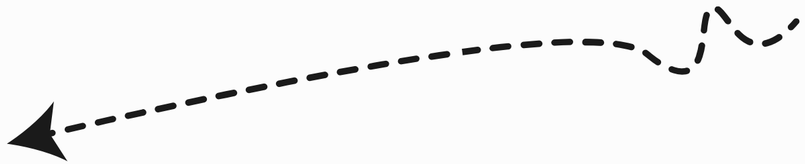
By : Shahd Tarek

FUNDAMENTALS OF OOP

IEEE COMPUTER SOCIETY
Mansoura Student Branch Chapter

ABOUT ME

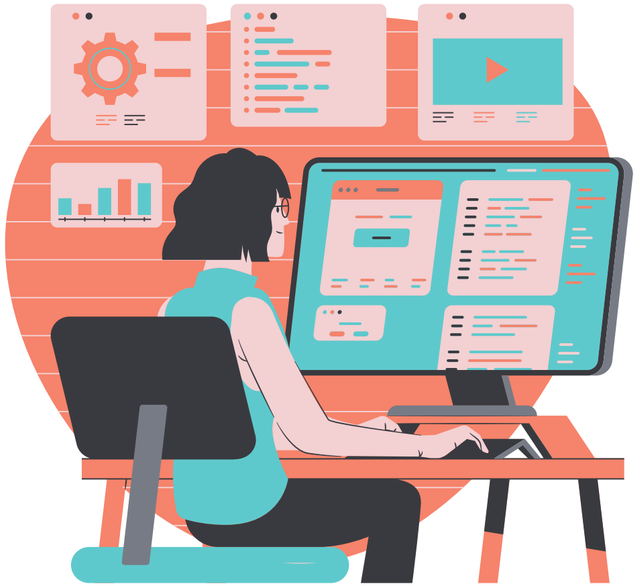
- Shahd Tarek
- BIS Student At MET Academy
- Level 2
- Linkedin: Shahd Tarek
- GitHub: shahd-tareq



IEEE COMPUTER SOCIETY
Mansoura Student Branch Chapter

Presentation content:

- Introduction in OOP
- Abstraction
- Polymorphism
- Inheritance
- Encapsulation
- Examples

