Explain Object-Oriented Programming Terms

IS IT THE INLET MANIFOLD or the exhaust? How many components to the peripheral nervous system exist? Does the unincorporated association have its bylaws in place? Does the building follow OSHA guidelines? Was it the bicuspid or the molar? What are we talking about? Each profession or field of study has its own unique language. Before you can be an expert in any area, you must learn the specific language or terms used in the field.

Objective:

Define common terms used in object-oriented programming.

Key Terms:

abstraction	encapsulation	object
attributes	field	OOP
behavior	inheritance	polymorphism
class	instance	superclass
class definition	method	

Understanding Basic Terminology

Each field of study has its own special terminology. The basics of any field begin with knowing and understanding its unique lexicon. Video game programming is no different. If a teacher asked you to write a list of possible attributes and behaviors for a class called DeckOfCards, what would you write? Stumped? It is impossible to answer this question if you do not fully comprehend what you are being asked. You have to begin by understanding the question—one word or term at a time. Once you know what the terms attributes, behaviors, and class mean, you can properly answer the question. Your object-oriented programming vocabulary is the foundation for your future programming knowledge.

COMMON TERMS AND DEFINITIONS

Several terms and definitions are the key to basic object-oriented programming:

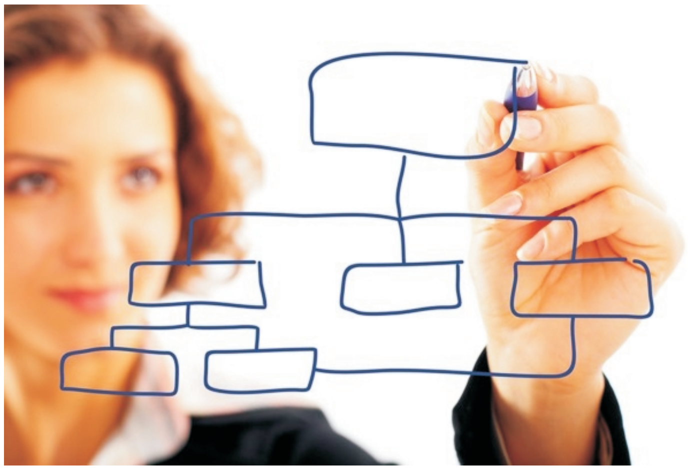
FIGURE 1. Visual maps can assist with understanding object-oriented programming.

- **Abstraction** is the process of extracting only the methods and/or attributes required by the current task, thereby ignoring all nonessential details.
- **Attributes** are the characteristics of an object as defined by the class to which the object belongs. A class called "automobile" would be an example. Some attributes that would be associated with any automobile might be color, make, and model. Each object created as a member of the class would have specific values stored in these attributes (e.g., red color) that set it apart from other member objects.
- **Behavior** is a description of something that an object can do or have done to it. The term is synonymous with method.
- **Class** is a description of the data attributes and methods for every object created as a member of the class. A class can be thought of as a single blueprint from which many objects can be created. From one house blueprint, many varying and specific houses can be built. However, that does not mean that every house will look exactly the same because a blueprint acts as an outline in that it does not provide details about items such as flooring types and light fixtures. The blueprint specifies only that there will be flooring and light fixtures. One house may have hardwood flooring in the kitchen, whereas another may have tile or linoleum. Yet both were created using the same blueprint.
    - A class named Players would provide a way to describe attributes common to all players in the game. Perhaps each has a weapon, a name, a score, and so forth. When a new

FURTHER EXPLORATION...

ONLINE CONNECTION: Object-Oriented Programming

Click and review the video on object-oriented programming located at http://www.youtube.com/watch?v=c5kfCH50wl0. Take notes while you listen. Then create your own presentation with emphasis on the terms presented in this E-unit. Create a script and visual cues or drawings to support your presentation. Make sure to label your drawings, maps, and/or diagrams with the correct terms.

player object is created using the class Players, the new object has all the attribute placeholders that are common to every player in the game. The specific values for those attributes make each player different.

- The class Players would also contain methods that describe the behaviors in which players may need to participate or actions players may need to perform. All players may need to move, choose a weapon, reload a weapon, and so on.
- The Player class is a blueprint to create a new player object. When a new player is created using the Player class as the blueprint, the programmer has an opportunity to describe the new player by assigning values to the attributes of that particular player object. The programmer can use methods in the class to cause the specific player object to perform behaviors.

