Notes on the AP Computer Science A Practice Exam

Multiple-Choice Section

Course Framework Alignment and Rationales

Question 1

Skill	Learning Objective	Topic
2.B: Determine the result or output based on statement execution order in a code segment without method calls (other than output).	CON-1.A: Evaluate arithmetic expressions in program code.	Expressions and Assignment Statements

(A)	Incorrect. This would be the result if the addition operation in the third assignment statement was evaluated before the remainder operation, as in `int c = 7 % (4 + 3)`.
(B)	Incorrect. This would be the result if the addition operation in the second assignment statement was evaluated before the integer division operation, as in `int b = (4 + 3) / 2`.
(C)	**Correct.** In the first assignment statement, the multiplication operation is evaluated before the addition operation and `a` is assigned the value `9`. In the second assignment statement, the integer division is evaluated first and produces a result of `1`, which is added to `4` so that the variable `b` is assigned the value `5`. In the third assignment statement, the remainder operation is evaluated before the addition operation and `c` is assigned the value `6`. The variable `d` is assigned the value `20.0`.
(D)	Incorrect. This would be the result if the division operator in the second assignment statement performed floating point division instead of integer division, as in `int b = 4 + 3.0 / 2`.
(E)	Incorrect. This would be the result if the addition operation in the first assignment statement was evaluated before the multiplication operation, as in `int a = (3 + 2) * 3`.

Question 2

Skill		Learning Objective	Topic
5.A: Describe the behavior of a given segment of program code.		CON-2.B: Represent branching logical processes by using nested conditional statements. CON-2.A: Represent branching logical processes by using conditional statements.	Compound Boolean Expressions if-else Statements
(A)	Incorrect. When `num` is zero or a negative integer, whether even or odd, the body of the outer `if` statement is not executed and nothing is printed. When `num` is a positive integer, either `"A"` or `"B"` is printed.		
(B)	Incorrect. When `num` is zero or a negative integer, whether even or odd, the body of the outer `if` statement is not executed and nothing is printed. When `num` is a positive integer, either `"A"` or `"B"` is printed.		
(C)	Incorrect. `"B"` is only printed in the case of a positive odd integer. Nothing is printed if `num` is not positive.		
(D)	**Correct.** When `num` is positive and even, `"A"` is printed. When `num` is positive and not even (odd), `"B"` is printed. When `num` is not positive, nothing is printed.		
(E)	Incorrect. This would be the result if the condition in the second `if` statement was `num % 2 != 0`.		

Question 3

Skill	Learning Objective	Topic
1.C: Determine code that would be used to interact with completed code.	CON-1.D: Evaluate expressions that use the Math class methods. CON-1.A: Evaluate arithmetic expressions in program code. CON-1.C: Evaluate arithmetic expressions that use casting.	Using the Math Class Expressions and Assignment Statements Casting and Ranges of Variables

(A)	Incorrect. Since `marker1` and `marker2` are always positive, taking the absolute value of each one has no effect. In this statement, if `marker2` is greater than `marker1`, `hours` is assigned a negative value.
(B)	Incorrect. In this statement, parentheses are incorrectly placed, so only `marker2` is divided by `60.0`, not the absolute value of the difference between `marker1` and `marker2`.
(C)	**Correct.** The code segment takes the absolute value of the difference between `marker1` and `marker2`, always producing a positive distance, and then divides the result by the vehicle's speed.
(D)	Incorrect. Since `marker1` and `marker2` are both of type `int`, the expression `(marker1 - marker2) / 60` performs integer division. For example, when `marker1` has the value `100` and `marker2` has the value `70`, the expression evaluates to `0` instead of the intended `0.5`.
(E)	Incorrect. Since `marker1` and `marker2` are both of type `int`, the expression `(marker1 - marker2) / 60` performs integer division. The casting of the result of the division to a `double` occurs too late.

Question 4

Skill		Learning Objective	Topic
2.B: Determine the result or output based on statement execution order in a code segment without method calls (other than output).		CON-2.B: Represent branching logical processes by using nested conditional statements. CON-2.A: Represent branching logical processes by using conditional statements.	Compound Boolean Expressions if Statements and Control Flow if-else Statements
(A)	Incorrect. This result would be printed as a result of the call `message(5, 5, 15)`.		
(B)	Incorrect. This result would be printed as a result of the call `message(5, 5, 5)`.		
(C)	Incorrect. This result would be printed as a result of the call `message(15, 15, 5)`.		
(D)	**Correct.** Since `a < 10` evaluates to `true`, the body of the `if` statement is executed. Since `b < 10` evaluates to `false`, `"X"` is not printed; `"Y"` is printed. Since `c < 10` evaluates to `true`, the body of the `if` statement is executed, and since `b > 10` evaluates to `true`, `"Y"` is printed.		
(E)	Incorrect. This result would be printed as a result of the call `message(15, 5, 5)`.		

Question 5

Skill		Learning Objective	Topic
1.C: Determine code that would be used to interact with completed code.		MOD-1.C: Identify, using its signature, the correct constructor being called.	Creating and Storing Objects (Instantiation)
(A)	Incorrect. This constructor's signature differs from the signature of the existing constructor, so the new constructor can safely be added to the class definition.		
(B)	Incorrect. This constructor's signature differs from the signature of the existing constructor, so the new constructor can safely be added to the class definition.		
(C)	Incorrect. This constructor's signature differs from the signature of the existing constructor, so the new constructor can safely be added to the class definition.		
(D)	Incorrect. This constructor's signature differs from the signature of the existing constructor, so the new constructor can safely be added to the class definition.		
(E)	**Correct.** This constructor has the same signature as the existing constructor `(String, String, boolean)`. A compiler error will occur.		

Question 6

Skill	Learning Objective	Topic
2.A: Apply the meaning of specific operators.	CON-1.A: Evaluate arithmetic expressions in program code. CON-1.C: Evaluate arithmetic expressions that use casting.	Expressions and Assignment Statements Casting and Ranges of Variables

(A)	**Correct.** In option I, the cast applies to the value `2`, so floating-point division is performed and the expression evaluates to `0.5 + 3`, or `3.5`. In option II, the cast applies to the result of the integer division `2 / 4`, so the expression evaluates to `0.0 + 3`, or `3.0`. In option III, the cast applies to the sum of `3` and the result of the integer division `2 / 4`, so the expression evaluates to `(double) (0 + 3)`, or `3.0`.
(B)	Incorrect. In option I, the cast applies to the value `2`, so floating-point division is performed and the expression evaluates to `0.5 + 3`, or `3.5`. In option III, the cast applies to the sum of `3` and the result of the integer division `2 / 4`, so the expression evaluates to `(double) (0 + 3)`, or `3.0`.
(C)	Incorrect. In option II, the cast applies to the result of the integer division `2 / 4`, so the expression evaluates to `0.0 + 3`, or `3.0`.
(D)	Incorrect. In option I, the cast applies to the value `2`, so floating-point division is performed and the expression evaluates to `0.5 + 3`, or `3.5`. In option II, the cast applies to the result of the integer division `2 / 4`, so the expression evaluates to `0.0 + 3`, or `3.0`. In option III, the cast applies to the sum of `3` and the result of the integer division `2 / 4`, so the expression evaluates to `(double) (0 + 3)`, or `3.0`.
(E)	Incorrect. In option II, the cast applies to the result of the integer division `2 / 4`, so the expression evaluates to `0.0 + 3`, or `3.0`. In option III, the cast applies to the sum of `3` and the result of the integer division `2 / 4`, so the expression evaluates to `(double) (0 + 3)`, or `3.0`.