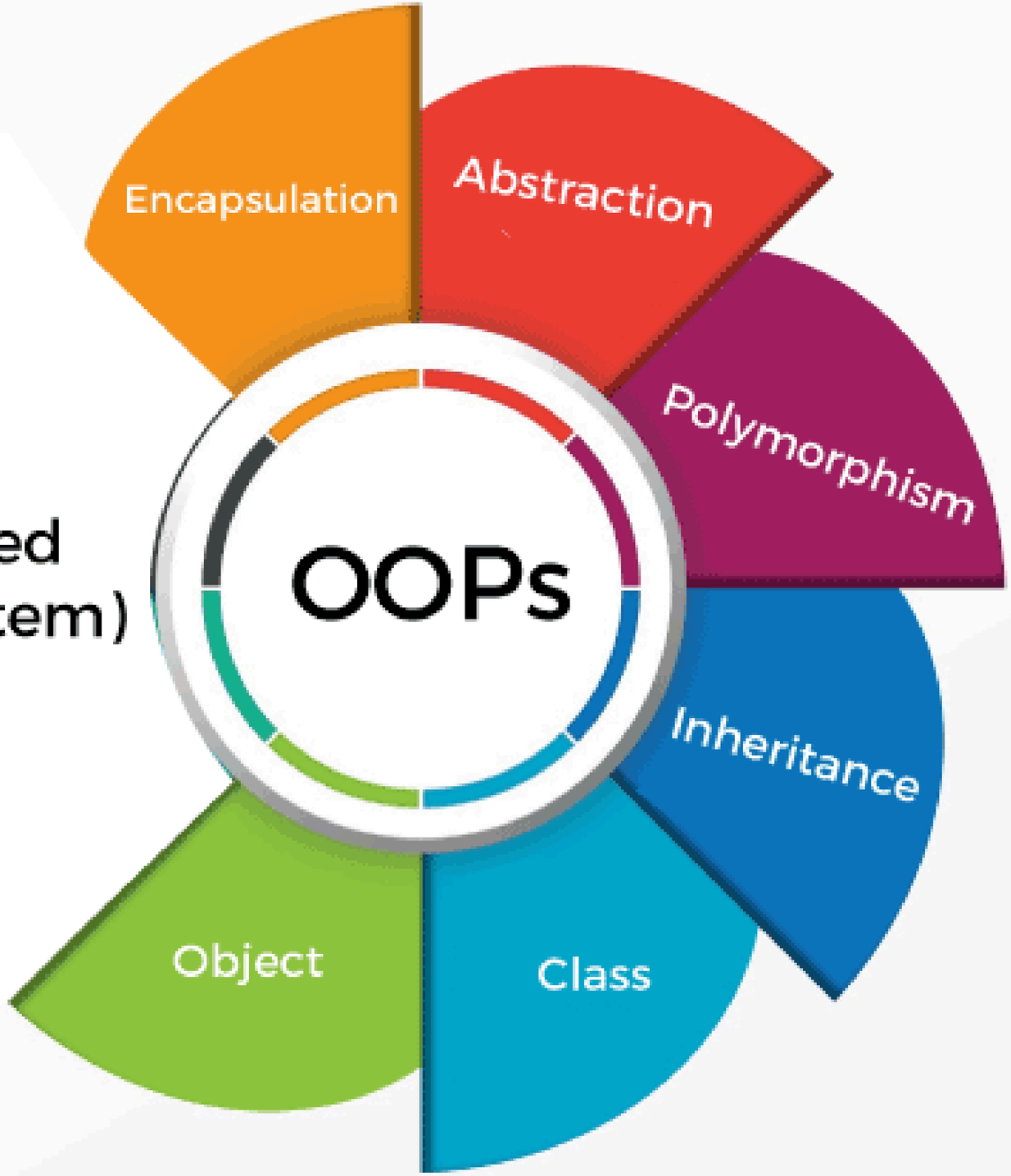


IEEE COMPUTER SOCIETY
Mansoura Student Branch Chapter

Introduction in OOP

Object-Oriented Programming (OOP) is a coding paradigm designed to organize programs in a way that makes them easier