؤسسة المقاييس القومية الأمريكية (ANSI) من خلال محاولات المنظمة العالمية لتوحيد المقاييس (ISO).

على مدى السنين الماضية، معظم المبرمجين تعلموا البرمجة بطريقة البرمجة الهيكلية structured programmign سوف نتعلمها من خلال هذه الدورة بالإضافة إلى طريقة البرمجة الموجهه object-oriented programming وهي نواة البرمجة الحديثة التي سوف نركز عليها بكثافة في هذه الدورة.

لماذا نتعلم الطريقتين ؟ الجواب يكمن في ضرورة تعلم الطريقتين ، البرمجة الهيكلية والبرمجة الموجهه، البرمجة الهيكلية لتركيب بعض التطبيقات لتعلم طريقة تركيب البرنامج بالشكل الصحيح واكتساب مهارات لتوسيع الحس البرمجي ، اما البرمجة الموجهه فهي مفتاح البرمجة الحديثة التي اضافت مزيدا من الوضوح والبساطة على تركيب البرنامج بشكل يجعله محاكيا للطبيعة.

أنظمة التشغيل الأولى

معالجة البيانات البدائية في الكمبيوترات الأولى تستخدم طريقة يستحيل استخدامها في وقتنا الحاضر وهي طريقة المعالجة بالدفعات batch processing ، وفيها ينفذ أمر (تعليمة) واحدة كل مرة.

مستخدمي هذه الطريقة عند استخدامها وتنفيذها يسمونها " المعالجة بالدفعة احادية المستخدم " batch processing single-user ، وذلك لأنها لمستخدم واحد يقوم بتشغيل برنامج واحد ويعالج أمر واحد في كل مرة لتكون المهمة منجزة عن طريق دفعات. بهذه الطريقة يحتاج المستخدم لساعات أو حتى أيام للحصول على تقرير نتائج بياناته.

أي نظام تشغيل هو عبارة عن نظام برمجي متكون من مجموعة برامج تعمل معا لإنجاز مهام معينة، تطورت انظمة التشغيل على مراح وذلك لزيادة انجازها للمهام لرفع طاقتها العلمية والإنتاجية.

البرمجة المتعددة multiprogramming - ذكرنا فيما سبق " البرمجة بالدفعات " وعليك أن تتخيل مدى الوقت المهدور والإنتظار الممل لو استمر الإعتماد عليها ، عملية واحدة كل مرة ويقوم بها مستخدم واحد!! - البرمجة المتعددة تقنية اضافة قفزة هائلة في انظمة التشغيل وذلك عن طريق اتاحة امكانية استخدام الكبيوتر لأكثر من مستخدم لعدة تطبيقات في نفس الوقت.

على رسل ما سبق ، " البرمجة المتعددة " ، تم تطوير هذه التقنية عن طريق مجموعة مبرمجين في الصناعة والجامعات لتوفير تقنية المشاركة الوقتية (timesharing) وذلك لإتاحة المزيد من التطبيقات في نفس الوقت لعدة مستخدمين لتبدو وكأنها متزامنة (تحدث في نفس الوقت) وهي ليست كذلك !!

المشاركة الزمنية يمكن تعريفها " بأنها عملية تنفيذ برمجية يتناوب فيها برنامجان أو اكثر بسرعة عالية بحيث تبدو وكأنها متزامنة ." المشاركة الزمنية من أهم التقنيات في انظمة التشغيل الخاصة بالشبكات، حيث يتم عن طريقها ربط مجموعة كمبيوترات بحيث تتشارك في المحتوى وتنفيذ المهام في نفس الوقت لعدة مستخدمين.

لغات الآلة ، لغات التجميع ، اللغات عالية المستوى

يمكن تقسيم اللغات التي يتعامل معها الكمبيوتر إلى ثلاث أنواع بشكل متوالي:

١- لغات الآلة.

٢- لغات التجميع.