Question 7

Skill		Learning Objective	Topic
4.C: Determine if two or more code segments yield equivalent results.		CON-2.B: Represent branching logical processes by using nested conditional statements. CON-2.A: Represent branching logical processes by using conditional statements. CON-1.F: Evaluate compound Boolean expressions in program code.	Compound Boolean Expressions if-else Statements
(A)	Incorrect. The statement assigns a different value to `b2` than the code segment assigns to `b1` when `num` is between `-100`, exclusive, and `0`, inclusive, or when `num` is less than `-100`.		
(B)	Incorrect. The statement assigns `true` to `b2` for all values of `num`.		
(C)	Incorrect. The statement assigns a different value to `b2` than the code segment assigns to `b1` when `num` is between `0` and `100`, exclusive, or when `num` is greater than `100`.		
(D)	Incorrect. The statement assigns a different value to `b2` than the code segment assigns to `b1` when `num` is between `0` and `100`, exclusive.		
(E)	**Correct.** In the body of the first `if` clause in the code segment, `b1` retains the value `true` if `num` is between `0` and `100`, exclusive. In the body of the `else` clause, `b1` retains the value `true` if `num` is less than `-100`. The statement assigns `true` to `b2` if `num` is less than `-100` or between `0` and `100`, exclusive.		

Question 8

Skill		Learning Objective	Topic
4.B: Identify errors in program code.		VAR-1.G: Explain where variables can be used in the program code. MOD-1.C: Identify, using its signature, the correct constructor being called.	Scope and Access Creating and Storing Objects (Instantiation)
(A)	Incorrect. A constructor signature consists of the constructor name and the parameter list. A correct constructor header does not include a return type.		
(B)	Incorrect. Assigning `int` values to `double` variables is allowed, although assigning `double` values to `int` variables is not allowed.		
(C)	Incorrect. The `void` return type of the `incrementPoints` method is correct because the method does not return a value.		
(D)	**Correct.** The variables `n1` and `n2` are not instance variables of the `Points` class, nor are they defined in the `incrementPoints` method. The instance variables `num1` and `num2` should have been used instead of `n1` and `n2`.		
(E)	Incorrect. The variable `value` is the parameter passed to the `incrementPoints` method.		

Question 9

Skill		Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.		VAR-2.D: Represent collections of related object reference data using ArrayList objects.	ArrayList Methods
(A)	Incorrect. This output would be printed if the two-parameter `add` method call were `numList.add(0, 1)` and the `set` method call were `numList.set(2, 0)`.		
(B)	**Correct.** The three single-parameter `add` method calls create an `ArrayList` with the contents `[3, 2, 1]`. The two-parameter `add` method call inserts an element with the value `0` at position `1`, so the `ArrayList` contains `[3, 0, 2, 1]`. The `set` method call sets the value of the element at position `0` to `2`, and the `ArrayList` contains `[2, 0, 2, 1]` at the end of the code segment.		
(C)	Incorrect. This output would be printed if the one-parameter `add` method calls were `numList.add(0, 3)`, `numList.add(0, 2)`, and `numList.add(0, 1)`.		
(D)	Incorrect. This output would be printed if the two-parameter `add` method call were `numList.add(0, 1)`.		
(E)	Incorrect. This output would be printed if the two-parameter `set` method call were `numList.set(2, 0)`.		

Question 10

Skill		Learning Objective	Topic
4.A: Use test-cases to find errors or validate results.		CON-2.E: Represent iterative processes using a for loop. CON-1.A: Evaluate arithmetic expressions in program code.	for Loops Expressions and Assignment Statements
(A)	Incorrect. The loop body is never executed as a result of this method call and nothing is printed.		
(B)	Incorrect. This method call results in the output `"0 "`.		
(C)	Incorrect. This method call results in the output `"0 "`.		
(D)	**Correct.** The `for` loop iterates from `i = 0` to `i = 19`. The expression `i % num2 == 0` evaluates to `true` when `i` is divisible by `5` and the expression `i % 2 == 0` evaluates to true when `i` is even. The only values in the range `0` to `19`, inclusive, that are both divisible by `5` and even are `0` and `10`, so the statement prints `"0 10 "`.		
(E)	Incorrect. This method call results in the output `"0 10 20 "`.		

Question 11

Skill		Learning Objective	Topic
1.B: Determine code that would be used to complete code segments.		CON-2.C: Represent iterative processes using a while loop. CON-1.B: Evaluate what is stored in a variable as a result of an expression with an assignment statement.	while Loops Compound Assignment Operators
(A)	Incorrect. This code segment prints `"10987654321"`.		
(B)	Incorrect. This code segment prints `"109876543210"`.		
(C)	**Correct.** During the first iteration of the `while` loop, `num` is decremented and `"9"` is printed. During the second iteration, `num` is decremented and `"8"` is printed. This continues until the last iteration of the loop, when `num` is decremented and `"1"` is printed. At this point, the Boolean expression in the `while` loop evaluates to `false` and the loop terminates. The code segment prints `"987654321"`.		
(D)	Incorrect. This code segment prints `"9876543210"`.		
(E)	Incorrect. This code segment prints `"10987654321"`.		

Question 12

Skill	Learning Objective	Topic
1.C: Determine code that would be used to interact with completed code.	CON-1.H: Compare object references using Boolean expressions in program code. CON-2.A: Represent branching logical processes by using conditional statements.	Comparing Objects if Statements and Control Flow

(A)	Incorrect. Condition I is incorrect. If no `Person` object has been assigned to `borrower`, the method call `borrower.equals(null)` throws a `NullPointerException`. Condition II is correct. This condition ensures that `borrower` contains a reference to an object when it is used in the `println` method call that follows.
(B)	**Correct.** Condition I is incorrect. If no `Person` object has been assigned to `borrower`, the method call `borrower.equals(null)` throws a `NullPointerException`. Condition II is correct. This condition ensures that `borrower` contains a reference to an object when it is used in the `println` method call that follows. Condition III is incorrect. If no `Person` object has been assigned to `borrower`, the method call `borrower.getName()` throws a `NullPointerException`.
(C)	Incorrect. Condition II is correct. This condition ensures that `borrower` contains a reference to an object when it is used in the `println` method call that follows. Condition III is incorrect. If no `Person` object has been assigned to `borrower`, the method call `borrower.getName()` throws a `NullPointerException`.
(D)	Incorrect. Condition I is incorrect. If no `Person` object has been assigned to `borrower`, the method call `borrower.equals(null)` throws a `NullPointerException`.
(E)	Incorrect. Condition III is incorrect. If no `Person` object has been assigned to `borrower`, the method call `borrower.getName()` throws a `NullPointerException`.

