

AP COMPUTER SCIENCE A

**UNIT 3**

**Boolean  
Expressions  
and `if`  
Statements**



**15–17.5%**  
AP EXAM WEIGHTING



**~11–13**  
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 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 3**

**Multiple-choice: ~20 questions**

**Free-response: 2 questions**

- Methods and Control Structures
- Methods and Control Structures: partial

# Boolean Expressions and if Statements



## Developing Understanding

### BIG IDEA 1

#### Control **CON**

- How can you use different conditional statements to write a pick-your-own-path interactive story?
- Why is selection a necessary part of programming languages?

Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators.

The third building block of all algorithms is iteration, which you will cover in Unit 4. Selection and iteration work together to solve problems.

## Building Computational Thinking Practices

**2.B 3.C 4.A 4.C**

Selection allows programmers to incorporate choice into their programs: to create games that react to interactions of the user, to develop simulations that are more real world by allowing for variability, or to discover new knowledge in a sea of information by filtering out irrelevant data. Students should be able to write program code that uses conditional statements for selection. Have students write their program code on paper and trace it with different sample inputs before writing code on the computer.

Programmers need to make design decisions when creating programs that determine how a program specification will be implemented. Typically, there are many ways in which statements can be written to yield the same result, and this final determination is dictated by the programmer. Exposing students to many different code segments that yield equivalent results allows them to be more confident in their solution and helps expose them to new ways of solving the problem that may be better than their current solution.

## Preparing for the AP Exam

The study of computer science involves the analysis of program code. On the multiple-choice section of the exam, students will be asked to determine the result of a given program code segment based on specific input and on the behavior of program code in general. Students should be able to determine the result of program code that uses conditional statements and nested conditional statements to represent nonlinear processes in a program.

Often, students will write program code that mishandles one of the given conditions.

The ability to trace program code can be valuable when testing programs to ensure that all conditions are met. Testing for the different expected behaviors of conditional statements is a critical part of program development and is useful when writing program code or analyzing given code segments. Students should develop sample test cases to illustrate each unique behavior to aid in finding errors and validating results.

## UNIT AT A GLANCE

Enduring Understanding	Topic	Suggested Skills	Class Periods
			~11–13 CLASS PERIODS
CON-1	3.1 Boolean Expressions	2.A Apply the meaning of specific operators.	
CON-2	3.2 if Statements and Control Flow	2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output).  3.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
	3.3 if-else Statements	3.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.  4.A Use test-cases to find errors or validate results.	
	3.4 else if Statements	3.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.  4.C Determine if two or more code segments yield equivalent results.	
CON-1, CON-2	3.5 Compound Boolean Expressions	2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output).  3.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
CON-1	3.6 Equivalent Boolean Expressions	4.C Determine if two or more code segments yield equivalent results.	
	3.7 Comparing Objects	2.C Determine the result or output based on the statement execution order in a code segment containing method calls.  3.A Write program code to create objects of a class and call methods.	
 Go to <b>AP Classroom</b> to assign the <b>Personal Progress Check</b> for Unit 3. Review the results in class to identify and address any student misunderstandings.			

## 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	3.3	<p><b>Code tracing</b></p> <p>Provide students with several code segments that contain conditional statements. Have students trace various sample inputs, keeping track of the statements that get executed and the order in which they are executed. This can help students find errors and validate results.</p>
2	3.2–3.5	<p><b>Pair programming</b></p> <p>Have students work with a partner to create a “guess checker” that could be used as part of a larger game. Students compare four given digits to a preexisting four-digit code that is stored in individual variables. Their program should provide output containing the number of correct digits in correct locations, as well as the number of correct digits in incorrect locations. This program can be continually improved as students learn about nested conditional statements and compound Boolean expressions.</p>
3	3.5	<p><b>Diagramming</b></p> <p>Have students create truth tables by listing all the possible true and false combinations and corresponding Boolean values for a given compound Boolean expression. Once students have created the truth table, provide students with input values. Have students determine the value of each individual Boolean expression and use the truth table to determine the result of the compound Boolean expression.</p>
4	3.6	<p><b>Student response system</b></p> <p>Provide students with a code segment that utilizes conditional statements and a compound Boolean expression, and ask them to choose an equivalent code segment that uses a nested conditional statement (and vice versa). Have them report their responses using a student response system.</p>
5	3.7	<p><b>Predict and compare</b></p> <p>Have students predict the output of several different code segments that compare object references—some that use <code>.equals()</code> and some that use <code>==</code>. Once done, have them create a program that contains those code segments and compare the actual and expected results. This is best illustrated using a simple class that you write yourself.</p>

## SUGGESTED SKILL

## 2.A

Apply the meaning of specific operators.



## AVAILABLE RESOURCES

- Practice-It!: BJP4 Chapter 4: Conditional Execution—Self-Check 4.2

## TOPIC 3.1

# Boolean Expressions

## Required Course Content

### ENDURING UNDERSTANDING

**CON-1**

The way variables and operators are sequenced and combined in an expression determines the computed result.

### LEARNING OBJECTIVE

**CON-1.E**

Evaluate Boolean expressions that use relational operators in program code.

### ESSENTIAL KNOWLEDGE

**CON-1.E.1**

Primitive values and reference values can be compared using relational operators (i.e., == and !=).

**CON-1.E.2**

Arithmetic expression values can be compared using relational operators (i.e., <, >, <=, >=).

**CON-1.E.3**

An expression involving relational operators evaluates to a Boolean value.

## TOPIC 3.2

# if Statements and Control Flow

### Required Course Content

#### ENDURING UNDERSTANDING

##### CON-2

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

#### LEARNING OBJECTIVE

##### CON-2.A

Represent branching logical processes by using conditional statements.