٣- اللغات عالية المستوى

لغة الآلة machine language

هي اللغة الطبيعية للكمبيوتر التي يفهمها مباشرة دون الحاجة إلى اي وسيط ترجمة. وهي تتكون من سلاسل من الأصفار والآحاد تكون معرفة بواسطة تصميم الكمبيوتر المادي. كل لغة آلة تعتمد على الآله نفسها، على سبيل المثال: كل آله لها لغة خاصة تستخدم على هذا النوع من الكمبيوتر فقط.

دائما ما تكون لغات الآلة صعبة ومعقدة للفهم البشري والبرمجة بها صعبة وتثقيلة جدا وغالبا ما تحتوي الأخطاء، فعند القيام بعملية جمع بسيطة نحتاج أن تمثل كل رقم أو عملية بسلسلة من الأصفار والآحاد مرتبة بطريقة مختلفه لكل رمز أو حرف أو رقم أو عملية.

كما ترون ، لغة الآلة لغة مستحيلة ، غير مقبولة ابدا للبرمجة، لذلك قام المبرمجين بإستخدام اللغة الإنجليزية لعمل اختصارات لتمثيل العمليات الإبتدائية ، هذه الإختصارات شكلت فيما بعد أساسيات لغة التجميع.

لغات التجميع assembly languages

البرامج التي تقوم بترجمة اختصارات هذه اللغة إلى لغة الآلة تسمى assemblers . تقوم هذه المترجمات بتحويل برامج لغات التجميع وترجمتها إلى لغة الآلة.

لغات البرمجة العالية المستوى level languages high

استخدام الكبيوتر يزيد بسرعة عالية مع تطور لغات التجميع، لكن المبرمجين لا يزالون يستخدمون الكثير من الإختصارات " التعليمات " لتأسيس أبسط المهام لزيادة سرعة عملية البرمجة.

مع ازدياد الحاجة لزيادة سرعة البرمجة بشكل مُلِحّ، ظهرت لغات البرمجة العالية المستوى.، فبعبارة مفردة في أحدى هذه اللغات تستطيع أن تنفذ عدد ضخم من المهام.

برامج الترجمه لهذه اللغة تسمى الـ compilers وهي برامج تقوم بتحويل برامج هذه اللغات إلى لغة الآلة.

اللغات عالية المستوى تبدو طبيعية أكثر من اللغات التي سبقتها، تشابه اللغة الإنجليزية اليومية في أوامرها وتعليماتها. ومن الأمثلة عليها C++ ، #C ، Java ، vb.net ... الخ.

عملية ترجمة برنامج إلى لغة الآلة غالبا ما يأخذ قدرا من وقت الكمبيوتر، هناك مايسمى بالبرامج المفسره programs interpreter ، وهي برامج تمت إضافتها في بيئة تطوير البرامج في المكونات الجديدة وذلك لتنفيذ برامج اللغات عالية المستوى مباشرة ، كل نسخة جديدة من هذه البرامج تكون منقحة أكثر من التي تليها.

تاريخ الـ C++

BCPL ..... > B .... > C ..... > C++

في عام ١٩٦٧ قام **مارتن ريتشارد** بتطوير النتاج البرمجي لمن سبقة لإنشاء لغة برمجة تسمى BCPL تكتب بها برمجيات ومترجمات انظمة التشغيل.

أتى بعد ذلك **كين ثومبسن** وقام بنمذجة أو تنميط الكثير من مكونات BCPL لينتج عن ذلك لغة جديدة وهي لغة الـ B ، وقام **كين ثومبسن** بكتابة نسخته الاولى من نظام التشغيل UNIX بهذه اللغة في معامل بيل عام ١٩٧٠.

أتى بعد ذلك **دينيس ريتشي** وقام بتطوير لغة الـ B وذلك لتكوين لغته الجديدة وهي

لغة الـ C وذلك في معامل بيل أيضا.

وحضيت هذه النسخة بالذات بإنتشار واسع حيث كتب بها معظم انظمة التشغيل على رأسها نظام الـ UNIX الذي تم تطويره بهذه اللغة. لغة C لغة قريبة من لغة الآلة ويمكن أن تتعامل مع معظم المنصات (صناعات الكمبيوتر).

