جامعة طرابلس كلية تقنية المعلومات

البرمجة الشيئية

Object Oriented Programming (with Java)

ITGS211

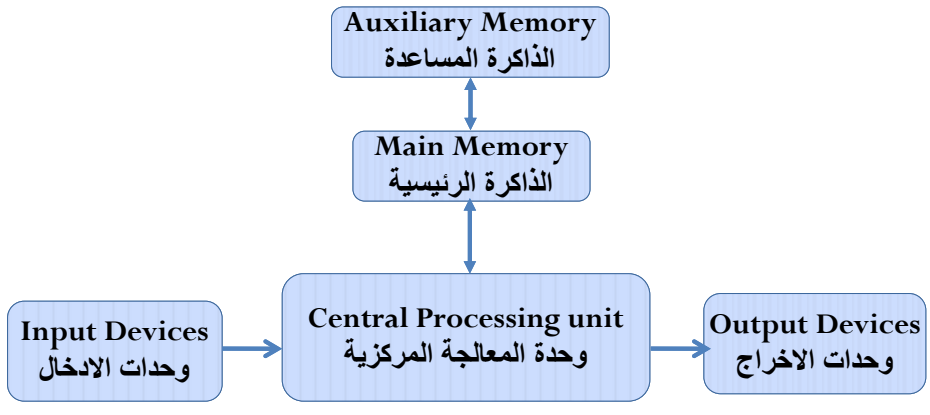
المحاضرة الأولى

أ/ ناهد فتحي فرح

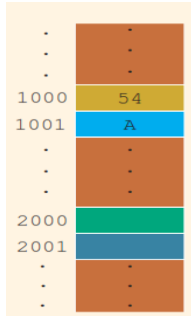
القصل الدراسي: خريف 2021/2022

مقدمة عن تنظيم الحاسوب Computer Organization

الحاسوب أساساً من خمسة عناصر أساسية:



1) الذاكرة الرئيسية Main memory أو ذاكرة الوصول العشوائي (RAM) Random Access Memory:



Main Memory
الذاكرة الرئيسية

- بما أن الذاكرة الرئيسية أو ذكراة الوصول العشوائي تتصل مباشرة بوحدة المعالجة المركزية CPU فإنها تعتبر مخزنا للبرامج قيد التنفيذ والبيانات اثناء المعالجة.
ويجب أن تُحمل لها كل البرامج قبل أن يتم تنفيذها. كذلك كل البيانات data قبل أن يقوم البرنامج بمعالجتها.
- الذاكرة مقسمة إلى خلايا مرتبة بشكل تسلسلي كما موضح بالشكل، وكل خلية لها عنوان فريد يساعد للوصول إلى البيانات المخزنة داخلها.
- عند قفل الحاسوب فإن هذه الذاكرة تنفقذ كل محتوياتها.

2) الذاكرة المساعدة Auxiliary أوالثانوية Secondary:

وهي مكان لحفظ البيانات والبرامج بصورة دائمة للأستخدام المستقبلي
ومنها: Hard disks وflash memory وCD-ROMs ....إلخ

3) أجهزة الادخال input devices:

هي المعدات المستخدمة لادخال البرامج والبيانات والأوامر للحاسوب مثل لوحة المفاتيح، الفارة و... غيرها.

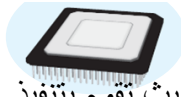
Input Devices
وحدات الادخال

4) أجهزة الاخراج Output devices:

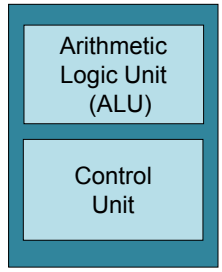
هي معدات تستخدم لاظهار النتائج ومراقبة عمل الحاسوب كالشاشات والطابعات و الراسمات و...غيرها.

Output Devices
وحدات الاخراج

5) وحدة المعالجة المركزية Central Processing Unit (CPU)



- هي أسرع وأغلى قطعة بالحاسوب، فهي لب الحاسوب حيث تقوم بتنفيذ البرامج ومعالجة البيانات المدخلة من أجهزة الادخال واظهار النتائج على وحدات الاخراج أو تخزينها فى الذاكرة المساعدة.
- هي المتحكمة في سير المعلومات وانسيابها بين الوحدات المختلفة. وتتكون من:
  - وحدة الحساب والمنطق (ALU)
  - وحدة التحكم Control Unit