Question 13

Skill		Learning Objective	Topic
4.C: Determine if two or more code segments yield equivalent results.		CON-1.G: Compare and contrast equivalent Boolean expressions.	Equivalent Boolean Expressions
(A)	Incorrect. The expressions are not equivalent when `a` has the value `false`, `b` has the value `true`, and `c` has the value `true`.		
(B)	Incorrect. The expressions are not equivalent when `a` has the value `false`, `b` has the value `false`, and `c` has the value `false`.		
(C)	Incorrect. The expressions are not equivalent when `a` has the value `true`, `b` has the value `false`, and `c` has the value `false`.		
(D)	Incorrect. The expressions are not equivalent when `a` has the value `true`, `b` has the value `false`, and `c` has the value `false`.		
(E)	**Correct.** By De Morgan's laws, `!(a && b)` is equivalent to `!a || !b` and the entire expression is equivalent to `!a || !b || c`.		

Question 14

Skill		Learning Objective	Topic
1.B: Determine code that would be used to complete code segments.		CON-2.A: Represent branching logical processes by using conditional statements.	if Statements and Control Flow else if Statements
(A)	Incorrect. Code segment III returns the correct category through the use of an immediate `return` within each of the one-way selection statements.		
(B)	Incorrect. Code segment I uses multi-way selection to assign and return the correct category.		
(C)	Incorrect. Code segment II returns `"rural"` for all values of `density` because it uses a series of one-way selection statements instead of multi-way selection. Code segment III returns the correct category through the use of an immediate `return` within each of the one-way selection statements.		
(D)	**Correct.** Code segment I uses multi-way selection to assign and return the correct category. Code segment II returns `"rural"` for all values of `density` because it uses a series of one-way selection statements instead of multi-way selection. Code segment III returns the correct category through the use of an immediate `return` within each of the one-way selection statements.		
(E)	Incorrect. Code segment II returns `"rural"` for all values of `density` because it uses a series of one-way selection statements instead of multi-way selection.		

Question 15

Skill	Learning Objective	Topic
5.A: Describe the behavior of a given segment of program code.	CON-1.D: Evaluate expressions that use the Math class methods.	Using the Math Class

(A)	Incorrect. This would describe the value assigned to `b` if the value returned by `random` was cast to an `int` before being multiplied by `a`, as in `int b = a + ((int) Math.random()) * a`.
(B)	Incorrect. This would describe the value assigned to `b` if the value returned by `random` was rounded up to `1` before being multiplied by `a`, as in `int b = a + (int) (Math.random() + 1) * a`.
(C)	Incorrect. This would describe the value assigned to `b` if the second assignment statement was `int b = (int) (Math.random() * a)`.
(D)	Incorrect. This would describe the value assigned to `b` if `random` returned values between `0.0` and `1.0`, inclusive. Instead, `random` returns values between `0.0`, inclusive, and `1.0`, exclusive.
(E)	**Correct.** The `random` method returns a value between `0.0`, inclusive, and `1.0`, exclusive. Multiplying that value by `a` and casting to an `int` produces a result between `0` and `a - 1`, inclusive. The sum of `a` and a value between `0` and `a - 1`, inclusive, is a value between `a` and `2 * a - 1`, inclusive.

Question 16

Skill	Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.	CON-2.O: Determine the result of executing recursive methods. CON-2.E: Represent iterative processes using a for loop.	Recursion for Loops

(A)	Incorrect. This output would be printed if the recursive call `stars(num - 1)` were missing.
(B)	**Correct.** The recursive call of the `stars` method occurs before any output is printed, so the method call `stars(5)` results in a recursive call to `stars(4)`, then to `stars(3)`, then to `stars(2)`, and finally to `stars(1)`. The call to `stars(1)` returns immediately without printing any output, so the first call that produces output is `stars(2)`, which prints a row of two stars. Then, `stars(3)` prints a row of three stars, `stars(4)` prints a row of four stars, and finally `stars(5)` prints a row of five stars.
(C)	Incorrect. This output would be printed if the base case was `num == 0` instead of `num == 1`.
(D)	Incorrect. This output would be printed if the recursive call `stars(num - 1)` was the last line of the method instead of occurring before the statements that produce output.
(E)	Incorrect. This output would be printed if the recursive call `stars(num - 1)` was the last line of the method instead of occurring before the statements that produce output and if the base case was `num == 0` instead of `num == 1`.

Question 17

Skill	Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.	MOD-3.B: Create an inheritance relationship from a subclass to the superclass. MOD-3.D: Call methods in an inheritance relationship.	Overriding Methods super Keyword Polymorphism

(A)	Incorrect. This value would be printed if `j` was instantiated as a `Hero` object instead of as a `SuperHero` object, and if the `powerUp` method in the `Hero` class assigned the value `p` to the instance variable `power` rather than incrementing `power` by `p`.
(B)	Incorrect. This value would be printed if the `powerUp` method in the `Hero` class assigned the value `p` to the instance variable `power` rather than incrementing `power` by `p`.
(C)	Incorrect. This value would be printed if `j` was instantiated as a `Hero` object instead of as a `SuperHero` object.
(D)	**Correct.** Since `j` is instantiated as a `SuperHero` object, the `j.powerUp(10)` method call accesses the subclass method. The subclass method uses the `super` keyword to access the superclass method with the parameter `20`. As a result, the instance variable `power` is incremented by `20`.
(E)	Incorrect. This value would be printed if the `powerUp` method in the `Hero` class doubled the value of the instance variable `power` rather than incrementing `power` by `p`.

Question 18

Skill	Learning Objective	Topic
1.B: Determine code that would be used to complete code segments.	VAR-2.B: Traverse the elements in a 1D array. CON-2.E: Represent iterative processes using a for loop.	Traversing Arrays for Loops

(A)	Incorrect. The reference `data[p + 1]` in the Boolean condition in the `if` statement causes an `ArrayIndexOutOfBoundsException` to be thrown when `p` has the value `data.length - 1`.
(B)	Incorrect. The reference `data[p - 1]` in the Boolean condition in the `if` statement causes an `ArrayIndexOutOfBoundsException` to be thrown when `p` has the value `0`.
(C)	Incorrect. The reference `data[p - 1]` in the Boolean condition in the `if` statement causes an `ArrayIndexOutOfBoundsException` to be thrown when `p` has the value `0`.
(D)	Incorrect. The reference `data[p + 1]` in the Boolean condition in the `if` statement causes an `ArrayIndexOutOfBoundsException` to be thrown when `p` has the value `data.length - 1`.
(E)	**Correct.** By definition, the first candidate for a local maximum is the element at index `1` (the second element in the array), and the last candidate is the element at index `data.length - 2` (the next to last element in the array).