الإنتشار السريع للـ C واستخدامها الواسع على انواع كثيره من منصات الكمبيوتر أدى ، وللأسف ، إلى الكثير من الإختلافات لكل منصه، حيث يكون لكل منصه أكواد معينه لا يمكن تنفيذها على منصات أخرى، وهذه مشكلة حقيقية لمطوري البرامج، من يحتاجون لكتابة برامج متوافقه مع كل أنظمة التشغيل التي تعمل على عدة منصات.

وحدت الجهود بين ANSI و ISO وذلك لتوحيد لغة C وانتاج النسخة القياسية منها وذلك على مستوى العالم. وفعلا تم انتاج المتسند القياسي المشترك الموحد في عام ١٩٩٠ ويشار إلي بـ ANSI/ISO 9899:1990 .

فكرة مفيدة لأكواد متنقلة

بسبب توحيد لغة الـ C واعتمادها الكبير على المكونات المادية وأنها ايضا لغة متاحة على نطاق واسع، البرامج المكتوبة بالـ C في كثير من الأحوال لا يتم التعديل عليها ، او يتم التعديل عليها قليلا على مجال عريض من أنظمة التشغيل.

نتيجة للمشاكل التي واجهها الكثير من المبرمجين في تنقلية البرامج وأمكانية إعادة استخدامها ، قام **بجارن ستروترب** بتطوير لغة الـ C للحصول على لغة مهندمة جملية وبسيطة ويمكن إعادة استخدام الأكواد فيها في أي برنامج، وتطبيق برامجها على أغلب انظمة التشغيل والمنصات بدون مشاكل، وهي لغة الـ C++ التي تعتمد بشكل أساسي على الكائن الموجه في برمجتها. لغة الـ C++ لغة واضحة سهلة الفهم ، طبيعية أكثر ويمكن اجراء عمليات التصحيح والتعديل عليها بسهولة. كل هذا حدث في الثمانينات في معامل بيل.

مكتبة الـ C++ القياسية

برامج الـ C++ تتكون من قطع تسمى الأصناف والدوال، تستطيع برمجة كل جزء بما تحتاجة لتشكيل برنامجك الذي تريد.

معظم مبرمجين الـ C++ يستفيدون كثيرا من المجموعات الكثيرة والغنية من الأصناف والدوال الموجودة في مكتبة الـC++ القياسية. مكتبة الـC++ القياسية جزء هام في

تعلم برمجة الـC++. خبراء البرمجة قسموا تعلم الـC++ إلى قسمين: القسم الأول تعلم اللغة نفسها، والقسم الثاني تعلم طريقة استخدام الدوال والاصناف في مكتبة الـC++ القياسية. خلال هذه الدورة سوف نقوم بشرح الكثير منها.

ملاحظة مفيدة في هندسة البرمجيات

استخدم وسيلة " قطع البناء " لإنشاء البرامج، تجنب إعادة اختراع العربة، استخدم قطع موجودة ما امكنك ذلك.
إعادة الإستخدام software reuse ، وتمارينه هو المركز والنواة لبرمجة الكائن الموجه.

ملاحظة مفيدة في هندسة البرمجيات

عندما تبرمج بالـC++، أنت وبشكل طبيعي تحتاج لإستخدام قطع البناء : مثل الأصناف والدوال من مكتبة الـC++ القياسية، الدوال والأصناف التي صنعتها أو قام بصنعها اصدقائك في الكلية ، وايضا تلك التي من مكاتب النسخة الثالثة المتنوعة.

البرمجة بتقنية الكائن الموجه object-oriented programming

تمهيد

الغاية من تطوير البرامج تكون في توفير برامج سهلة الصيانة ذات أداء عالي بحيث تعمل على اكبر قدر ممكن من انظمة التشغيل بدون مشاكل. ذكرنا فيما سبق المشاكل التي يواجهها المبرمجين في لغة الـC عند ازدياد استعمالها على منصات وانظمة تشغيل متنوعة، والمشكلة كما ذكرنا تكمن في التوافقية وقابلية البرامج للتنقل من نظام لآخر portability ، بمعنى آخر، العجز والقصور في إعادة استخدام البرمجيات على انظمة ومنصات مختلفة software reuse .