وحدة المعالجة المركزية
Central Processing unit

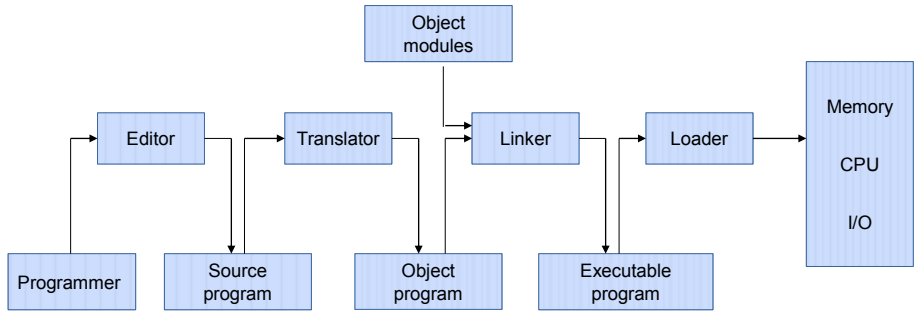
مقدمة عن برمجة الحاسوب Computer Programming

البرنامج Program: هو الخطة أو الطريقة التي يتبعها الحاسوب لحل المسألة مكتوباً بلغة البرمجة المناسبة.

لغة البرمجة Programming language:

عبارة عن مجموعة من الأوامر، تكتب وفق مجموعة من القواعد تحدد بواسطة لغة البرمجة ، وهذه الأوامر تمر بعدة مراحل (كما في الشكل التالي) إلى ان تنفذ على جهاز الحاسوب.
مع ملاحظة أن لكل لغة خصائصها التي تميزها عن الأخرى وتجعلها مناسبة بدرجات متفاوتة لكل نوع من أنواع البرامج والمهمة المطلوبة من هذا البرنامج. كما للغات البرمجة خصائص مشتركة وحدود مشتركة بحكم أنها كلها صممت للتعامل مع الحاسوب.

البرنامج يمر بعدة مراحل لتطويره وترجمته وتنفيذه باستخدام عدة برامج للنظام.



تطور برمجة الحاسوب

تطورت برمجة الحاسوب على عدة مراحل:

1) لغة الآلة **Machine language** : حيث تُكتب شفرات التعليمات وتُمثل البيانات والمعاملات وتُحدد العناوين بالرقمين 0و1؛ مما جعل البرامج صعبة الفهم والتتبع. علاوة على وجوب تعرف المبرمج على معمارية (Architecture) الآلة المستخدمة أولاً. فالبرنامج مرتبط بالآلة نفسها ولايمكن تنفيذه على آلة أخرى.

2) لغة التجميع **Assembly Language**: تم فيها استخدام رموز مختصرة للتعليمات ولتمثيل البيانات والعناوين. ولكن الحاسوب لا يستطيع تفسير وتنفيذ تلك الرموز مباشرة. لذلك يتم ترجمتها الى شفرات لغة الآلة باستخدام برنامج المجمع Assembler

Assembly Language	Machine Language
LOAD	100100
STOR	100010
MULT	100110
ADD	100101

3) لغات عالية المستوى **High Level Languages**: فيها كُتبت التعليمات على شكل جمل قريبة جدا من مفهوم الانسان. وكل لغة تتميز عن اللغة الاخرى من حيث:

- جمل اللغة Language statements
- الكلمات المفتاحية أوالمحجوزة Key or reserved words
- الشكل العام للبرنامج وتنظيمه General program structure
- الآساليب البرمجية للغة Language features and techniques

لعل أشهر هذه اللغات:

- BASIC
- COBOL
- Pascal
- C
- C++
- Java

لازالت الآلة لا تفهم الا لغتها (0 و1). لهذا وجب وجود مترجمات للغات عالية المستوى. كل لغة لها مترجمها الخاص بها ولكل نوع من الحواسيب مترجم خاص به لتلك اللغة.

المُترجمات

المترجم يقوم بترجمة البرنامج المصدري المكتوب بلغة البرمجة عالية المستوى إما إلى **لغة الآلة المستخدمة مباشرة** أو الى **لغة التجميع أولا ثم يقوم بترجمة لغة التجميع الناتجة الى لغة الآلة المستخدمة**.

وتصنف المترجمات إلى نوعين:

1) **المُترجم أو المُجمع compiler:** حيث يقوم بالمرور على البرنامج المصدري بالكامل واكتشاف أي أخطاء لغوية وتنبيه المبرمج إليها. ثم يقوم بترجمة البرنامج المصدري بالكامل الى لغة الآلة Object program.

2) **المُفسر interpreter:** حيث يقوم بالمرور على البرنامج جملة جملة وفي الاثناء يقوم بفحصها من الاخطاء اللغوية وترجمتها.

