Lazaron Shyta (g4g)

Object Oriented Programming

OOP is a programming paradigm based on the concept of "objects", which can contain data also known as fields or attributes and functions called methods. There are two types of methods, accessors and modifiers. With accessors we can access a given attribute, as they should be private. (Private by default on Classes). Via modifiers we can modify a given attribute.

The concepts and rules used in object-oriented programming provide these important benefits:

Encapsulation: Wrapping up(combing) of data and functions into a single unit is known as encapsulation. The data is not accessible to the outside world and only those functions which are wrapping in the class can access it. This insulation of the data from direct access by the program is called data hiding or information hiding.

Data abstraction refers to, providing only needed information to the outside world and hiding implementation details. For example, consider a class Complex with public functions as getReal() and getImag(). We may implement the class as an array of size 2 or as two variables. The advantage of abstractions is, we can change implementation at any point, users of Complex class won't be affected as out method interface remains same. Had our implementation be public, we would not have been able to change it.

Inheritance: The process by which objects of one class acquire the properties of objects of another class. It supports the concept of hierarchical classification. Inheritance provides re usability, which means that we can also add additional features to an existing class without modifying it.

Polymorphism: polymorphism means ability to take more than one form. An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation.
C++ supports operator overloading and function overloading.
Operator overloading is the process of making an operator to exhibit different behaviors in different instances is known as operator overloading.
Function overloading is using a single function name to perform different types of tasks.
Polymorphism is extensively used in implementing inheritance.

Dynamic Binding: In dynamic binding, the code to be executed in response to function call is decided at runtime. C++ has virtual functions to support this.

Message Passing: Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

One main difficulty may be when pointing out the difference between Encapsulation and Abstraction as they seem to be very similar.

Difference (stackOverflow)

Encapsulate hides variables or some implementation that may be changed so often **in a class** to prevent outsiders access it directly. They must access it via getter and setter methods.

Abstraction is used to hiding something too but in a **higher degree (class, interface)**. Clients use an abstract class (or interface) do not care about who or which it was, they just need to know what it can do.