Question 19

Skill		Learning Objective	Topic
2.B: Determine the result or output based on statement execution order in a code segment without method calls (other than output).		VAR-2.G: For 2D array objects— a. Traverse using nested for loops. b. Traverse using nested enhanced for loops. VAR-2.F: Represent collections of related primitive or object reference data using two-dimensional (2D) array objects.	Traversing 2D Arrays 2D Arrays
(A)	Incorrect. This would be the result if the Boolean expression in the outer `for` loop was `j < values.length - 1`.		
(B)	Incorrect. This would be the result if the Boolean expression in the inner `for` loop was `k < values[0].length - 1`.		
(C)	Incorrect. This would be the result of adding the original elements of the `values` array, without doubling the first element of every row.		
(D)	**Correct.** The nested `for` loops traverse the two-dimensional array `values`. The first element of each row is doubled and then the sum of all elements is computed as 2+2+3+8+5+6=26.		
(E)	Incorrect. This would be the result if the condition in the `if` statement was `j == 0` instead of `k == 0`. This would have the effect of doubling the elements in the first row of `values` instead of those in the first column.		

Question 20

Skill	Learning Objective	Topic
1.C: Determine code that would be used to interact with completed code.	VAR-2.C: Traverse the elements in a 1D array object using an enhanced for loop. CON-2.I: For algorithms in the context of a particular specification that requires the use of array traversals—a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	Enhanced for Loop for Arrays Developing Algorithms Using Arrays

(A)	Incorrect. The instance variable `pages` is declared `private` in the `Book` class, so it must be accessed from outside the `Book` class using the accessor method `getPages`.
(B)	**Correct.** The enhanced `for` loop traverses `bookArr` and the loop control variable `b` is assigned `Book` objects. The instance variable `pages` is declared `private` in the `Book` class, so it must be accessed from outside the `Book` class using the accessor method `getPages`. The `if` statement compares the value returned by the call `b.getPages()` to the current value of `maxPages`. If the returned value is greater than `maxPages`, `maxPages` is updated with the new maximum value.
(C)	Incorrect. In the enhanced `for` loop, `b` is a `Book` object and cannot be used as an index. In addition, `Book` is a class name, not a variable, and cannot be indexed. Finally, the instance variable `pages` is declared `private` in the `Book` class, so it must be accessed from outside the `Book` class using the accessor method `getPages`.
(D)	Incorrect. In the enhanced `for` loop, `b` is a `Book` object and cannot be used as an index. The instance variable `pages` is declared `private` in the `Book` class, so it must be accessed from outside the `Book` class using the accessor method `getPages`.
(E)	Incorrect. In the enhanced `for` loop, `b` is a `Book` object and cannot be used as an index.

Question 21

Skill	Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.	CON-2.I: For algorithms in the context of a particular specification that requires the use of array traversals—a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm. VAR-1.E: For String class—a. Create String objects. b. Call String methods.	Developing Algorithms Using Arrays String Methods

(A)	Incorrect. This would represent the contents of `resultOne` if the method assigned values to elements of `resultOne` in ascending, alphabetical order.
(B)	Incorrect. This would represent the contents of `resultOne` if the method assigned values to elements of `resultOne` in increasing order of string length.
(C)	Incorrect. This would represent the contents of `resultOne` if the inner `for` loop initialization were `k = j + 2` instead of `k = j + 1`.
(D)	**Correct.** The method assigns the shortest string that occurs in any element of `arr` between `arr[n]` and `arr[arr.length - 1]`, inclusive, to `result[n]`. The shortest string found between `arr[0]` and `arr[3]` is `"of"`, so `result[0]` is assigned the value `"of"`. The shortest string found between `arr[1]` and `arr[3]` is also `"of"`, so `result[1]` is also assigned the value `"of"`. The same is true for the part of the array that begins at index `2` and ends at index `3`, so `result[2]` is also assigned the value `"of"`. In the last iteration of the outer `for` loop, there are no values to consider after `arr[3]`, so `result[3]` is assigned the value `"spring"`.
(E)	Incorrect. This would represent the contents of `resultOne` if the method assigned values to elements of `resultOne` in decreasing order of string length.

Question 22

Skill		Learning Objective	Topic
2.D: Determine the number of times a code segment will execute.		CON-2.I: For algorithms in the context of a particular specification that requires the use of array traversals—a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm. VAR-1.E: For String class—a. Create String objects. b. Call String methods.	Developing Algorithms Using Arrays String Methods
(A)	**Correct.** Line 12 is executed each time the variable `sm` is updated because a new smallest value is found. When `j` has the value `0`, `sm` is updated for `"day"` and `"of"`. When `j` has the value `1`, `sm` is updated for `"of"`. When `j` has the value `4`, `sm` is updated for `"year"`. When `j` has any of the values `2`, `3`, or `5`, `sm` is not updated. Line 12 is executed four times.		
(B)	Incorrect. This would be the result if `sm` was updated once each time a string shorter than `arr[j]` was found instead of once each time a new smallest value was identified.		
(C)	Incorrect. This would be the result if line 12 were executed once for each element of `arr`.		
(D)	Incorrect. This would be the result if the method had no `if` statement and `sm` was updated once for each pair `arr[j]` and `arr[k]` encountered in the nested `for` loops.		
(E)	Incorrect. This would be the result if the method had no `if` statement and the initialization in the inner `for` loop was `k = 1` instead of `k = j + 1`.		

Question 23

Skill	Learning Objective	Topic
1.B: Determine code that would be used to complete code segments.	VAR-2.G: For 2D array objects— a. Traverse using nested for loops. b. Traverse using nested enhanced for loops.	Traversing 2D Arrays

(A)	Incorrect. The outer `for` loop in this code segment declares `j`, a row of `arr`, as an `int` rather than an `int[]`. This code segment also attempts to print `j`, a row of `arr`, instead of `k`, an element of `j`.
(B)	Incorrect. The outer `for` loop in this code segment declares `j`, a row of `arr`, as an `int` rather than an `int[]`.
(C)	Incorrect. This code segment attempts to print `j`, a row of `arr`, instead of `k`, an element of `j`.
(D)	**Correct.** The outer `for` loop stores each row of the two-dimensional array in `j`, a one-dimensional array. The inner `for` loop stores each element of `j` in `k` and prints `k`.
(E)	Incorrect. This code segment uses `k`, an element of the two-dimensional array, as an index in a one-dimensional array.