#### ESSENTIAL KNOWLEDGE

##### CON-2.A.1

Conditional statements interrupt the sequential execution of statements.

##### CON-2.A.2

if statements affect the flow of control by executing different statements based on the value of a Boolean expression.

##### CON-2.A.3

A one-way selection (if statement) is written when there is a set of statements to execute under a certain condition. In this case, the body is executed only when the Boolean condition is true.

#### SUGGESTED SKILLS

##### 2.B

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

##### 3.C

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.



#### AVAILABLE RESOURCES

- [Runestone Academy: AP CSA—Java Review: 5.1—Conditionals](#)
- [Practice-It!: BJP4 Chapter 4: Conditional Execution—Self-Check 4.3; Exercises 4.2 and 4.3](#)
- [The Exam > 2017 AP Computer Science A Exam Free-Response Question #1, Part A \(Phrase\)](#)

## SUGGESTED SKILLS

## 3.C

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

## 4.A

Use test-cases to find errors or validate results.



## AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 5.1—Conditionals
- Practice-It!: BJP4 Chapter 4: Conditional Execution—Self-Check 4.5–4.6, 4.10–4.12

## TOPIC 3.3

**if-else Statements**

## Required Course Content

## ENDURING UNDERSTANDING

## CON-2

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

## LEARNING OBJECTIVE

## CON-2.A

Represent branching logical processes by using conditional statements.

## ESSENTIAL KNOWLEDGE

## CON-2.A.4

A two-way selection is written when there are two sets of statements— one to be executed when the Boolean condition is `true`, and another set for when the Boolean condition is `false`. In this case, the body of the “if” is executed when the Boolean condition is `true`, and the body of the “else” is executed when the Boolean condition is `false`.



# TOPIC 3.4

## else if Statements

**SUGGESTED SKILLS**

**3.C**

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

**4.C**

Determine if two or more code segments yield equivalent results.



**AVAILABLE RESOURCES**

- [Runestone Academy: AP CSA—Java Review: 5.2—Three or More Options](#)

### Required Course Content

#### ENDURING UNDERSTANDING

**CON-2**

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

#### LEARNING OBJECTIVE

**CON-2.A**

Represent branching logical processes by using conditional statements.

#### ESSENTIAL KNOWLEDGE

**CON-2.A.5**

A multi-way selection is written when there are a series of conditions with different statements for each condition. Multi-way selection is performed using `if-else-if` statements such that exactly one section of code is executed based on the first condition that evaluates to true.

**SUGGESTED SKILLS**

**2.B**

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

**3.C**

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.



**AVAILABLE RESOURCES**

- [Runestone Academy: AP CSA—Java Review: 5.3—Complex Conditionals](#)
- [Practice-It!: BJP4 Chapter 4: Conditional Execution—Exercise 4.12](#)

**TOPIC 3.5**

**Compound Boolean Expressions**

**Required Course Content**

**ENDURING UNDERSTANDING**

**CON-2**

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

**LEARNING OBJECTIVE**

**CON-2.B**

Represent branching logical processes by using nested conditional statements.

**ESSENTIAL KNOWLEDGE**

**CON-2.B.1**

Nested `if` statements consist of `if` statements within `if` statements.

**ENDURING UNDERSTANDING**

**CON-1**

The way variables and operators are sequenced and combined in an expression determines the computed result.

**LEARNING OBJECTIVE**

**CON-1.F**

Evaluate compound Boolean expressions in program code.

**ESSENTIAL KNOWLEDGE**

**CON-1.F.1**

Logical operators `!` (not), `&&` (and), and `||` (or) are used with Boolean values. This represents the order these operators will be evaluated.

**CON-1.F.2**

An expression involving logical operators evaluates to a Boolean value.

**CON-1.F.3**

When the result of a logical expression using `&&` or `||` can be determined by evaluating only the first Boolean operand, the second is not evaluated. This is known as short-circuited evaluation.

## TOPIC 3.6

# Equivalent Boolean Expressions

**SUGGESTED SKILL**

**4.C**

Determine if two or more code segments yield equivalent results.



**AVAILABLE RESOURCE**

- [Runestone Academy: AP CSA—Java Review: 5.5—De Morgan's Laws](#)

### Required Course Content

#### ENDURING UNDERSTANDING

**CON-1**

The way variables and operators are sequenced and combined in an expression determines the computed result.

#### LEARNING OBJECTIVE

**CON-1.G**

Compare and contrast equivalent Boolean expressions.

#### ESSENTIAL KNOWLEDGE

**CON-1.G.1**

De Morgan's Laws can be applied to Boolean expressions.

**CON-1.G.2**

Truth tables can be used to prove Boolean identities.

**CON-1.G.3**

Equivalent Boolean expressions will evaluate to the same value in all cases.

## SUGGESTED SKILLS

## 2.C

Determine the result or output based on the statement execution order in a code segment containing method calls.

## 3.A

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

## TOPIC 3.7

# Comparing Objects

## Required Course Content

### ENDURING UNDERSTANDING

**CON-1**

The way variables and operators are sequenced and combined in an expression determines the computed result.

### LEARNING OBJECTIVE

**CON-1.H**

Compare object references using Boolean expressions in program code.

### ESSENTIAL KNOWLEDGE

**CON-1.H.1**

Two object references are considered aliases when they both reference the same object.

**CON-1.H.2**

Object reference values can be compared, using `==` and `!=`, to identify aliases.

**CON-1.H.3**

A reference value can be compared with `null`, using `==` or `!=`, to determine if the reference actually references an object.

**CON-1.H.4**

Often classes have their own `equals` method, which can be used to determine whether two objects of the class are equivalent.