

AP COMPUTER SCIENCE A

UNIT 9

Inheritance



5–10%
AP EXAM WEIGHTING



~14–15
CLASS PERIODS

The icon consists of a white circle containing a blue square with the letters 'AP' in white. Below the square is a small blue monitor-like shape with two vertical lines representing a stand.

Remember to go to [AP Classroom](#) to assign students the online **Personal Progress Check** for this unit.

Whether assigned as homework or completed in class, the **Personal Progress Check** provides each student with immediate feedback related to this unit's topics and skills.

Personal Progress Check 9

Multiple-choice: ~15 questions

Free-response: 2 questions

- Class
- Class: partial

Inheritance



Developing Understanding

BIG IDEA 1

Modularity **MOD**

- How might the use of inheritance help in writing a program that simulates crops being grown in a virtual world?
- How does inheritance make programs more versatile?

Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through *inheritance*. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. These newly created classes are called *subclasses*. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a *superclass* and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Building Computational Thinking Practices

1.A 1.C 3.A 3.B 5.A 5.B 5.D

Students can design hierarchies by listing the attributes and behaviors for each object and pulling common elements into a superclass, leaving unique attributes and behaviors in the subclass. By creating a hierarchy system, students only need to write common program code one time, reducing potential errors and implementation time. This also allows for changes to be made more easily, because they can be made at the superclass level in the hierarchy and automatically apply to subclasses.

During the development of a program, programmers often use comments to describe the behavior of a given segment of program code and to describe the initial conditions that are used. Students who develop the skill of explaining why program code does not work, such as methods being overloaded improperly or superclass objects attempting to call subclass methods, are much better equipped to foresee and avoid these hierarchy issues.

Preparing for the AP Exam

One question on the free-response section of the exam will require students to write a class. This class could be part of an inheritance relationship. When overriding superclass methods in a subclass, method signatures must be the same. This includes the number, type, and order of any parameters of the overridden method. It is important for students to recognize when a method should be overridden, as well as when they can and should use methods from the superclass. Students who duplicate code unnecessarily will not earn full points on this question.

Students will be asked to analyze program code that uses inheritance. In many cases, students struggle with determining whether a method is available to be called by an object of a class. When a method is called on a subclass object, the method that is executed is determined during run-time. If the subclass does not contain the called method, the superclass method will automatically be executed.

UNIT AT A GLANCE

Enduring Understanding	Topic	Suggested Skills	Class Periods
			~14–15 CLASS PERIODS
MOD-3	9.1 Creating Superclasses and Subclasses	<p>1.A Determine an appropriate program design to solve a problem or accomplish a task (<i>not assessed</i>).</p> <p>3.B Write program code to define a new type by creating a class.</p>	
	9.2 Writing Constructors for Subclasses	<p>3.B Write program code to define a new type by creating a class.</p> <p>5.A Describe the behavior of a given segment of program code.</p>	
	9.3 Overriding Methods	<p>3.B Write program code to define a new type by creating a class.</p> <p>5.D Describe the initial conditions that must be met for a program segment to work as intended or described.</p>	
	9.4 super Keyword	<p>1.C Determine code that would be used to interact with completed program code.</p> <p>3.B Write program code to define a new type by creating a class.</p>	
	9.5 Creating References Using Inheritance Hierarchies	<p>3.A Write program code to create objects of a class and call methods.</p> <p>5.B Explain why a code segment will not compile or work as intended.</p>	
	9.6 Polymorphism	<p>3.A Write program code to create objects of a class and call methods.</p> <p>5.B Explain why a code segment will not compile or work as intended.</p>	
	9.7 Object Superclass	<p>1.C Determine code that would be used to interact with completed program code.</p> <p>3.B Write program code to define a new type by creating a class.</p>	
	<p>Go to AP Classroom to assign the Personal Progress Check for Unit 9. Review the results in class to identify and address any student misunderstandings.</p>		

