**Al- Mustaqbal University**

**College of Sciences**

**Department of Cybersecurity**

**Object Oriented Programming**

**(OOP)**

 **) Object & Encapsulation & Inheritance (**

Second stage

Lecture 6/ Practical Part

2025-2024

**Lectuer : Muntather AL-Mousawi**

***Object Oriented Programming (OOP)***

**Objectالكائن**

**هو مثيل لفئة تحتوي على سمات (بيانات) وطرق (وظائف) تعمل على البيانات مثل السيارة أو الشخص .**

 **Classالفئة**

**هي مخطط لإنشاء الكائنات وهي تحدد السمات والطرق التي ستحتوي عليها كائناتها .**

**#include <iostream>**

**using namespace std;**

**class Car { // تعريف الكلاس**

**public: // القسم العام من الكلاس**

 **void startEngine() { // دالة لتشغيل المحرك**

 **cout << "Engine started!" << endl;**

 **}**

 **void drive() { // دالة لقيادة السيارة**

 **cout << "Car is driving!" << endl;**

 **}**

**};**

**int main() {**

 **Car myCar; // إنشاء كائن من النوع Car**

 **myCar.startEngine(); // استدعاء دالة startEngine من الكائن**

 **myCar.drive(); // استدعاء دالة drive من الكائن**

 **return 0;**

**}**

**Cالكلاسات والكائنات في ++ باستخدام ( OOP)هذا المثال يوضح الأساسيات في البرمجة الكائنية التوجه\***

**Encapsulation**

**(OOPالكبسلة هي واحدة من مبادئ البرمجة الكائنية (**

وهي تعني إخفاء البيانات (الخصائص) داخل الكلاس ومنع الوصول إليها مباشرة، والسماح فقط بالوصول إليها من خلال دوال محددة.

#include <iostream>

using namespace std;

**class** Person {

**private:**  // البيانات الخاصة (لا يمكن الوصول إليها مباشرة من خارج الكلاس)

 string name;

 int age;

**public:**  // دالة setter لتعيين القيم

 **void** setName(string personName) {

 name = personName;

 }

  **void** setAge(int personAge) {

 if (personAge > 0) {

 age = personAge;

 } else {

 cout << "Invalid age!" << endl;

 }

 }

 **string getName**() { // دالة getter للحصول على القيم

 return name;

 }

 **int getAge**() {

 return age;

 }

};

**int main**() {

 Person person1;

 person1.setName("Ahmed"); // تعيين البيانات باستخدام الدوال

 person1.setAge(25);

 cout << "Name: " << person1.getName() << endl; // الوصول إلى البيانات باستخدام الدوال

 cout << "Age: " << person1.getAge() << endl;

 return 0;

} Name: Ahmed

 Age: 25

**Inheritance**

 **(OOP)الوراثه هي واحدة من مفاهيم البرمجة الكائنية**

وتسمح لك بإعادة استخدام الكود بين الكلاسات. من خلال الوراثة، يمكنك إنشاء كلاس جديد (الابن) يرث الخصائص والدوال من كلاس آخر (الأب). الكلاس الابن يمكنه إضافة أو تعديل الخصائص والدوال الخاصة به بالإضافة إلى ما ورثه .

#include <iostream>

using namespace std;

class Animal { // الكلاس الأساسي الأب

 **public:**

 void eat() {

 cout << "This animal is eating." << endl;

 }

};

class Dog : public Animal { // Animal الكلاس المشتق (الابن) يرث من الكلاس

**public:**

 void bark() {

 cout << "The dog is barking." << endl;

 }

};

**int main**() {

 Dog myDog;

 myDog.eat(); // استدعاء دالة من الكلاس الأب

 myDog.bark(); // استدعاء دالة من الكلاس الابن

 return 0;