الأخطـــاء Errors

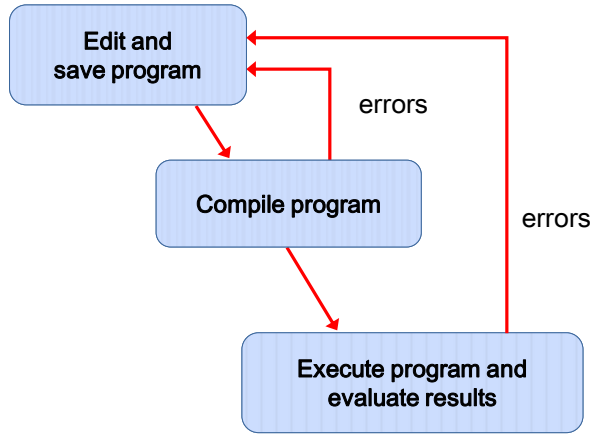
نعلم جميعاً انه أثناء إعداد أي برنامج قد تحدث أخطاء، وهي :

- **أخطاء لغوية Syntax errors:** وهي التي تحدث نتيجة مخالفة قواعد اللغة ويكتشفها البرنامج المترجم.
- **أخطاء منطقية Logical errors:** وهي التي لايمكن اكتشافها إلا بتجربة البرنامج واختباره ببعض البيانات التي نعرف أو نتوقع نتائجها.
- **أخطاء زمن التنفيذ Run_Time errors:** وهي التي تحدث أثناء تنفيذ البرنامج كالقسمة على صفر مثلاً.

ولكي يكون البرنامج صحيح يجب أن يكون خالٍ من الأخطاء اللغوية والمنطقية معاً. فالبرنامج الصحيح لغوياً ليس بالضرورة أن يكون صحيح منطقياً.

A program that is syntactically correct is not necessarily logically (semantically) correct.

الأخطـــاء Errors



اتجاهات البرمجة:

رغم التطور في البرمجة ووصولها من مستوى مفهوم الآلة إلى مستوى مفهوم الانسان، إلا أنها أخذت آساليب واتجاهات جديدة، ولازالت.
ولعل أهم الأسباب **Reasons** التي أدت إلى ذلك هي:

- تطور الكيان المادي للحاسوب Hardware evolution
- تطور انظمة التشغيل Operating systems evolution
- تطور التطبيقات Applications evolution
- تنظيم البيانات Data organizations: (types ، structures ، databases)
- التخاطب/التفاعل بين الآلة و المستخدم Man-Machine interaction (GUI, multimedia,…)

نماذج البرمجة Programming Paradigms:

تم تصنيف لغات البرمجة إلة عدة نماذج أهمها وأشهرها:

- البرمجة الأوامرية Imperative programming.
- البرمجة الوظائفية Functional programming.
- البرمجة المنطقية Logic programming.
- **البرمجة الشيئية Object-Oriented programming**

البرمجة الشيئية Object-Oriented programming

يشار إليها اختصاراً OOP

يطلق عليها أيضاً:

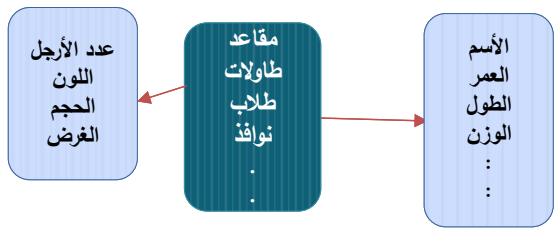
- **البرمجـة الكائنية**
- **البرمجة كائنية المنحى.**
- **البرمجـة الموجهة نحو الكائنات.**
- **البرمجة الموجهة نحو الهدف (العنصر).**

البرمجة الشيئية Object-Oriented programming

هي عبارة عن نمط برمجة متقدم، فيه يقسم البرنامج إلى وحدات كل وحدة تسمى كائن.

المفهوم الأساسي الذي جاءت منه هو أن كل شئ نراه من حولنا هو عبارة عن كائن.

انظر حولك في المدرج (أو أينما أنت)