to understand and maintain. OOP is based on objects, which are units that contain data (Attributes) and functions (Methods) that define their behavior.

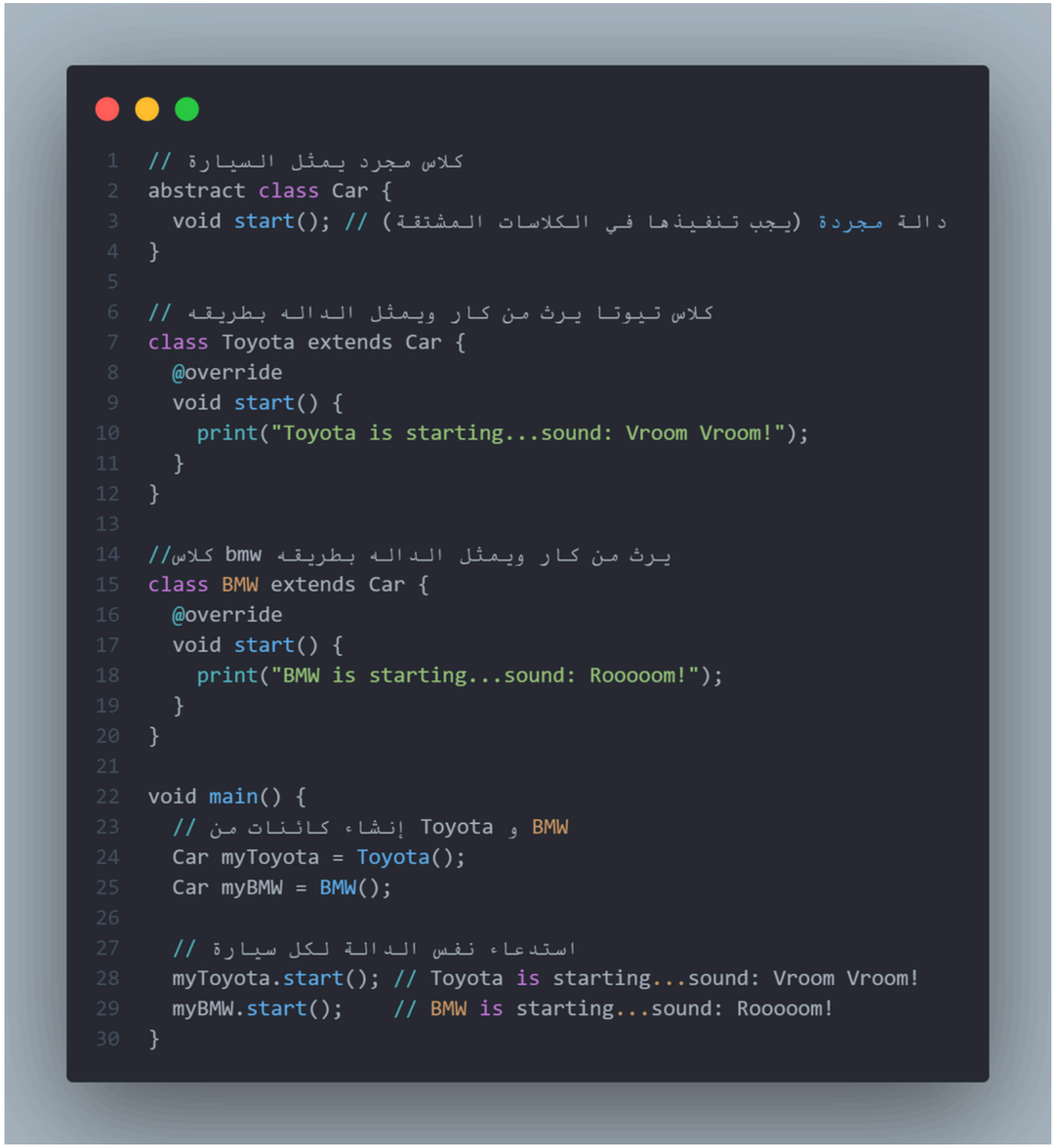
(Object - Oriented
Programming System)