Question 24

Skill	Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.	MOD-2.H: Define the static variables that belong to the class.	Static Variables and Methods

(A)	Incorrect. This would be the result if `y` was not declared as a `static` variable and the `SomeClass` constructor did not increment `y`.
(B)	Incorrect. This would be the result if `y` was not declared as a `static` variable.
(C)	Incorrect. This would be the result if the `SomeClass` constructor did not increment `y`.
(D)	**Correct.** Since `y` is declared as a `static` variable, it is associated with the class and all objects of the class share the single variable `y`. Each time a new `SomeClass` object is instantiated, the value of `y` is incremented by `1`. After the `third` object is instantiated, the value of `y` is `3`. The call to `incrementY` with no parameter increments the value of `y` by `1`, and the call to `incrementY` with a parameter value of `10` adds `10` to the value of `y`, resulting in `14`.
(E)	Incorrect. This would be the result if the `getY` method returned `x` instead of `y`.

Question 25

Skill	Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.	CON-2.F: For algorithms in the context of a particular specification that involves String objects— a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm. VAR-1.E: For String class— a. Create String objects. b. Call String methods. CON-2.E: Represent iterative processes using a for loop.	Developing Algorithms Using Strings String Methods for Loops

(A)	Incorrect. This would be the result if the statement assigning a value to `temp` were `temp = str.substring(i - 1, i) + temp`.
(B)	Incorrect. This would be the result if the statement assigning a value to `temp` were `temp = str.substring(i - 1, i) + temp` and if the loop control variable in the `for` loop were initialized to `str.length()` instead of `str.length() - 1`.
(C)	Incorrect. This would be the result if the loop control variable in the `for` loop were initialized to `str.length()` instead of `str.length() - 1`.
(D)	**Correct.** When `i` has the value `4`, `temp` is assigned the value `"l"`. When `i` has the value `3`, `"p"` is appended to `temp`, resulting in `"lp"`. When `i` has the value `2`, `"p"` is appended to `temp` again, resulting in `"lpp"`. In the last iteration of the `for` loop, `i` has the value `1` and `"a"` is appended to temp, resulting in `"lppa"`.
(E)	Incorrect. This would be the result if the condition in the `for` loop were `i >= 0` instead of `i > 0`.

Question 26

Skill	Learning Objective	Topic
2.D: Determine the number of times a code segment will execute.	CON-2.H: Compute statement execution counts and informal run-time comparison of iterative statements. CON-2.E: Represent iterative processes using a for loop. CON-2.G: Represent nested iterative processes.	Informal Code Analysis for Loops Nested Iteration

(A)	Incorrect. This would be the correct comparison if the initialization and Boolean condition in the inner `for` loop of code segment II were `k = 0` and `k <= n`, respectively.
(B)	Incorrect. This would be the correct comparison if the initializations in the outer and inner `for` loops of code segment II were `j = 0` and `k = 0`, respectively.
(C)	**Correct.** There are `m * n` iterations of the `for` loop in code segment I. In code segment II, the outer loop executes `m` times and the inner loop executes `n - 1` times for each iteration of the outer loop. There are `m * n - m` iterations of the inner loop in code segment II, so `"A"` is printed `m` more times than `"B"` is printed.
(D)	Incorrect. This would be the correct comparison if the Boolean conditions in the outer and inner `for` loops of code segment II were `j < m` and `k <= n`, respectively.
(E)	Incorrect. This would be the correct comparison if the Boolean condition in the inner `for` loop of code segment II were `k <= n`.

Question 27

Skill		Learning Objective	Topic
5.D: Describe the initial conditions that must be met for a program segment to work as intended or described.		CON-1.G: Compare and contrast equivalent Boolean expressions.	Equivalent Boolean Expressions
(A)	Incorrect. The variable `c` will be assigned the value `true` when `a` and `b` both have the value `true` or when `a` has the value `false` and `b` has the value `true`.		
(B)	Incorrect. The variable `c` will be assigned the value `false` when `b` has the value `false`, regardless of the value of `a`.		
(C)	Incorrect. If `a` and `b` are both `true`, then `(a && b)` is `true`, `(!a && b)` is `false`, and the entire expression `true || false` evaluates to `true`.		
(D)	Incorrect. If a has the value `false` and `b` has the value `true`, then `(a && b)` is `false`, `(!a && b)` is `true`, and the entire expression `false || true` evaluates to `true`.		
(E)	**Correct.** When `b` has the value `false`, both of the expressions `(a && b)` and `(!a && b)` evaluate to `false`, regardless of the value of `a`. The entire expression evaluates to `false || false`, or `false`. When `b` has the value `true`, one of the expressions `(a && b)` or `(!a && b)` evaluates to `true`. The entire expression, in this case, is either `true || false` or `false || true`, or `true`. A truth table can be used to summarize these results.		

Question 28

Skill	Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.	CON-2.F: For algorithms in the context of a particular specification that involves String objects— a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm. VAR-1.E: For String class— a. Create String objects. b. Call String methods. CON-2.C: Represent iterative processes using a while loop.	Developing Algorithms Using Strings String Methods while Loops

(A)	Incorrect. This value would be returned if the first statement in the `while` loop was `a = a.substring(0, x)`.
(B)	**Correct.** The method `abMethod(String a, String b)` removes all non-overlapping occurrences of string `b` from string `a` and returns the resulting `String`. It does this by repeatedly setting `x` to the index of an occurrence of `b` in `a`, then assigning `a` the result of the concatenation of the parts of `a` before and after the occurrence of `b`. The method call `abMethod("sing the song", "ng")` removes all occurrences of `"ng"` from `"sing the song"`, returning `"si the so"`.
(C)	Incorrect. This value would be returned if the statements inside the `while` loop were executed only one time.
(D)	Incorrect. This value would be returned if the first statement in the `while` loop was `a = a.substring(0, x) + a.substring(x + b.length() - 1)`.
(E)	Incorrect. The method does not attempt to access a substring that begins at an index less than `0` or ends at an index greater than `length - 1`.

Question 29

Skill		Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.		CON-2.O: Determine the result of executing recursive methods.	Recursion
(A)	Incorrect. This would be the result if the value returned in cases other than the base case was `calcMethod(num / 2)`.		
(B)	Incorrect. This would be the result if the value returned in cases other than the base case was `num + calcMethod(num % 2)`.		
(C)	Incorrect. This would be the result if the value returned in the base case was `0` instead of `10`.		
(D)	Incorrect. This would be the result if the base case was `num == 2` instead of `num == 0`.		
(E)	**Correct.** The result of the method call `calcMethod(16)` is `16 + calcMethod(8)`. The result of the method call `calcMethod(8)` is `8 + calcMethod(4)`. The recursive calls continue until the call `calcMethod(0)`, which returns the value `10`. The result of the method call `calcMethod(16)` can be calculated as 16+8+4+2+1+10=41.		