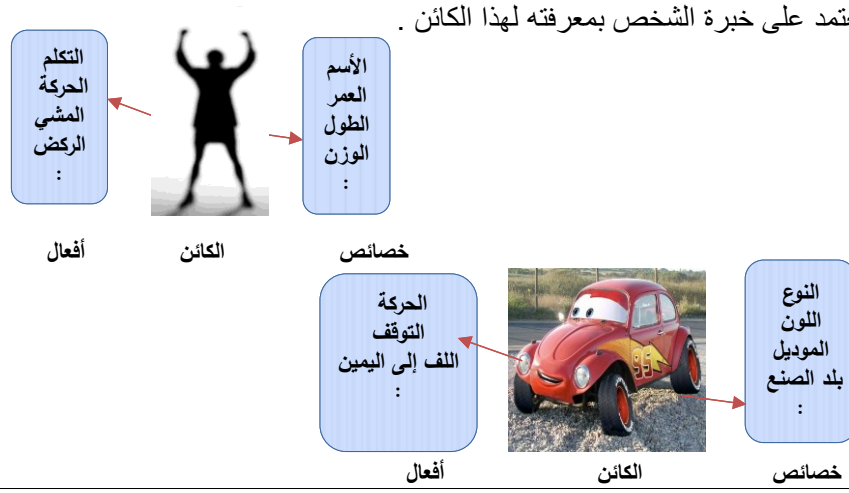


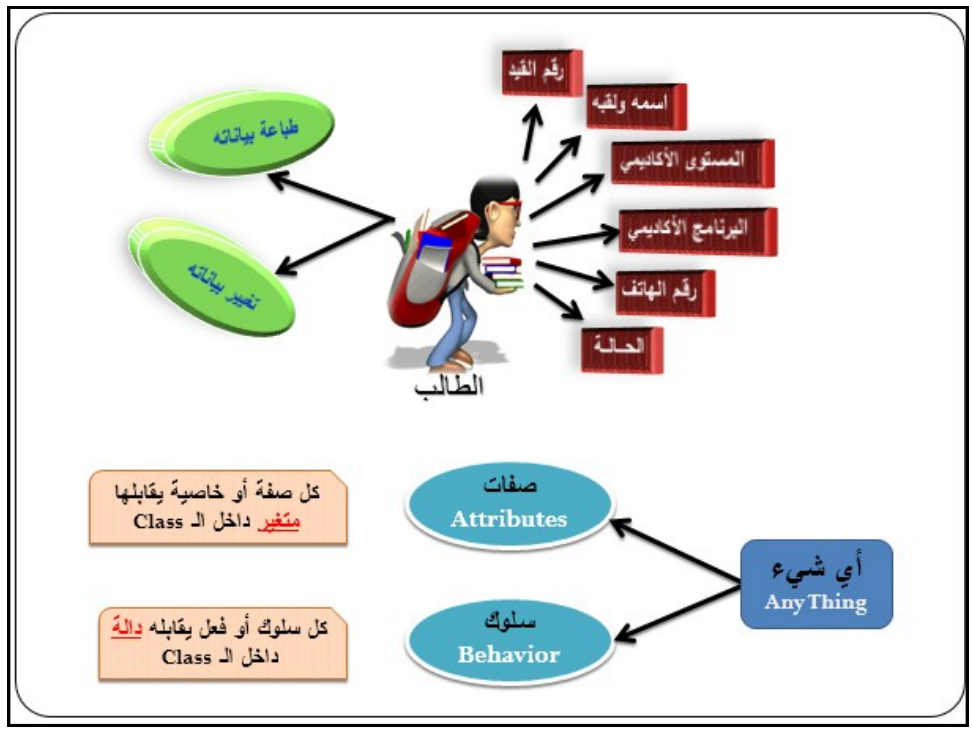
Every things is an Object

ستجد أن كل كائن له:

خصائص (Attributes، Properties): وهي الأشياء الموجودة والملازمة للكائن ولا ينفك عنها والخصائص هي التي تأتي على صيغة Name= value.

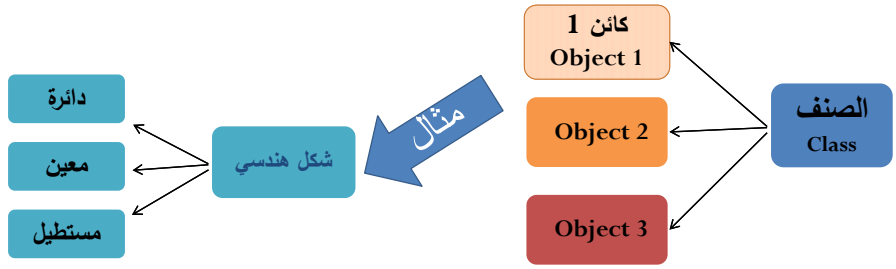
- **أفعال** (Action, Methods, Behaviour): وهي الافعال التي يقوم بها الكائن ويعتمد على خبرة الشخص بمعرفته لهذا الكائن .





