Object-Oriented Programming Terms

Unit: Programming

Problem Area: Document Code

Lesson: Object-Oriented Programming Terms

Student Learning Objective. Instruction in this lesson should result in students achieving the following objective:

Define common terms used in object-oriented programming.

Resources. The following resources may be useful in teaching this lesson:

"C++ Language Tutorial," *cplusplus.com*. Accessed July 26, 2010. <http://www.cplusplus.com/doc/tutorial/>.

Farrell, Joyce. *An Object-Oriented Approach to Programming Logic and Design*, 3rd ed. South-Western Cengage Learning, 2011.

Equipment, Tools, Supplies, and Facilities

- ✓ Overhead or PowerPoint projector
- ✓ Visual(s) from accompanying master(s)
- ✓ Copies of sample test, lab sheet(s), and/or other items designed for duplication
- ✓ Materials listed on duplicated items
- ✓ Computers with printers and Internet access
- ✓ Classroom resource and reference materials

Key Terms.

The following terms are presented in this lesson (shown in bold italics):

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

Interest Approach.

Use an interest approach that will prepare the students for the lesson. Teachers often develop approaches for their unique class and student situations. A possible approach is included here.

> *Announce to the class, "To begin today's lesson, I want each student to write a list of possible attributes and behaviors for a class called DeckOfCards." Most students will react with either blank stares or questions. Do not answer student questions, but instead ask the class a question: "How difficult is it to complete this task without knowing the definition of each element (attributes, behaviors, class)?" Follow this with a classroom discussion about the importance of understanding the basic terminology associated with any task one wishes to accomplish.*

CONTENT SUMMARY AND TEACHING STRATEGIES

Objective 1: Define common terms used in object-oriented programming.

Anticipated Problem: What are the definitions of common terms used in object-oriented programming?

I. Common terms and definitions

A. ***Abstraction*** is the process of extracting only the methods and/or attributes required by the current task, thereby ignoring all nonessential details.

B. ***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, such as red color, that set it apart from other member objects.

C. ***Behavior*** is a description of something that an object can do or have done to it. The term is synonymous with *method*.

D. ***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 specific houses can be built. However, that does not mean that every house will look exactly the same, because a blueprint acts more like an outline in that it does not provide details about things 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, but both were created using the same blueprint.

1. A class named Players would provide a way to describe attributes that are common to all players in the game. Perhaps each has a weapon, a name, a score, and so forth. When a new 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.
2. The class Players would also contain methods, which describe the behaviors in which a player may need to participate or actions a player may need to perform. All players may need to move, choose a weapon, reload a weapon, and so on.
3. The Player class is used as 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 attrib-

utes of that particular player object. The programmer can also use methods in the class to cause the specific player object to perform behaviors.

E. ***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.

F. ***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.

G. ***Field*** is another term used to describe an object attribute or characteristic.

H. ***Inheritance*** is the process that 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. The class from which a subclass is created is known as a ***superclass***. A programmer can then take advantage of existing class definitions and alter the subclass slightly as needed, 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 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.

I. ***Instance*** is a specific object created as a member of a class. The terms *object* and *instance* are used interchangeably.

J. ***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 that any object created as a member of the class is able to perform the behaviors described by the method or methods. For example, 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.

K. ***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.

L. ***OOP*** is the acronym commonly used to refer to object-oriented programming.

M. ***Polymorphism*** is the property that allows code, objects, or functions to behave differently under different circumstances.

Teaching Strategy: *Use VM–A to review the concepts of classes in object-oriented programming. LS–A provides an exercise to reinforce the terms and their definitions.*

- **Review/Summary.** Use the student learning objectives to summarize the lesson. Have students explain the content associated with each objective. Student responses can be used in determining which objectives need to be reviewed or taught from a different angle. Questions at the ends of chapters in the textbook may also be used in the Review/Summary.

- **Application.** Use the included visual master(s) and lab sheet(s) to apply the information presented in the lesson.

- **Evaluation.** Evaluation should focus on student achievement of the objectives for the lesson. Various techniques can be used, such as student performance on the application activities. A sample written test is provided.

- **Answers to Sample Test:**

Part One: Matching

1. a
2. c
3. f
4. d
5. e
6. b

Part Two: Short Answer

Instance is another term for *object*, and it represents a specific object created as a member of a class.