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics in this unit. Please refer to the Instructional Approaches section beginning on p. 159 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	9.1	Activating prior knowledge Have students review what they know about classes, methods, and the scope of variables by having them write a class based on specifications that can easily be extended by subclasses. This class will become the superclass for subclasses they write later in the unit.
2	9.2–9.4	Create a plan Given a class design problem that requires the use of multiple classes in an inheritance hierarchy, students identify the common attributes and behaviors among these classes and write these into a superclass. Any additional information that does not belong in the superclass will be categorized to determine the additional classes that might be necessary and what methods will need to be added or overridden in the subclasses.
3	9.4	Think aloud Provide students with a code segment that contains method calls using the <code>super</code> keyword. Have students describe the code segment out loud to themselves. Give students several individual statements that attempt to interact with the given code segment, and have them talk through each one, describing which statements would work and which ones would not, as well as the reasons why those statements wouldn't work.
4	9.5–9.6	Student response system Provide students with several statements where objects are created and the reference type and object type are different but related. Then provide students with calls to methods on these created objects. Use a student response system to have students determine whether each statement is legal, would result in a compile-time error, or would result in a run-time error.



Unit Planning Notes

Use the space below to plan your approach to the unit. Consider how you want to pace your course and where you will incorporate writing and analyzing program code.

.....

.....

.....



After completing this unit, students will have covered all of the necessary content for the Celebrity Lab. The proposed class periods for this unit include time to complete the provided lab activities.

SUGGESTED SKILLS

1.A

Determine an appropriate program design to solve a problem or accomplish a task.

3.B

Write program code to define a new type by creating a class.



AVAILABLE RESOURCES

- [Runestone Academy: AP CSA—Java Review: 11.3—Inheritance](#)
- [Classroom Resources > Object-Oriented Design](#)
- [An Introduction to Polymorphism in Java](#)

TOPIC 9.1

Creating Superclasses and Subclasses

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.1

A class hierarchy can be developed by putting common attributes and behaviors of related classes into a single class called a superclass.

MOD-3.B.2

Classes that extend a superclass, called subclasses, can draw upon the existing attributes and behaviors of the superclass without repeating these in the code.

MOD-3.B.3

Extending a subclass from a superclass creates an “is-a” relationship from the subclass to the superclass.

MOD-3.B.4

The keyword `extends` is used to establish an inheritance relationship between a subclass and a superclass. A class can extend only one superclass.

TOPIC 9.2

Writing Constructors for Subclasses

SUGGESTED SKILLS

3.B

Write program code to define a new type by creating a class.

5.A

Describe the behavior of a given segment of program code.



AVAILABLE RESOURCES

- [Practice-It!: BJP4 Chapter 9: Inheritance and Interfaces—Self-Check 9.3](#)
- Classroom Resources >
 - [An Introduction to Polymorphism in Java](#)
 - [Gradebook Project](#)

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.5

Constructors are not inherited.

MOD-3.B.6

The superclass constructor can be called from the first line of a subclass constructor by using the keyword `super` and passing appropriate parameters.

MOD-3.B.7

The actual parameters passed in the call to the superclass constructor provide values that the constructor can use to initialize the object's instance variables.

MOD-3.B.8

When a subclass's constructor does not explicitly call a superclass's constructor using `super`, Java inserts a call to the superclass's no-argument constructor.

MOD-3.B.9

Regardless of whether the superclass constructor is called implicitly or explicitly, the process of calling superclass constructors continues until the `Object` constructor is called. At this point, all of the constructors within the hierarchy execute beginning with the `Object` constructor.

SUGGESTED SKILLS

3.B

Write program code to define a new type by creating a class.

5.D

Describe the initial conditions that must be met for a program segment to work as intended or described.



AVAILABLE RESOURCES

- [Runestone Academy: AP CSA—Java Review: 11.8—Overriding vs Overloading](#)
- [Practice-It!: BJP4 Chapter 9: Inheritance and Interfaces—Exercises 9.4, 9.9](#)
- Classroom Resources >
 - [An Introduction to Polymorphism in Java](#)
 - [Gradebook Project](#)

TOPIC 9.3

Overriding Methods

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.10

Method overriding occurs when a public method in a subclass has the same method signature as a public method in the superclass.