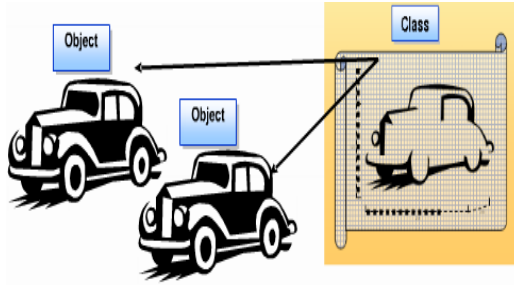
الفرق بين الـصنف Class والكائن Object

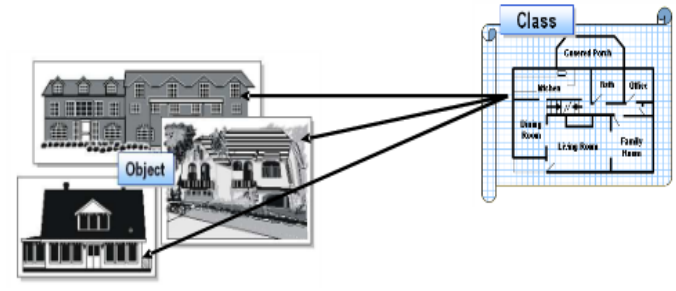
Object:له خصائص وأفعال أو سلوك.
Class: له خصائص وأفعال أو سلوك.
إذاً ما الفرق بينهما؟؟
الكائن Object هو نسخة من الصنف Class.
الـ class يتم انشاؤه مرة واحدة والـ object يكون أكثر من نسخة
ومن دون الـ class لايوجد شئ اسمه الـ object .



الكائن Object هو نسخة من الصنف Class.

لكي نتمكن من إنشاء Object لابد من انشاء الـClass أولاً.





لماذا وجد الصنف Class؟

- إذا طلبنا من كل طالب موجود بالقاعة أن يأخذ ورقة ويكتب بياناته الأساسية ويُسلمها للموظف المختص بالتسجيل، الموجود معنا. فسنجد أن كل طالب سجل البيانات الأساسية من وجهة نظره هو. فيما يلي عينة لما تم استلامه

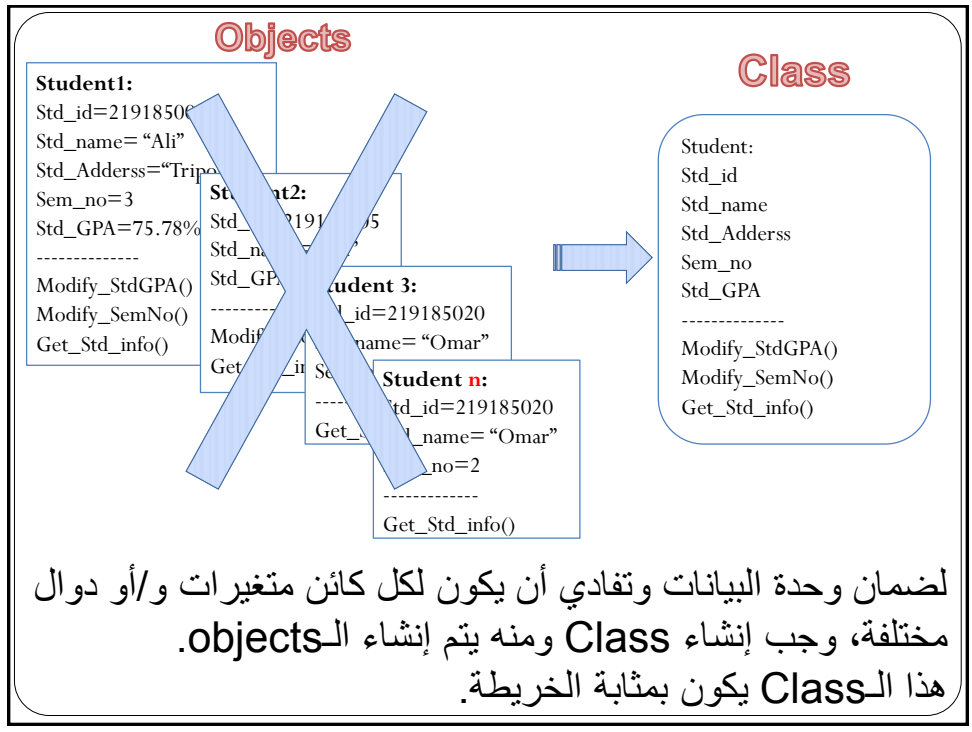
Student1:
رقم القيد: 219185001
الاسم: "Ali"
العنوان : "Tripoli"
الفصل 3
المعدل 75.78%

Student2:
رقم القيد 219185005
الاسم "Ali"
عدد المواد 5

Student 3:
رقم القيد :219185020
الاسم "Omar"
الفصل 2
الطول 1.7م

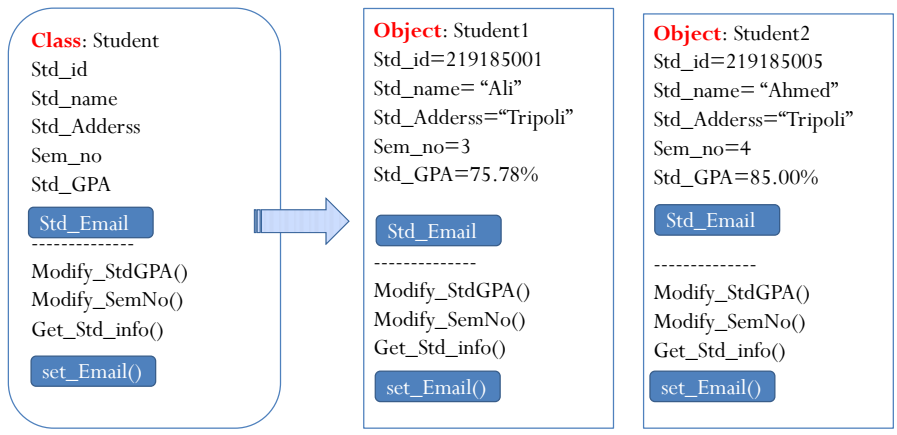
Student n:
الأسم "Omar"
الفصل 2
تاريخ الميلاد 2000