Question 30

Skill		Learning Objective	Topic
1.C: Determine code that would be used to interact with completed program code.		MOD-3.B: Create an inheritance relationship from a subclass to the superclass.	Writing Constructors for Subclasses
(A)	Incorrect. Calling the no-argument superclass constructor would initialize both `height` and `width` to `1`, which is not necessarily the intended behavior.		
(B)	**Correct.** A call to the one-argument superclass constructor with the single parameter `x` will set both the `height` and the `width` instance variables to `x`.		
(C)	Incorrect. This statement would result in a compiler error. Constructors cannot be called by name from other constructors. The `super` keyword is used to call the constructor of the superclass.		
(D)	Incorrect. This statement would result in a compiler error. Constructors cannot be called by name from other constructors. In addition, the `Square` class does not define a constructor with a matching signature.		
(E)	Incorrect. This code segment would result in a compiler error. The instance variables `height` and `width` are defined as `private` in the superclass and cannot be accessed directly from the subclass.		

Question 31

Skill	Learning Objective	Topic
1.B: Determine code that would be used to complete code segments.	VAR-2.B: Traverse the elements in a 1D array. VAR-2.C: Traverse the elements in a 1D array object using an enhanced for loop. CON-2.I: For algorithms in the context of a particular specification that requires the use of array traversals—a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	Traversing Arrays Enhanced for Loop for Arrays Developing Algorithms Using Arrays

(A)	**Correct.** In code segment I, `i` takes on the values `-1` through `nums.length - 2`, inclusive, in the `while` loop. Since `i` is incremented before the `if` statement, the array elements `nums[0]` through `nums[nums.length - 1]` are compared to `0`. In code segment II, array element `nums[0]` is excluded since the first iteration of the `for` loop accesses `nums[1]`. In code segment III, the variable `i` represents an element of the array rather than an index.
(B)	Incorrect. In code segment I, `i` takes on the values `-1` through `nums.length - 2`, inclusive, in the `while` loop. Since `i` is incremented before the `if` statement, the array elements `nums[0]` through `nums[nums.length - 1]` are compared to `0`. In code segment II, array element `nums[0]` is excluded since the first iteration of the `for` loop accesses `nums[1]`.
(C)	Incorrect. In code segment II, array element `nums[0]` is excluded since the first iteration of the `for` loop accesses `nums[1]`.
(D)	Incorrect. In code segment III, the variable `i` represents an element of the array rather than an index.
(E)	Incorrect. In code segment II, array element `nums[0]` is excluded since the first iteration of the `for` loop accesses `nums[1]`. In code segment III, the variable `i` represents an element of the array rather than an index.

Question 32

Skill	Learning Objective	Topic
2.C: Determine the result or output based on the statement execution order in a code segment containing method calls.	MOD-3.D: Call methods in an inheritance relationship. MOD-2.B: Define instance variables for the attributes to be initialized through the constructors of a class. MOD-3.B: Create an inheritance relationship from a subclass to the superclass.	Polymorphism Constructors Creating Superclasses and Subclasses

(A)	Incorrect. This output would be printed if `obj` was instantiated by calling the `ClassA` constructor instead of the `ClassB` constructor.
(B)	**Correct.** Since `obj` is instantiated as a `ClassB` object but the `showValue` method is not defined in `ClassB`, the `showValue` method call accesses the `showValue` method in the superclass, `ClassA`. Since the `getValue` method is defined in `ClassB`, the `getValue` method call accesses the `getValue` method in the subclass, `ClassB`, and `"B"` is printed.
(C)	Incorrect. This output would be printed if the `getValue` method in `ClassB` returned `super.getValue() + "B"` instead of `"B"`. The value returned by the `getValue` method of `ClassA ("A")` would be concatenated with `"B"` and the `String "AB"` would be printed by the `showValue` method.
(D)	Incorrect. This output would be printed if the `getValue` method in `ClassB` returned `"B" + super.getValue()` instead of `"B"`. The String `"B"` would be concatenated with the value returned by the `getValue` method of `ClassA` and the `String "BA"` would be printed by the `showValue` method.
(E)	Incorrect. The code compiles without error since `obj` is declared as an object of type `ClassA` and the `showValue` method is defined in `ClassA`.

Question 33

Skill		Learning Objective	Topic
2.B: Determine the result or output based on statement execution order in a code segment without method calls (other than output).		VAR-2.G: For 2D array objects— a. Traverse using nested for loops. b. Traverse using nested enhanced for loops.	Traversing 2D Arrays
(A)	Incorrect. This output would be generated if `col` and `row` were initialized to `0` and `col`, respectively, in the `for` loops.		
(B)	Incorrect. This output would be generated if `col` and `row` were initialized to `1` and `0`, respectively, in the `for` loops.		
(C)	Incorrect. This output would be generated if `col` and `row` were initialized to `0` and `1`, respectively, in the `for` loops.		
(D)	Incorrect. This output would be generated if the positions of the two `for` loop headers were reversed, with `row` as the loop control variable in the outer `for` loop and `col` as the loop control variable in the inner `for` loop.		
(E)	**Correct.** The code segment performs a column-major traversal of the array, beginning with the second column and the second row. For each column, all values in that column after the value in the first row are printed on a single line. The `println` method call causes data from subsequent columns to appear on new lines in the output.		

Question 34

Skill	Learning Objective	Topic
5.B: Explain why a code segment will not compile or work as intended.	VAR-2.E: For ArrayList objects— a. Traverse using a for or while loop. B. Traverse using an enhanced for loop.	Traversing ArrayLists

(A)	Incorrect. The order of addition and removal does not matter, since the value to be added to `returnList` is stored in a separate variable `num` prior to its removal from `numList`.
(B)	Incorrect. When there are no matches, the method returns an empty `ArrayList`.
(C)	Incorrect. The `get` and `remove` methods are always called with a valid index, since the body of the `while` loop is executed for values of `i` between `0` and the current value of `numList.size() - 1`.
(D)	Incorrect. The expression `num % key == 0` is correctly used to identify values of `num` that are divisible by `key`.
(E)	**Correct.** When the element at position `i` is removed from `numList`, subsequent elements are shifted left. After the removal, the element that used to be at position `i + 1` is now at position `i`. Because the method increments `i` regardless of whether the element at position `i` was removed, the method does not always work as intended. For example, if two adjacent elements are both divisible by `key`, only the first element is removed. The method could be corrected by incrementing `i` only when the element at position `i` is not removed or by decrementing `i` when an element is removed.

