

REVIEW LESSON

MTA Course: Software Development Fundamentals

Lesson name: Software Development Fundamentals 2.3

Topic: Understand polymorphism (One 50-minute class period)

File name: SoftDevFund_RL_2.3

Lesson Objective

2.3: Understand polymorphism. *This objective may include but is not limited to:* extending the functionality in a class after inheriting from a base class; overriding methods in the derived class

Preparation Details

Prerequisite student experiences and knowledge

Students should have experience with creating and implementing methods using inheritance hierarchies. This MTA Certification Exam Review lesson is written for students who have learned about object-oriented programming. Students who do not have the prerequisite knowledge and experiences cited in the objective will find additional learning opportunities using resources such as those listed in the Microsoft® resources and Web links at the end of this review lesson.

Instructor preparation activities

None

Resources, software, and additional files needed for this lesson:

- SoftDevFund_PPT_2.3

Teaching Guide

Essential Vocabulary (C# specific):

base—a keyword used to access members of the base class from within a derived class.

new—when used as a modifier, this keyword explicitly hides a member inherited from a base class. When you hide an inherited member, the derived version of the member replaces the base-class version. Although you can hide members without the use of the `new` modifier, the result is a warning. If you use `new` to hide a member explicitly, it suppresses this warning and documents the fact that the derived version is intended as a replacement.

override—a modifier required to extend or modify the abstract or virtual implementation of an inherited method, property, indexer, or event.

polymorphism—an object-oriented programming (OOP) feature that allows methods in a base class to be redefined in derived classes. (a class that inherited its data structures and methods from another class). In polymorphism, a class can be used as more than one type; it can be used as its own type, any base types, or any interface type if it implements interfaces.

sealed—a modifier that can be applied to classes, instance methods, and properties. A `sealed` class cannot be inherited. A `sealed` method overrides a method in a base class but itself cannot be overridden further in any derived class. When applied to a method or property, the `sealed` modifier must always be used with an `override`.

virtual—a keyword used to modify a method or property declaration, in which case the method or the property is called a virtual member. The virtual member allows its implementation to be replaced within derived classes.

Lesson Sequence

Activating prior knowledge/lesson staging (5 minutes)

1. Show the Activator slide in the Microsoft PowerPoint® presentation for this lesson.
 - a. Students will determine the output to the polymorphism code example.
 - b. Define *polymorphism* and explain method overriding.

Lesson activity (35 minutes)

1. Show the PowerPoint presentation.
 - a. Describe polymorphism and what it allows a programmer to do.

- b. Show how to use the `new` keyword to create a new definition of a method, field, or property.
- c. Show how to use the `virtual` and `override` keywords in conjunction with one another.
- d. Explain how no matter how many classes have been declared between the class that originally declared the virtual member, virtual members remain virtual.
- e. Show how to use the `sealed` keyword to stop virtual inheritance.
- f. Show how to use the `base` keyword to access the method of the base class even though the derived class has replaced or overridden the method.

Assessment/lesson reflection (10 minutes)

1. Show the Lesson Review slide in the PowerPoint presentation.
2. Students should be given some time to write and share their answers with a peer. Call on students to share their answers with the class.

Microsoft resources and Web links

Polymorphism (C# Programming Guide)

<http://msdn.microsoft.com/en-us/library/ms173152%28VS.80%29.aspx>

Knowing When to Use `Override` and `New` Keywords

<http://msdn.microsoft.com/en-us/library/ms173153%28VS.80%29.aspx>

Suggested best practices:

- Students may have trouble understanding the difference between the use of the keywords `new` and `override`, especially if they are most familiar with Java. The above resources will clarify these concepts.

Additional notes to the instructor:

- This lesson demonstrates the concepts with the language C#.
- An extended lesson reflection replaces a student activity.