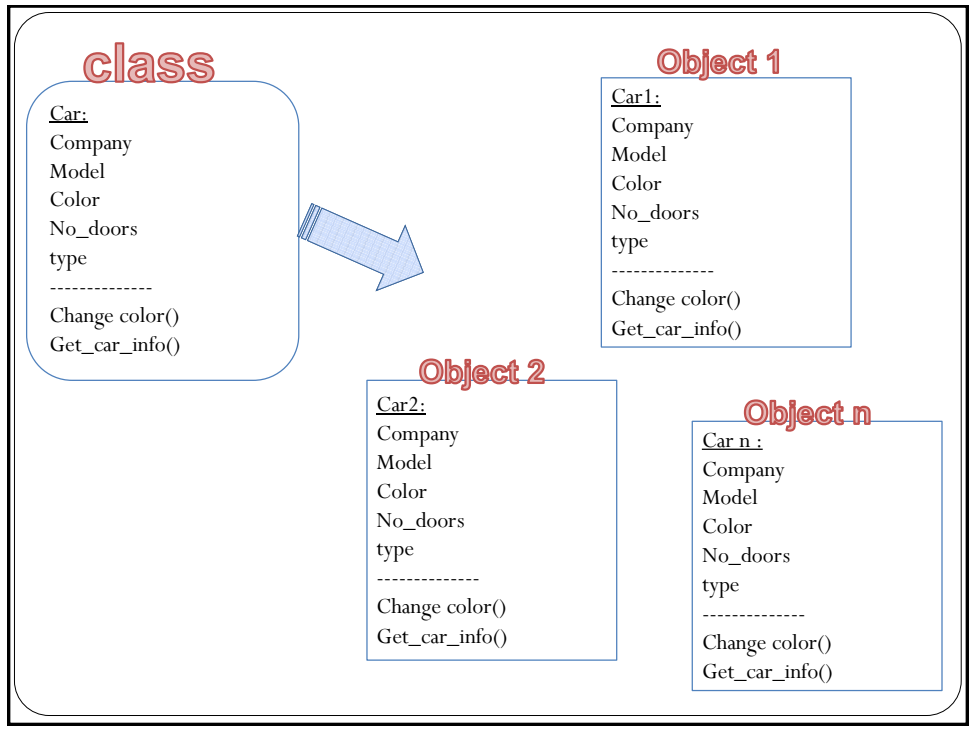
- لاحظ،، هل النماذج أعلاه تحوي نفس الحقول؟ لا
- وبالتالي سيكون لكل طالب سِجل بحقول مختلفة. من هنا وجب انشاء نموذج واحد
- ومن تم يوزع على الطلبة بحيث كل طالب يملء النموذج بالبيانات المطلوبة في النموذج لاغير.
- هذا النموذج هو الصنف Class والذي سيوحد البيانات



لضمان وحدة البيانات وتفادي أن يكون لكل كائن متغيرات و/أو دوال مختلفة، وجب إنشاء Class ومنه يتم إنشاء الـobjects.
هذا الـClass يكون بمثابة الخريطة.

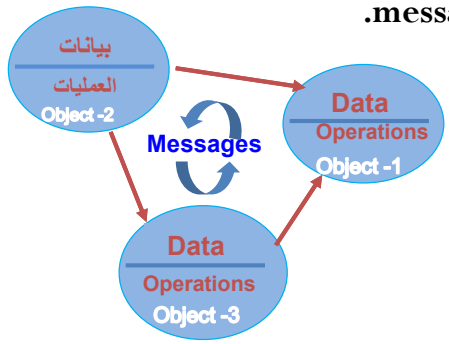
أي تعديل يحدث على الـClass ينطبق فورا على الـ objects سواء بالإضافة أو الحذف.
ففي المثال أعلاه عند إضافة المتغير Std_Email إلى الكلاس Student ستجد أنه ظهر في الـobjects المكونة منه وهي Student1 و Student2.
أيضا عند إضافة الدالة set_Email()





التفاعل بين الكائنات Objects

بما أن البرنامج مكون من مجموعة الكائنات objects، فإن هذه الكائنات تتفاعل وتتواصل فيما بينها عن طريق الرسائل messages.