1-Abstraction

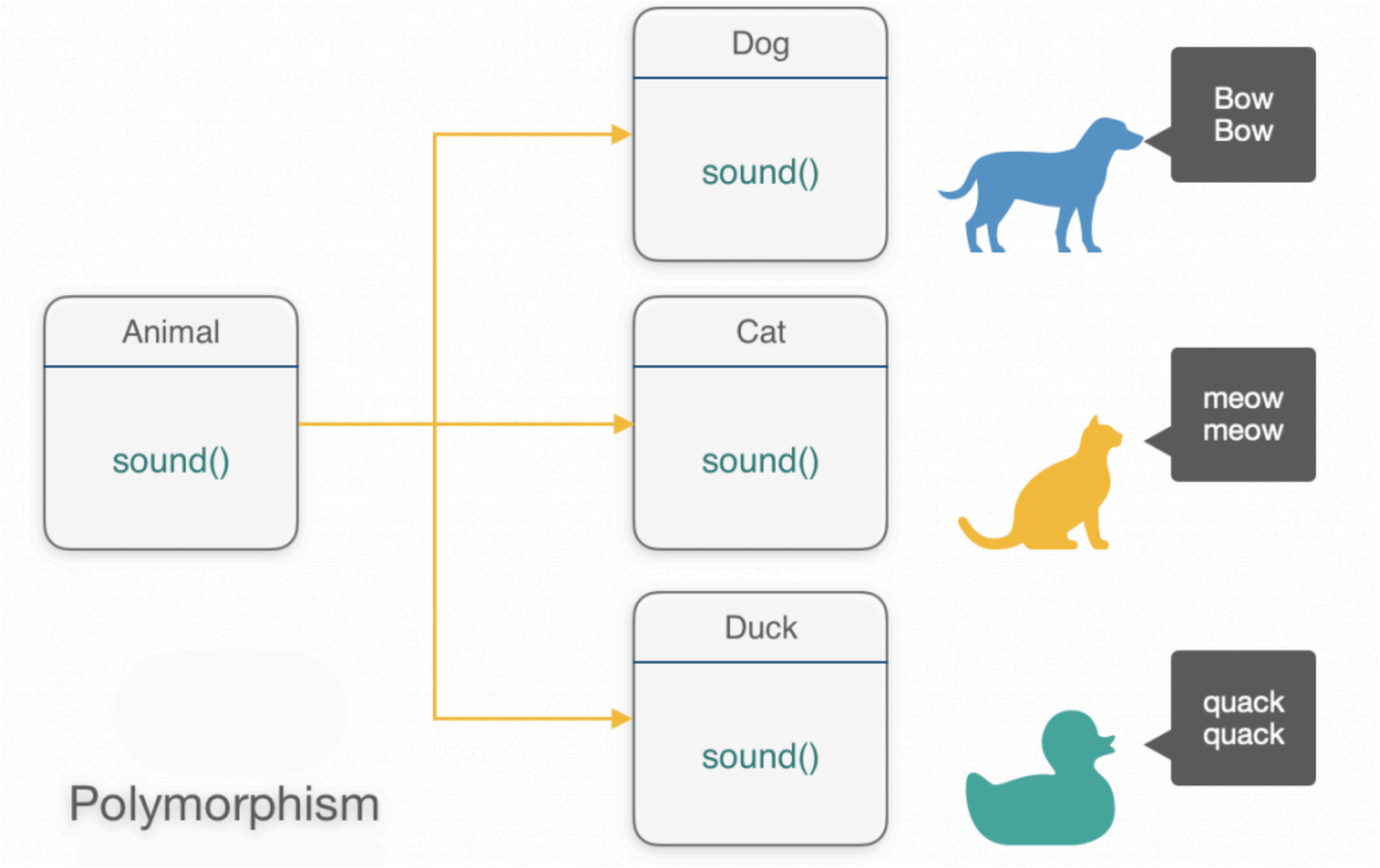
هشرحها بطريقه بسيطه اوي تخيل انت بتسوق عربيه وبتستخدم الفرامل وعداد السرعه مثلا والمحرك والدواسه انت اه بتسخدمهم بس مش مهتم تعرف هم اتعملو ازاي هو ده ال Abstraction بيخفي ما وراء الكواليس عشان يبسط التعامل مع الكود .

