Object-Oriented Terms

New terms are entering the mainstream information technology vocabulary. A shared definition of the most common Object-Oriented (OO) terms may help your reading of this special issue of *AIXpert*.

CDE refers to the COSE Desktop Environment from the group of vendors who formed Common Open Software Environment (COSE).

Encapsulation is the practice of making an object self-contained and hiding its internal structure. Thus, an object maintains its own data and the logic to perform operations, while exposing to other objects only what is necessary to request actions. Encapsulation leads to flexible designs, because internal structures—or even whole objects—can be modified without affecting the rest of the system.

Framework is a set of objects that work together to perform a specific task. Graphical user interface frameworks, for example, allow objects such as windows, menus, and dialogs to work together to provide an efficient way to build consistent graphical applications. There are also frameworks for business problems, such as hospital management or inventory control. Well-designed frameworks provide the general design and implementation for a specific problem and allow a developer to customize it to a particular situation.

Inheritance allows developers to create a customized object based on the implementation of a general one. For example, an account object may handle debits and credits, and produce statements. A savings account object may inherit all of those features and accrue interest on the outstanding balance. Inheritance increases productivity by providing reusable software without sacrificing flexibility.

Message is how one object requests services from another. A customer object may send the print statement message to a specific checking account object. The message identifies the method that the object will use to perform the request.

Method is an operation performed by an object. For example, an account object may have separate methods for performing a debit or a credit, or printing a statement.

Module is a loosely but widely used term that describes a piece of code that performs some function. Objects are sometimes referred to as self-contained modules to emphasize the notion of encapsulation. An object's methods are usually implemented as code modules.

Object (also referred to as a *component*) is the data and logic that represents a useful element in an application. For example, in a financial application, objects can represent account, branch, and customer. The customer object can be defined to perform certain functions, but the details of how it does them are hidden from all but the designer of the customer object itself. Objects are valuable in designing and implementing software because they hide complexity.

Object Request Broker (ORB) is the mechanism that enables objects to communicate with each other across a network. It provides services for security, registration, and object management. The Object Management Group® (OMG®), an industry consortium, has defined the Common Object Request Broker Architecture (CORBA) to specify standards that allow different ORB implementations to work together.

OpenDoc is a new vendor-neutral standard for compound documents that offers cross-platform support.

Polymorphism (literally, many shapes) is the property that allows different objects to respond to the same request. For example, savings account, checking account, and loan account objects can each handle a request-to-print statement, even though each statement may contain different information. Polymorphism reinforces the value of encapsulation in an application by reducing the amount of information we need to know about an object. This allows objects to be modified or replaced without affecting other areas of the application.

Spec 1170 is a "universal UNIX®" standard that has been recommended to X/Open™ for approval, but is not yet approved. It was designed by an ad hoc committee of different companies in the UNIX arena. The "1170" refers to the popular notion that it contained 1,170 different programming interfaces (but it actually contains more).

System Object Model (SOM), the cornerstone for IBM's offerings for the emerging multiplatform, distributed object computing environment, is available today in IBM's SOMObjects Developer Toolkit. SOM defines an infrastructure for sharing objects. It allows objects to be packaged in a way that exposes only their interface.