MOD-3.B.11

Any method that is called must be defined within its own class or its superclass.

MOD-3.B.12

A subclass is usually designed to have modified (overridden) or additional methods or instance variables.

MOD-3.B.13

A subclass will inherit all public methods from the superclass; these methods remain public in the subclass.

TOPIC 9.4

super Keyword

SUGGESTED SKILLS

1.C

Determine code that would be used to interact with completed program code.

3.B

Write program code to define a new type by creating a class.



AVAILABLE RESOURCES

- [Runestone Academy: AP CSA—Java Review: 11.9—Using Super to Call an Overridden Method](#)
- Classroom Resources >
 - [Gradebook Project](#)
 - [Inheritance and Polymorphism with Sudoku](#)

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.14

The keyword `super` can be used to call a superclass's constructors and methods.

MOD-3.B.15

The superclass method can be called in a subclass by using the keyword `super` with the method name and passing appropriate parameters.

SUGGESTED SKILLS

3.A

Write program code to create objects of a class and call methods.

5.B

Explain why a code segment will not compile or work as intended.



AVAILABLE RESOURCES

- Practice-It!: BJP4 Chapter 9: Inheritance and Interfaces—Self-Check 9.8, 9.10
- Classroom Resources > Gradebook Project

TOPIC 9.5

Creating References Using Inheritance Hierarchies

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.C

Define reference variables of a superclass to be assigned to an object of a subclass in the same hierarchy.

ESSENTIAL KNOWLEDGE

MOD-3.C.1

When a class *S* “is-a” class *T*, *T* is referred to as a superclass, and *S* is referred to as a subclass.

MOD-3.C.2

If *S* is a subclass of *T*, then assigning an object of type *S* to a reference of type *T* facilitates polymorphism.

MOD-3.C.3

If *S* is a subclass of *T*, then a reference of type *T* can be used to refer to an object of type *T* or *S*.

MOD-3.C.4

Declaring references of type *T*, when *S* is a subclass of *T*, is useful in the following declarations:

- Formal method parameters
- arrays — `T[]` `var ArrayList<T>` `var`

TOPIC 9.6

Polymorphism

SUGGESTED SKILLS

3.A

Write program code to create objects of a class and call methods.

5.B

Explain why a code segment will not compile or work as intended.



Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.D

Call methods in an inheritance relationship.

ESSENTIAL KNOWLEDGE

MOD-3.D.1

Utilize the `Object` class through inheritance.

MOD-3.D.2

At compile time, methods in or inherited by the declared type determine the correctness of a non-static method call.

MOD-3.D.3

At run-time, the method in the actual object type is executed for a non-static method call.

AVAILABLE LAB

- Classroom Resources > [AP Computer Science A: Celebrity Lab](#)

AVAILABLE RESOURCES

- [Runestone Academy: AP CSA—Java Review: 11.15—Polymorphism](#)
- [Practice-It!: BJP4 Chapter 9: Inheritance and Interfaces—Self-Check 9.9](#)

SUGGESTED SKILLS

1.C

Determine code that would be used to interact with completed program code.

3.B

Write program code to define a new type by creating a class.



AVAILABLE LAB

- Classroom Resources > [AP Computer Science A: Celebrity Lab](#)

AVAILABLE RESOURCE

- Java Quick Reference (see Appendix)

TOPIC 9.7

Object Superclass

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.E

Call `Object` class methods through inheritance.

ESSENTIAL KNOWLEDGE

MOD-3.E.1

The `Object` class is the superclass of all other classes in Java.

MOD-3.E.2

The `Object` class is part of the `java.lang` package

MOD-3.E.3

The following `Object` class methods and constructors—including what they do and when they are used—are part of the Java Quick Reference:

- `boolean equals(Object other)`
- `String toString()`

MOD-3.E.4

Subclasses of `Object` often override the `equals` and `toString` methods with class-specific implementations.