Example



```dart
// كلاس مجرد يمثل السيارة
abstract class Car {
  void start(); // دالة مجردة (يجب تنفيذها في الكلاسات المشتقة)
}

// كلاس تيوتا يرث من كار ويمثل الداله بطريقه
class Toyota extends Car {
  @override
  void start() {
    print("Toyota is starting...sound: Vroom Vroom!");
  }
}

//كلاس bmw يرث من كار ويمثل الداله بطريقه
class BMW extends Car {
  @override
  void start() {
    print("BMW is starting...sound: Roooooom!");
  }
}

void main() {
  // إنشاء كائنات من Toyota و BMW
  Car myToyota = Toyota();
  Car myBMW = BMW();

  // استدعاء نفس الدالة لكل سيارة
  myToyota.start(); // Toyota is starting...sound: Vroom Vroom!
  myBMW.start();    // BMW is starting...sound: Roooooom!
}
```

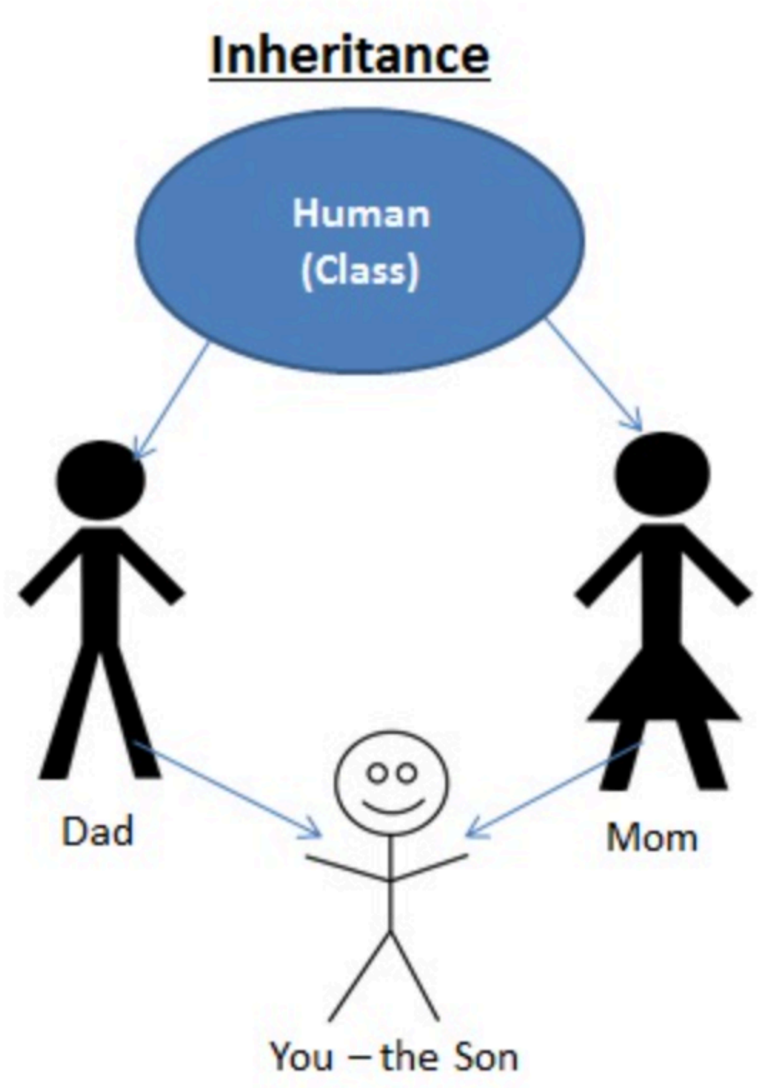
2-Polymorphism



Example

```python
class TV:
    def turn_on(ob):
        print("The TV is now ON!")

class Computer:
    def turn_on(ob):
        print("The computer is now ON!")

# Using polymorphism
devices = [TV(), Computer()]

for device in devices:
    device.turn_on()
//يعني كل نوع مختلف عن الاخر بس بيستخدمو نفس الداله  او عندهم نفس الحاجه
```