عني **ستروترب** بهذه المشكلة واستعان بلغة برمجة تم تطويرها في اوربا وهي لغة الـ simula67 المحاكية للواقع، كانت هذه اللغة اكثر اقترابا من الواقع لإستخدامها تقنية الكائن الموجه ولكن بإمكانات أقل لا تؤهلها لأن تقوم بذاتها بدلا عن لغة الـC لذلك قام **ستروترب** بإنشاء لغة جديدة بنفس مكونات وامكانات الـC واضاف عليها مقدرات الـ simula67 لينتج لنا لغة الـC++ بتقنية الكائن الموجه.

إذا ، ماهي تقنية الكائن الموجه؟ وبماذا تكمن أهميته؟

تقنية الكائن الموجه هي تقنية يمكن عن طريقها تعليب (تحزيم) البرمجيات (مناطق معينه من البرنامج) وتحويلها إلى كائنات، لكل كائن بياناته واجراءاته التي يقوم بها بحيث يتم تطويره على حدة بطريقة تمكن مستخدمه من إعادة استخدامه على معظم الأنظمه والبرامج والمنصات بدون مشاكل. فهناك على سبيل المثال كائن التاريخ، كائن الوقت، كائن الرواتب ، كائن الفاتورة .... الخ.

لغات البرمجة التي سبقت الـ C++ تركز دائما في برمجتها على الدوال (الأفعال) فنجد أن لغة الـ C على سبيل المثال يكون البرنامج على شكل هيكل من الأوامر والدوال التي تعمل معا ، أما في لغة الـ C++ فنجد أن التركيز على الكائنات (الأسماء) فترى أن البرنامج عبارة عن أجزاء منفصلة كل جزء يقوم بوظيفه معينه يمكن تطويرة على حدة، وهذا الجزء هو الوحدة البرمجية للغة الـ C++ وهو ما يسمى بالكائن.

الوحدة الأساسية التي يتشكل منها الكائنات هي الصنف (class)، فعندما تصمم الـ class بشكل وطريقة مناسبة بحيث يمكن استخدامها على معظم المشاريع والأنظمه ويمكن تطويره على حدة فإننا بذلك وفرنا الكثير من الوقت والجهد.

المطورين في شركة مايكروسوفت وغيرها يعرفون جيدا مدى فائدة الـMFC (microsoft foundation classes) عند كتابتهم أو صيانتهم للبرامج، فهي مكتبة تحتوي على الكثير من الأصناف التي يمكن إعادة استخدمها في أي برنامج بشكل فعال وخالي من الأخطاء.

ملاحظة مفيدة في هندسة البرمجيات

مكتبة الأصناف معادة الإستخدام متاحة خلال الإنترنت بدون رسوم.

الكثير من المنظمات البرمجية قالت بأن البرمجة الموجهه تحتوي على فوائد أخرى بالإضافة لإعادة الإستخدام، وذلك أنها قدمت تقنيات جعلت البرامج أكثر وضوحا وسهولة في عمليات الصيانة والتصحيح والتعديل.

كيفما كانت الفوائد المدركة فإنه من الواضح أن برمجة الكائن سوف تكون برمجة المستقبل على الأقل لعدة عقود.

مصطلحات ومفاهيم الكائن الأساسية.

- Encapsulating - OOD - functions data members and member - Class Object and UML - OOAD

Class Object and الكائن والصنف

الصنف هو الوحدة الأساسية للكائنات، فعن طريق الصنف ممكن أن تنشي عدة كائنات.

بمعنى آخر ، يمكن أن نمثل الصنف بسيارات النقل الخاص ونحن نعرف أن سيارات النقل الخاص تشترك في صفات معينه مثل احتوائها لأربعة اطارات ومحرك وتقوم بمهام معينه مثل التحرك للأمام والخلف والفرمله.

اذا الصنف هو السيارات وهو الوحدة الأساسية ، وعن طريق هذا الصنف ممكن ان ننشيء كائنات (سيارات) من نوع مرسيدس ، bmw أو Lexus ، وبذلك يكون لدينا صنف موحد وقد انشأنا منه عدة كائنات لها نفس الخصائص والمهام.