FIGURE 2. Creativity has no limits after you know the programming basics.

♦ **Class definition** is the programming statements that define the attributes and methods that can be applied to any object created as a member of the class.

♦ **Encapsulation** is an accepted best practice among object-oriented programmers in which an object's attributes may be changed only by the methods associated with the object's own class and not by methods that are part of unrelated classes.

♦ **Field** is another term used to describe an object's attribute or characteristic.

♦ **Inheritance** is the process that allows a class created as a subclass to automatically gain access to all data attributes and methods of the class from which it was created. The class from which a subclass is created is a **superclass**. A programmer can take advantage of existing class definitions and alter the subclass slightly as needed, thereby saving time and development costs.

For example, the class Player describes all generic data attributes and behavior methods of any player in the game. Perhaps the game design calls for novice players to have fewer abilities than advanced players. A parent class known as Players would be created to define all the data attributes and methods (behaviors) associated with all players. A subclass called AdvancedPlayer could be created that would inherit all the attributes and methods of the parent class. Therefore, all attributes and behaviors are common to all players.

Additional attributes and methods would be included in the AdvancedPlayer class to reflect the abilities allowed to the more experienced players. These new attributes and methods would be available to every new object created from the AdvancedPlayer class, as would all the attributes and methods inherited from the parent class: Players.

- ♦ **Instance** is a specific object created as a member of a class. The terms "object" and "instance" are used interchangeably.
- ♦ **Method** is a section of code that describes a behavior or function. In object-oriented programming, one or more methods may be included in a class when it is programmed so any object created as a member of the class is able to perform the behaviors described by the method or methods. For instance, consider a class called Student. All students need to be able to pay fees. Therefore, a method could be included in the class Student that allows every specific student created as a member of the class Student to be able to perform the task of paying fees.
- ♦ **Object** is a specific instance of a class. For example, student_num_88596 is an object representing a specific student. Whereas the class Student is the blueprint from which many students can be created, the object student_num_88596 is a specific student that has been created based on that blueprint.
- ♦ **OOP** is an abbreviation commonly used to refer to object-oriented programming.
- ♦ **Polymorphism** is the property that allows code, objects, or functions to behave differently under different circumstances.

Further Exploration…

Summary:

Your object-oriented programming vocabulary is the foundation for your future programming knowledge. Once you know the basic vocabulary (e.g., attributes, behavior, class, and encapsulation) and the principles they define, you are ready for more specifics. The terms presented in this unit are the cornerstones for object-oriented programming.

Checking Your Knowledge:

1. Define the term "instance."
2. What are the programming statements that define the attributes and methods that can be applied to any object created as a member of the class?
3. What allows a class created as a subclass to automatically have access to all data attributes and methods of the class from which it was created?
4. What term is defined as a section of code that describes a behavior or function?
5. What term describes something that an object can do or have done to it?

Expanding Your Knowledge:

To better visualize the terms and corresponding concepts presented, practice mapping the following class concepts scenario. Mark your diagrams with the terms presented in this unit.

Suppose a class called Player has been created for a military game. Two subclasses have been created as members of this class (now Player is a superclass) to reflect the differences between novice players and advanced players. Any player object created as a member of the NovicePlayer subclass will have access to all attributes and behaviors of the superclass Player and the subclass NovicePlayer. Any player object created as a member of the AdvancedPlayer subclass will have access to all of the attributes and behaviors of the superclass Player and the subclass AdvancedPlayer. In this game, all novice players are infantry, while advanced players can choose to be officers. A novice player can be attacked only if the player is in a bunker or a tank. An advanced player can be attacked in an open field as well as a bunker, but not in a tank. Advanced players can climb lookout towers and swim. Novice players cannot.

Web Links:

C++ Language Tutorial
http://www.cplusplus.com/doc/tutorial/

Object-Oriented Programming
http://en.wikipedia.org/wiki/Object-oriented_programming

Interactive Web Tutorial for Object-Oriented Programming
http://staffweb.londonmet.ac.uk/~chalkp/proj/ootutor/