Question 35

Skill	Learning Objective	Topic
1.B: Determine code that would be used to complete code segments.	CON-2.I: For algorithms in the context of a particular specification that requires the use of array traversals—a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	Developing Algorithms Using Arrays

(A)	**Correct.** For each element `arr[j]`, the inner `for` loop counts the number of times that `arr[j]` appears in `arr` and stores the result in `valCount`. It does this by evaluating the condition `arr[j] == arr[k]` and incrementing `valCount` when the condition evaluates to `true`. After the inner loop completes, the method evaluates the condition `valCount > modeCount`. If the condition evaluates to `true`, a new mode has been found and `mode` and `modeCount` are updated.
(B)	Incorrect. The replacement for `/* missing condition 2 */` is incorrect. It would update the mode value to be returned only if it was less common than values considered previously.
(C)	Incorrect. The replacement for `/* missing condition 1 */` is incorrect. Instead of counting the number of times that `arr[j]` appears in `arr`, it would count the number of times that values different than `arr[j]` appear in `arr`.
(D)	Incorrect. The replacements for `/* missing condition 1 */` and `/* missing condition 2 */` are incorrect. Because of the incorrect replacement for `/* missing condition 1 */`, the method would count the number of times that values different than `arr[j]` appear in `arr` instead of the number of times that `arr[j]` appears in `arr`. Because of the incorrect replacement for `/* missing condition 2 */`, the method would update the mode value to be returned only if it was less common than values considered previously.
(E)	Incorrect. The replacements for `/* missing condition 1 */` and `/* missing condition 2 */` are incorrect. Because of the incorrect replacement for `/* missing condition 1 */`, the method would count the number of times that values different than `arr[j]` appear in `arr` instead of the number of times that `arr[j]` appears in `arr`. Because of the incorrect replacement for /* *missing condition 2* `*/`, the method would update the mode value to be returned only if it occurred either more or less frequently than values considered previously.

Question 36

Skill	Learning Objective	Topic
5.D: Describe the initial conditions that must be met for a program segment to work as intended or described.	CON-2.E: Represent iterative processes using a for loop. CON-2.C: Represent iterative processes using a while loop. CON-1.A: Evaluate arithmetic expressions in program code.	for Loops while Loops Expressions and Assignment Statements

(A)	Incorrect. When `a` has the value `6` and `b` has the value `4`, `methodOne` returns `1` and `methodTwo` returns `2`. In general, if `a % b` is not equal to `0`, `methodOne` returns `a / b` and `methodTwo` returns `a / b + 1`.
(B)	Incorrect. When `a` has the value `21` and `b` has the value `5`, `methodOne` returns `4` and `methodTwo` returns `5`. In general, if `a % b` is not equal to `0`, `methodOne` returns `a / b` and `methodTwo` returns `a / b + 1`.
(C)	Incorrect. When `a` has the value `10` and `b` has the value `3`, `methodOne` returns `3` and `methodTwo` returns `4`. In general, if `a % b` is not equal to `0`, `methodOne` returns `a / b` and `methodTwo` returns `a / b + 1`.
(D)	**Correct.** The body of the `for` loop in `methodOne` is executed `a / b` times. The body of the `while` loop in `methodTwo` is executed `a / b` times only when `a % b` is equal to `0`. When `a % b` is not equal to `0`, the body of the `while` loop in `methodTwo` is executed an additional time. For example, when `a` has the value `11` and `b` has the value `5`, `a / b` evaluates to `2` and the `for` loop is executed two times but the `while` loop is executed three times.
(E)	Incorrect. When `a` has the value `7` and `b` has the value `3`, `methodOne` returns `2` and `methodTwo` returns `3`. In general, if `a % b` is not equal to `0`, `methodOne` returns `a / b` and `methodTwo` returns `a / b + 1`.

Question 37

Skill	Learning Objective	Topic
5.A: Describe the behavior of a given segment of program code.	CON-2.C: Represent iterative processes using a while loop. CON-2.D: For algorithms in the context of a particular specification that does not require the use of traversals— a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	while Loops

(A)	Incorrect. The product of `num2` and `num3` could be computed by an algorithm that adds `num2` to `num1` a total of `num3` times.
(B)	Incorrect. The product of `num2` and `num3 - 1` could be computed by an algorithm that adds `num2` to `num1` a total of `num3 - 1` times.
(C)	Incorrect. The sum of `num2` and `num3` could be computed by an algorithm that adds `1` to `num2` a total of `num3` times.
(D)	Incorrect. This would correctly describe the behavior of the code segment if the Boolean condition in the `while` loop was `num2 <= num3`.
(E)	**Correct.** Each iteration of the `while` loop adds `num2` to `num1` and then increments `num2`. The last value assigned to `num2` and added to `num1` is `num3 - 1`. Since each value of `num2` is added to `num1`, the code segment computes the sum of the integers from `num2` to `num3 - 1`.

Question 38

Skill	Learning Objective	Topic
1.C: Determine code that would be used to interact with completed program code.	VAR-2.D: Represent collections of related object reference data using ArrayList objects. VAR-2.E: For ArrayList objects— a. Traverse using a for or while loop. b. Traverse using an enhanced for loop.	ArrayList Methods Traversing ArrayLists

(A)	Incorrect. Option II is correct. The code segment uses an enhanced `for` loop to traverse the `valueList` array. The statement inside the loop calls the `getNum` method to access the `num` instance variable.
(B)	Incorrect. Option I is correct. The code segment uses a `for` loop to traverse the `valueList` array. The statement inside the loop calls the `get` method to access a `Value` object and then calls the `getNum` method to access the `num` instance variable.
(C)	Incorrect. Option I is correct. The code segment uses a `for` loop to traverse the `valueList` array. The statement inside the loop calls the `get` method to access a `Value` object and then calls the `getNum` method to access the `num` instance variable. Option II is correct. The code segment uses an enhanced `for` loop to traverse the `valueList` array. The statement inside the loop calls the `getNum` method to access the `num` instance variable. Option III is incorrect. The code segment causes a compilation error because the `getNum` method must be called using the dot operator, not by passing the object reference as an argument.
(D)	**Correct.** Option I is correct. The code segment uses a `for` loop to traverse the `valueList` array. The statement inside the loop calls the `get` method to access a `Value` object and then calls the `getNum` method to access the `num` instance variable. Option II is correct. The code segment uses an enhanced `for` loop to traverse the `valueList` array. The statement inside the loop calls the `getNum` method to access the `num` instance variable. Option III is incorrect. The code segment causes a compilation error because the `getNum` method must be called using the dot operator, not by passing the object reference as an argument.
(E)	Incorrect. Option II is correct. The code segment uses an enhanced `for` loop to traverse the `valueList` array. The statement inside the loop calls the `getNum` method to access the `num` instance variable. Option III is incorrect. The code segment causes a compilation error because the `getNum` method must be called using the dot operator, not by passing the object reference as an argument.