الذي يتم التعامل معه برمجيا هو الكائن وليس الصنف، لأن الصنف هو المخطط الأساسي للكائن، ومن هذه الناحية يمكن تمثيل الصنف بمخطط فيلا على ورق والكائن هو الفيلا على ارض الواقع. اذا لا يمكن عمل طبخة معينه في المطبخ في مخطط الفليلا على الورق ولا يمكن النوم في غرفة النوم على الورق ، فقط يمكنك ذلك في مبنى الفيلا على ارض الواقع.

أعضاء البيانات ودوال الأعضاء functions datat members and member

اعضاء البيانات أو الخصائص (attributes) ، على سبيل المثال ، خصائص المرسيدس أن لها لون واربع اطارات، اما دوال الأعضاء أو العمليات ، ويمكن تسميتها أيضا بالسلوك (behaviors) ، هي الإجراءات التي يقوم بها الكائن، مثلا في السيارة ، فالسلوك الذي تقوم به السيارة هو الحركة للخلف والأمام والفرمله.

تصميم الكائن الموجه (Object-Oriented Design (OOD

وهي عملية تحديد خصائص وسلوك كل كائن عند انشاء الصنف.

الكبسلة Encapsulating

الكبسلة أو اخفاء المعلومات ، أي اخفاء خصائص وسلوك الكائنات فيما بينها، بحيث يمكن أن يتصل كل كائن بآخر دون معرفة التفاصيل. فأنت عندما تقود السيارة فإنك تقوم بحركات معينة دون معرفة التفاصيل ، فعند الضغط على دواسة البنزين فإنك سوف تزيد السرعة وليس من الضروري معرفة تفاصيل وكيفية حدوث ذلك.

تحليل وتصميم (OOAD) Object_Oriented Analysis and Design الكائن الموجه

عرفنا فيما سبق ماذا يعني تصميم الكائن. أما عملية التحليل فهي تستخدم مع كائنات انظمة التشغيل الضخمة، كبرنامج آلة صراف اتوماتيكية لبنك كبير. فعند ذلك يلزمك أن تقوم بعملية تحليل مفصلة لهذا الكائن بما يناسب متطلبات هذا البنك بحيث يمكنك هذا التحليل من اكتشاف الأخطاء عند صيانة وتطوير النظام بحيث توفر الوقت والجهد

Unified Modeling Language (UML) لغة الأنماط الموحدة.

عندما تقوم بإنشاء كائن في نظام برمجي معين، فإنه من المستحسن عمل تصميم لتبسيط هذا الكائن. الـ UML وسيلة رسومية موحدة عالمية لتسهيل تمثيل الكائنات والأنظمة. وما يميز هذه الوسيلة مرونتها عن التصحيح أو احداث أي تغيير بحيث تتمدد وتتقلص عن اضافة أو حذف بعض الخصائص.

انتهت المحاضرة.

أرجو من الجميع عدم التساهل في أهمية المقدمة ، فهذه المقدمات هي من أهم المحاضرات الذي يبنى عليها القادم.

في جميع المحاضرات سوف نقوم بوضع تمارين في نهاية كل محاضرة.
وتنقسم التمارين لقسمين ، قسم الواجب

وقسم اسئلة للتحدي.

وبما أن هذه المحاضرة لا يوجد بها سوى النضري ، فليس هناك اسئلة تحدي ، بل اسئلة واجب فقط.

أسئلة الواجب

١- اكتب بإستفاضة بقدر ما تريد من السطور عن سبب التركيز في لغات البرمجة الحديثة بوجه عام

ولغة الـ C++ بوجه خاص على برمجة الكائن الموجه؟

٢- ربما يكون في يدك ساعة معصم ، اشرح كيف يمكن تطبيق المصطلحات التالية على فكرة الساعة :

الكائن (Object) الخصائص (attributes) ، الصنف (class) ، الكبسلة (encapsulating) ، دوال العضو (functions member) ، أعضاء البيانات (data members)

تم تحميل الدرس من شبكة المنهل التعليمية
http://111000.net

إعتبر الساعة ساعة منبه ..

امنياتي الطيبة

للمراسلة
w@111000.net