Part Three: Completion

1. class definition
2. Inheritance
3. *Field*
4. method
5. *behavior*
6. Abstraction

Sample Test

Name _____

Object-Oriented Programming Terms

Part One: Matching

Instructions: **Match the term with the correct definition.**

a. object
b. inheritance
c. encapsulation
d. class
e. superclass
f. attributes

_____1. A specific instance of a class

_____2. A practice in which an object's attributes may be changed only by the methods associated with the object's own class

_____3. The characteristics of an object as defined by the class to which the object belongs

_____4. A description of the data attributes and methods for every object created

_____5. The class from which a subclass is created

_____6. The process that 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

Part Two: Short Answer

Instructions: **Answer the following.**

What does the term *instance* refer to?

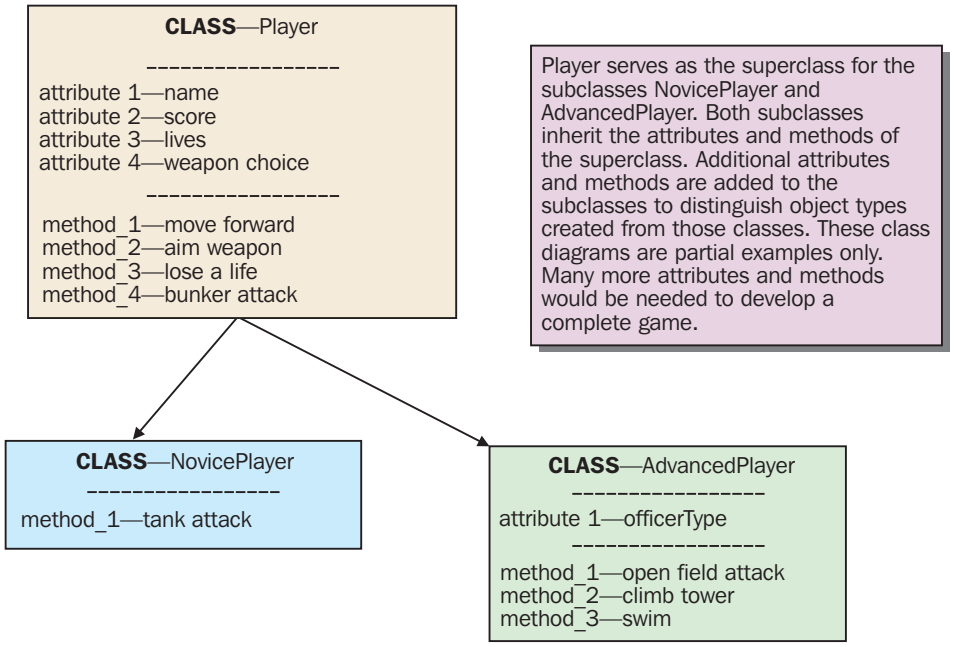
Part Three: Completion

Instructions: **Provide the word or words to complete the following statements.**

1. The programming statements that define the attributes and methods that can be applied to any object created as a member of the class are known as the _____.
2. _____ 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.
3. _____ is another term used to describe an object attribute or characteristic.
4. A _____ is a section of code that describes a behavior or function.
5. Another term for *method*, the term _____ describes something that an object can do or have done to it.
6. _____ is the process of extracting only the methods and/or attributes required by the current task, thereby ignoring all nonessential details.

OBJECT-ORIENTED PROGRAMMING— CLASS CONCEPTS EXAMPLE

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.



LS–A

Name _____

Identify Common OOP Terms

Purpose

The purpose of this activity is to become more familiar with common object-oriented programming terms.

Objective

Correctly identify the definitions of common object-oriented programming terms.

Materials

- lab sheet
- writing utensil

Procedure

In the space following each definition below, write the term or terms that best describe the definition.

1. An accepted best practice in which an object's attributes are protected because only the class of which the object is a member is allowed to change object attributes

2. A specific object created as a member of a class

3. A section of code that describes a function or action that a member object can perform

4. Programming statements that describe a class

5. The process in which a new class is created as a member of an existing class, allowing the new class to have access to all the attributes and methods associated with the existing class from which it was created

6. An upper-level class from which objects and subclasses can be made

7. The characteristics in which values used to describe an object are stored

8. The property that allows a method to behave differently because of current circumstances within the program

9. The process of using only the attributes and/or methods required to complete a task while disregarding nonessential items

10. The acronym for object-oriented programming

Identify Common OOP Terms

1. encapsulation
2. instance
3. method or behavior
4. class definition
5. inheritance
6. superclass
7. attributes or fields
8. polymorphism
9. abstraction
10. OOP