**객체 지향 프로그래밍(OOP, Objective-Oriented Programming)**

* 객체 지향 프로그래밍이란 프로그램을 어떻게 설계해야 하는지에 대한 일종의 방법론
* 기존 절차 지향 프로그래밍에서는 데이터와 처리 방법으로 나누고 프로그램의 흐름에 따라 설계를 하는 반면, 객체 지향 프로그래밍에서는 여러 객체라는 단위로 데이터와 처리 방법을 하나의 덩어리로 묶어 각 객체들이 유기적으로 상호작용 하도록 설계한다.
* 절차 지향이 모든 요리 재료를 순서에 맞게 조리하여 나오는 일품요리라면, 객체 지향은 이미 조리된 각 재료들을 필요에 따라 조합하여 먹는 구절판 같은 것.
* 작은 문제점들을 해결하는 객체들을 만들고, 이를 조합해서 큰 문제를 해결하는 상향식(Bottom-up)방법을 택함.
* 이 후에, 필요한 객체만을 재사용함으로써 개발 기간과 비용을 줄이는 데도 이점이 있음.
* 현재 C++, C#, JAVA, Objective C 등이 객체 지향 언어로 사용 중.

**객체 지향 프로그래밍의 세가지 특징**

1. **캡슐화(Encapsulation)**

* 객체 지향 방식의 기본 개념
* 계층적으로 기능(메서드)와 특성(데이터, 변수)을 분류한 뒤 클래스(Class)라는 캡슐(Capsule)에 분류된 집단 별로 집어 넣는다.
* 이러한 클래스를 실체화(Instance)하면 객체(Object)를 만들 수 있다.

1. **상속(Inheritance)**

* 이전 언어들에서 사용하던 라이브러리보다 더 논리적이고 체계적으로 기능과 데이터를 계층적으로 분류해서 사용의 편의성을 도모
* 상속을 통해 자식 클래스는 부모 클래스의 특성과 기능을 그대로 이어받고 필요한 기능 일부분만을 변경하여 재정의가 가능하다.

[출처]  
http://goo.gl/BfgOx3 객체지향-vs-절차지향-프로그래밍  
http://goo.gl/VHDH45  객체 지향 프로그래밍