الرسالةMessage : هي طلب للكائن object لاستدعاء أحد العمليات methods الخاصة به.

وتتكون الرسالة من ثلاثة مكونات:

- اسم الكائن object الذي يوجه رسالة
- اسم الـ method لتنفذ
- أي معاملات parameters تحتاجها الـ يحتاجها method.

ObjectName.method(parameter1,parameter2,…)

بعض مزايا البرمجة الشيئية OOP

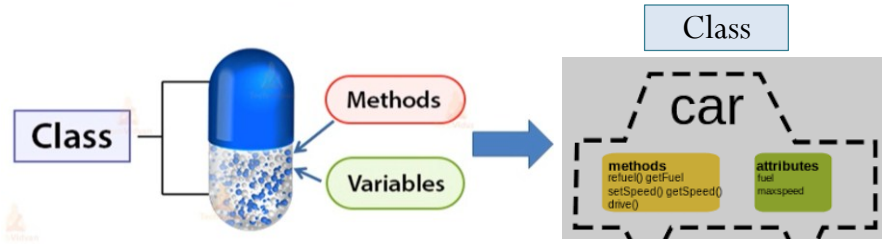
1) **البساطة**: حيث أن الكائنات تحاكي الكائنات الحقيقية، مما يقلل التعقيد وهيكل البرنامج واضح وسهل الفهم.
2) **إعادة الاستخدام:** يمكن اعادة استخدام الكائنات في عدة برامج أخرى.
3) **سهولة الصيانة:** حيث ستعرف مكان الخطأ بالتحديد فكل كائن مستقل بشكل تام.
4) **سهولة التعديل:** من السهل اجراء تغييرات في تمثيل البيانات داخل أي Class دون التأثير على أجزاء البرنامج الأخرى.
5) **التطوير**: عملية التوسع ستكون سهلة من خلال اضافة كائنات جديدة أو اجراء تعديل على كائنات موجودة من باب الاستجابة لتغيرات جديدة.

المفاهيم الأساسية في البرمجة الشيئية OOP

- Object Oriented concepts
  - Encapsulation
  - Abstraction
  - inheritance
  - polymorphism

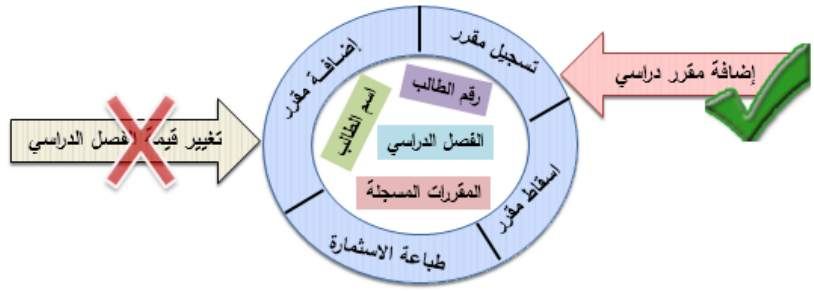
التغليف Encapsulation

- التغليف (Encapsulation): هو عملية وضع الخصائص و العمليات ضمن وحدة واحدة (Class) فيتم اخفاء البيانات والدوال و يُسمح بالوصول لهم بصلاحيات معينة. و لذلك يسمى هذا المفهوم أيضًا اخفاء البيانات (Data Hiding).



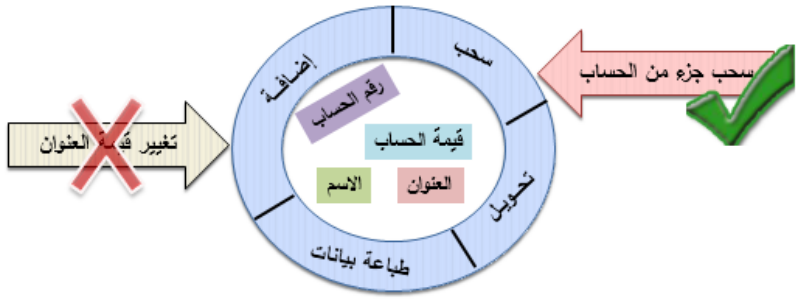
التغليف Encapsulation

- في المثال أدناه: يسمح بتنفيذ إحدى العمليات: تسجيل مقرر دراسي ، إضافة مقرر دراسي، إسقاط مقرر دراسي، وطباعة اشمارة التسجيل ولا يسمح بالوصول للبيانات وتغيير قيمها.



التغليف Encapsulation

في المثال أدناه: يسمح بتنفيذ إحدى العمليات: سحب ، اضافة، تحويل ، وطباعة البيانات. ولا يمكن أن يسمح بالوصول للبيانات وتغيير قيمها. كتغيير الاسم أو العنوان أو غيره.