Question 39

Skill		Learning Objective	Topic
4.A: Use test-cases to find errors or validate results.		CON-2.O: Determine the result of executing recursive methods.	Recursion
(A)	Incorrect. This method call returns `false` because the first character is lexicographically less than the second character of the string.		
(B)	Incorrect. This method call returns `false` because the first character is lexicographically less than the second character of the string.		
(C)	Incorrect. This method call returns `false` because the first character is lexicographically less than the second character of the string.		
(D)	**Correct.** If the first character of `str` is lexicographically greater than the second character of `str`, the method returns the result of the recursive call with a parameter that contains all but the first character of `str`. If the first character of `str` is lexicographically less than or equal to the second character of `str`, the method returns `false`. If no such character pair (where the first character of `str` is lexicographically less than or equal to the second character of `str`) is found, the base case is reached and the value `true` is returned.		
(E)	Incorrect. This method call returns `false` because the third character is lexicographically less than the fourth character of the string.		

Question 40

Skill	Learning Objective	Topic
5.C: Explain how the result of program code changes, given a change to the initial code.	MOD-3.D: Call methods in an inheritance relationship.	Polymorphism

(A)	**Correct.** At compile time, methods in or inherited by the declared type determine the correctness of a non-static method call. In line 1, `obj1` is declared as an object of type `A`. Therefore, at compile time, there must be a `message` method in class `A` or its superclass. If the `message` method in class `A` is removed, the statement in line 3 will no longer compile.
(B)	Incorrect. At compile time, methods in or inherited by the declared type determine the correctness of a non-static method call. In line 2, `obj2` is declared as an object of type `B`. Therefore, at compile time, there must be a `message` method in class `B` or its superclass. Since the `message` method in class `B` has not been removed, this statement will not cause a compiler error.
(C)	Incorrect. The `message` method in class `A` is not executed in the original code segment. At run-time, the method in the actual object type is executed for a non-static method call. In line 1, `obj1` is instantiated as an object of type `B`. Therefore, in line 3 at run-time, the `message` method of class `B` rather than the `message` method of class `A` is executed.
(D)	Incorrect. The `message` method in class `A` is not executed in the original code segment. At run-time, the method in the actual object type is executed for a non-static method call. In line 2, `obj2` is instantiated as an object of type `B`. Therefore, in line 4 at run-time, the `message` method of class `B` rather than the `message` method of class `A` is executed.
(E)	Incorrect. At compile time, methods in or inherited by the declared type determine the correctness of a non-static method call. In line 1, `obj1` is declared as an object of type `A`. Therefore, at compile time, there must be a `message` method in class `A` or its superclass. If the `message` method in class `A` is removed, the code will no longer compile.

Answer Key and Question Alignment to Course Framework

Multiple-Choice Question	Answer	Skill	Learning Objective	Topic
1	C	2.B	CON-1.A	Expressions and Assignment Statements
2	D	5.A	CON-2.B	Compound Boolean Expressions
			CON-2.A	if-else Statements
3	C	1.C	CON-1.D	Using the Math Class
			CON-1.A	Expressions and Assignment Statements
			CON-1.C	Casting and Ranges of Variables
4	D	2.B	CON-2.B	Compound Boolean Expressions
			CON-2.A	if Statements and Control Flow if-else Statements
5	E	1.C	MOD-1.C	Creating and Storing Objects (Instantiation)
6	A	2.A	CON-1.A	Expressions and Assignment Statements
			CON-1.C	Casting and Ranges of Variables
7	E	4.C	CON-2.B	Compound Boolean Expressions
			CON-2.A	if-else Statements
			CON-1.F	Compound Boolean Expressions
8	D	4.B	VAR-1.G	Scope and Access
			MOD-1.C	Creating and Storing Objects (Instantiation)
9	B	2.C	VAR-2.D	ArrayList Methods
10	D	4.A	CON-2.E	for Loops
			CON-1.A	Expressions and Assignment Statements
11	C	1.B	CON-2.C	while Loops
			CON-1.B	Compound Assignment Operators
12	B	1.C	CON-1.H	Comparing Objects
			CON-2.A	if Statements and Control Flow
13	E	4.C	CON-1.G	Equivalent Boolean Expressions
14	D	1.B	CON-2.A	if Statements and Control Flow else if Statements
15	E	5.A	CON-1.D	Using the Math Class
16	B	2.C	CON-2.O	Recursion
			CON-2.E	for Loops

Multiple-Choice Question	Answer	Skill	Learning Objective	Topic
17	D	2.C	MOD-3.B	Overriding Methods super Keyword
			MOD-3.D	Polymorphism
18	E	1.B	VAR-2.B	Traversing Arrays
			CON-2.E	for Loops
19	D	2.B	VAR-2.G	Traversing 2D Arrays
			VAR-2.F	2D Arrays
20	B	1.C	VAR-2.C	Enhanced for Loop for Arrays
			CON-2.I	Developing Algorithms Using Arrays
21	D	2.C	CON-2.I	Developing Algorithms Using Arrays
			VAR-1.E	String Methods
22	A	2.D	CON-2.I	Developing Algorithms Using Arrays
			VAR-1.E	String Methods
23	D	1.B	VAR-2.G	Traversing 2D Arrays
24	D	2.C	MOD-2.H	Static Variables and Methods
25	D	2.C	CON-2.F	Developing Algorithms Using Strings
			VAR-1.E	String Methods
			CON-2.E	for Loops
26	C	2.D	CON-2.H	Informal Code Analysis
			CON-2.E	for Loops
			CON-2.G	Nested Iteration
27	E	5.D	CON-1.G	Equivalent Boolean Expressions
28	B	2.C	CON-2.F	Developing Algorithms Using Strings
			VAR-1.E	String Methods
			CON-2.C	while Loops
29	E	2.C	CON-2.O	Recursion
30	B	1.C	MOD-3.B	Writing Constructors for Subclasses
31	A	1.B	VAR-2.B	Traversing Arrays
			VAR-2.C	Enhanced for Loop for Arrays
			CON-2.I	Developing Algorithms Using Arrays
32	B	2.C	MOD-3.D	Polymorphism
			MOD-2.B	Constructors
			MOD-3.B	Creating Superclasses and Subclasses
33	E	2.B	VAR-2.G	Traversing 2D Arrays

Multiple-Choice Question	Answer	Skill	Learning Objective	Topic
34	E	5.B	VAR-2.E	Traversing ArrayLists
35	A	1.B	CON-2.I	Developing Algorithms Using Arrays
36	D	5.D	CON-2.E CON-2.C CON-1.A	for Loops while Loops Expressions and Assignment Statements
37	E	5.A	CON-2.C CON-2.D	while Loops
38	D	1.C	VAR-2.D VAR-2.E	ArrayList Methods Traversing ArrayLists
39	D	4.A	CON-2.O	Recursion
40	A	5.C	MOD-3.D	Polymorphism