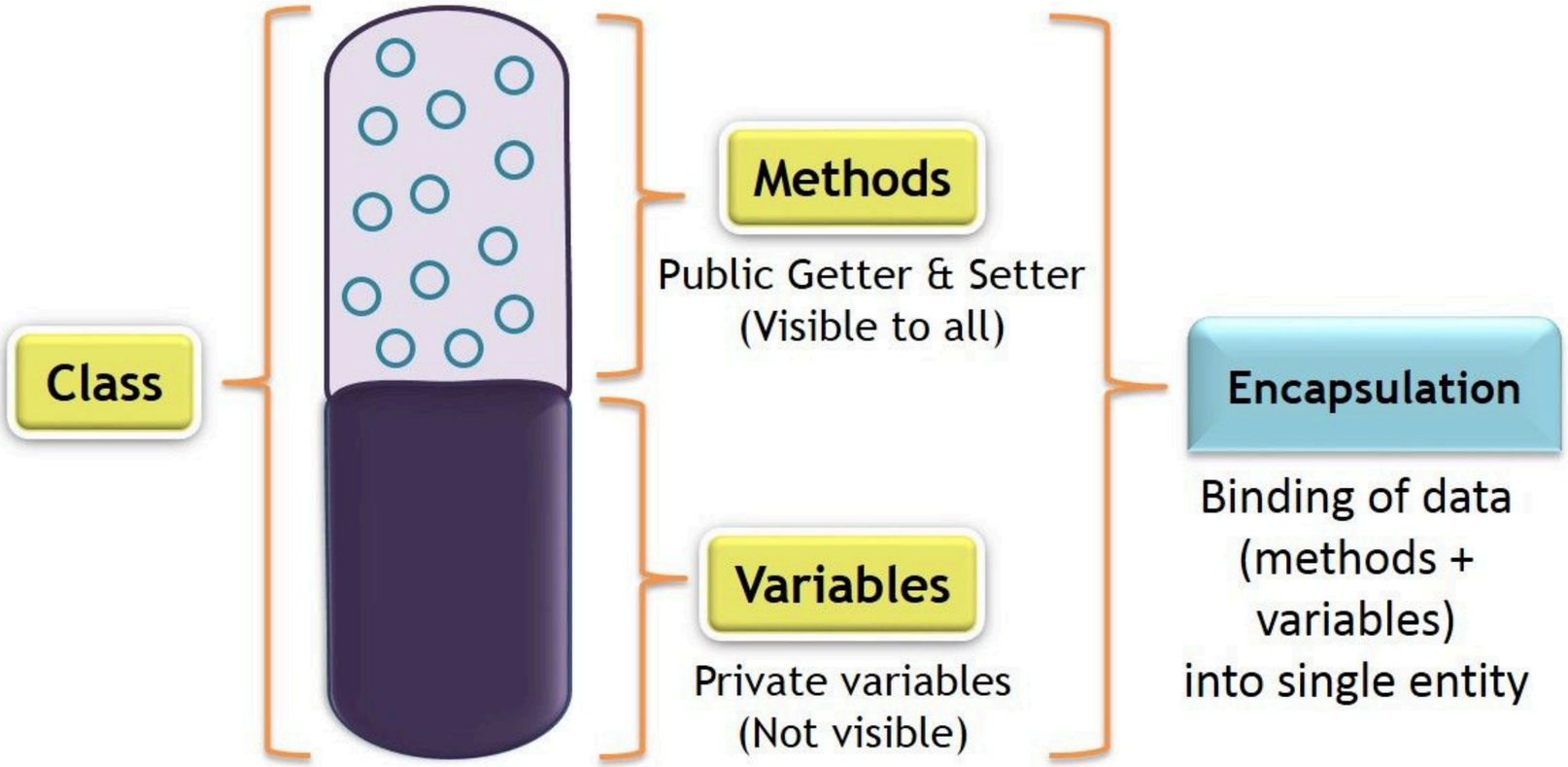
3-Inheritance



Example

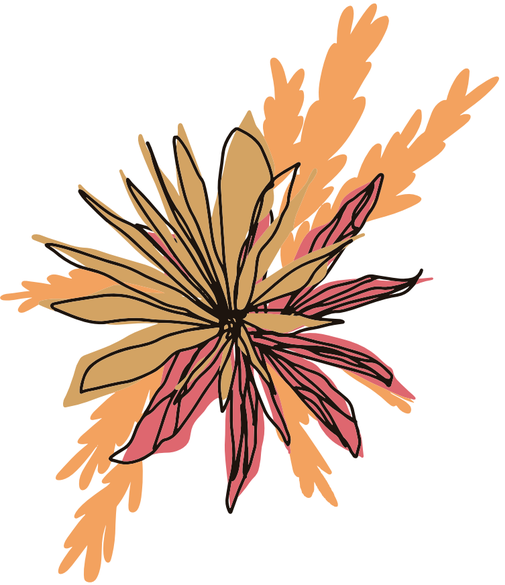
```
// كلاس الأب (Father)
class Father {
  String Name = "Ahmed";
  String job = "Engineer";

  void work() {
    print("$Name is working as a $job.");
  }
}

// كلاس الأم (Mother)
class Mother {
  String Name = "Sara";
  String hobby = "Cooking";

  void doHobby() {
    print("$motherName loves $hobby.");
  }
}

// كلاس الابن (Son) يرث من الأب والأم
class Son extends Father {
  String Name = "Omar";
  int age = 10;

  void introduce() {
    print("I am $Name,$age years old.");
    print("My father is $Name and he is a $job.");
  }
}

void main() {
  // إنشاء كائن من الابن
  Son mySon = Son();

  // استدعاء دوال الابن والأب والأم
  mySon.introduce();
  mySon.work();
}

```

4-Encapsulation



Example

```dart
class Person {
  String _name = ""; (Private)//برايفت

  // Getter لقراءة الاسم
  String get name => _name;

  // Setter لتعديل الاسم بشرط ألا يكون فارغًا
  void set name(String newName) {
    if (newName.isNotEmpty) {
      _name = newName;
    } else {
      print("Name can not be empty!");
    }
  }
}

void main() {
  Person person = Person();

  // تعيين اسم جديد
  person.name = "Ali";

  // طباعة الاسم
  print("Name: ${person.name}");

  // محاولة تعيين اسم فارغ
  person.name = "";
}
```



Thank You

IEEE COMPUTER SOCIETY
Mansoura Student Branch Chapter