مفهـــوم التجريـد Abstraction

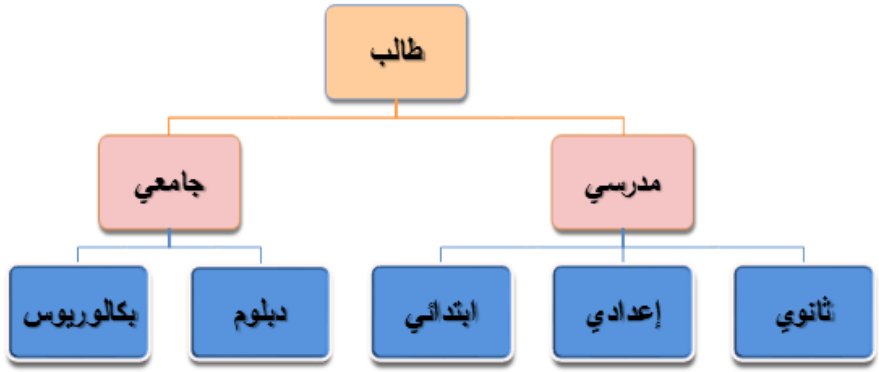
عملية التجريد تعمل على إخفاء التعقيد الناتج عن كيفية عمل الكائنات

التعقيد الذي داخل جسم الفيل أو اي حيوان آخر لا يتم التعرض له إطلاقا ولكن يتم التعامل معه كحيوان له حركات ووظائف وصفات ظاهره فقط، وكذلك الأمر لباقي الكائنات كالسيارة وغيرها.

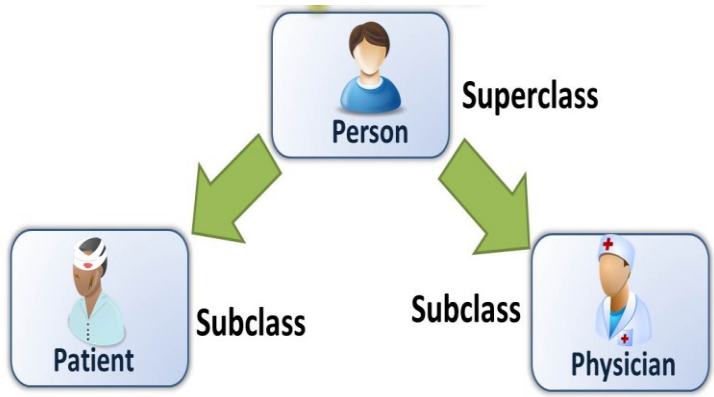
- مفهوم التجريد (Abstraction): هو عملة اخفاء طريقة تطبيق العمل داخل الـClass ، فأنت تعلم أن السيارة تسير و لكنك لا ترى كيف تنتج هذه الحركة.
- فمثلا : عند النظر للسيارة ننتبه فقط لكونها سيارة و إذا دققنا النظر ننتبه لـ :
  - اللون
  - الشكل
  - عدد الأبواب
- و لكن لا يلفت انتباهنا كيفية سيرها و طريقة وصول الوقود للمحركات و طريقة توقفها فهذا التعقيد كله لا نتعامل معه في حياتنا و البرمجة الشيئية تتمتع بهذه الميزة من خلال مفهوم Abstraction

مفهـــوم التوارث Inheritance

مفهوم التوراث هو مفهوم مشتق من علم الوراثة الموجود في الكائنات الحية، حيث أن الابن يرث صفات و مهارات معينة من أبويه و يظهر فيه صفات و مهارات أخرى، و هذا الأمر متوفر في البرمجة الشيئية مع بعض التغيير.

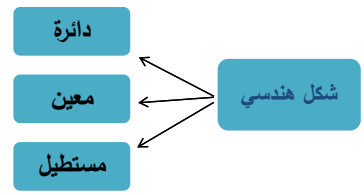


مفهـــوم التوارث Inheritance



مفهـــوم تعدد الأشكـال Polymorphism

✓ **مفهوم تعدد الأشكال (Polymorphism): هو قدرة المبرمج من خلال لغات البرمجة التي تدعم الشيئية أن يُنتج عدد من الدوال التي يتم تطبيقها بشكل مختلف لمحاكاة الواقع فمثلا نقوم بحساب مساحة الأشكال الهندسية ولكن كل مساحة يتم حسابها بشكل مختلف.**



كل هذه المخلوقات تشترك في إصدارها للأصوات ولكن كلاً له صوت مختلف.
أي أن اصدار الصوت واحد ولكن يختلف حسب الكائن.
أي يأخذ أشكال متعددة

أسئلة حول المحاضرة:

- أذكر المفاهيم الأساسية للبرمجة الشيئية، وتكلم عن كل منها بإيجاز.
- عرف كلا من الـ Class و الـObject ، وما الفرق بينهما؟