

Unit 3: Boolean (Part A)

1. Consider the following code segment.

```
String str1 = new String("Advanced Placement");
String str2 = new String("Advanced Placement");
if (str1.equals(str2) && str1 == str2)
{
    System.out.println("A");
}
else if (str1.equals(str2) && str1 != str2)
{
    System.out.println("B");
}
else if (!str1.equals(str2) && str1 == str2)
{
    System.out.println("C");
}
else if (!str1.equals(str2) && str1 != str2)
{
    System.out.println("D");
}
```

What, if anything, is printed when the code segment is executed?

- (A) A
 - (B) B
 - (C) C
 - (D) D
 - (E) Nothing is printed.
2. Consider the following method that is intended to determine if the double values `d1` and `d2` are close enough to be considered equal. For example, given a tolerance of 0.001, the values 54.32271 and 54.32294 would be considered equal.

```
/** @return true if d1 and d2 are within the specified tolerance,
 *     false otherwise
 */
public boolean almostEqual(double d1, double d2, double tolerance)
{
    /* missing code */
}
```

Which of the following should replace `/* missing code */` so that `almostEqual` will work as intended?

Unit 3: Boolean (Part A)

- (A) `return (d1 - d2) <= tolerance;`
- (B) `return ((d1 + d2) / 2) <= tolerance;`
- (C) `return (d1 - d2) >= tolerance;`
- (D) `return ((d1 + d2) / 2) >= tolerance;`
- (E) `return Math.abs(d1 - d2) <= tolerance;`

3. Assume that `a` and `b` have been defined and initialized as `int` values. The expression

`!(! (a != b) && (b > 7))`

is equivalent to which of the following?

- (A) `(a != b) || (b < 7)`
- (B) `(a != b) || (b <= 7)`
- (C) `(a == b) || (b <= 7)`
- (D) `(a != b) && (b <= 7)`
- (E) `(a == b) && (b > 7)`

4. Assume that the boolean variables `a`, `b`, `c`, and `d` have been declared and initialized. Consider the following expression.

`!(!(a && b) || (c || !d))`

Which of the following is equivalent to the expression?

- (A) `(a && b) && (!c && d)`
- (B) `(a || b) && (!c && d)`
- (C) `(a && b) || (c || !d)`
- (D) `(!a || !b) && (!c && d)`
- (E) `!(a && b) && (c || !d)`

5. Assume that `x` and `y` are boolean variables and have been properly initialized.

`(x && y) && !(x || y)`

Which of the following best describes the result of evaluating the expression above?

Unit 3: Boolean (Part A)

- (A) true always
 - (B) false always
 - (C) true only when x is true and y is true
 - (D) true only when x and y have the same value
 - (E) true only when x and y have different values
6. Assume that x and y are boolean variables and have been properly initialized.

```
(x || y) && x
```

Which of the following always evaluates to the same value as the expression above?

- (A) x
 - (B) y
 - (C) x && y
 - (D) x || y
 - (E) x != y
7. Assume that x and y have been declared and initialized with int values. Consider the following Java expression.

```
(y > 10000) || (x > 1000 && x < 1500)
```

Which of the following is equivalent to the expression given above?

- (A) (y > 10000 || x > 1000) && (y > 10000 || x < 1500)
- (B) (y > 10000 || x > 1000) || (y > 10000 || x < 1500)
- (C) (y > 10000) && (x > 1000 || x < 1500)
- (D) (y > 10000 && x > 1000) || (y > 10000 && x < 1500)
- (E) (y > 10000 && x > 1000) && (y > 10000 && x < 1500)

Unit 3: Boolean (Part A)

8. At a certain high school students receive letter grades based on the following scale.

<u>Integer Score</u>	<u>Letter Grade</u>
93 or above	A
From 84 to 92 inclusive	B
From 75 to 83 inclusive	C
Below 75	F

Which of the following code segments will assign the correct string to grade for a given integer score ?

- I.

```
if (score >= 93)
    grade = "A";
if (score >= 84 && score <= 92)
    grade = "B";
if (score >= 75 && score <= 83)
    grade = "C";
if (score < 75)
    grade = "F";
```
- II.

```
if (score >= 93)
    grade = "A";
if (84 <= score <= 92)
    grade = "B";
if (75 <= score <= 83)
    grade = "C";
if (score < 75)
    grade = "F";
```
- III.

```
if (score >= 93)
    grade = "A";
else if (score >= 84)
    grade = "B";
else if (score >= 75)
    grade = "C";
else
    grade = "F";
```

Unit 3: Boolean (Part A)

- (A) II only
 - (B) III only
 - (C) I and II only
 - (D) I and III only
 - (E) I, II, and III
9. Consider the following method, `between`, which is intended to return `true` if `x` is between `lower` and `upper`, inclusive, and `false` otherwise.

```
// precondition: lower <= upper

// postcondition: returns true if x is between lower and upper,
//                inclusive; otherwise, returns false

public boolean between(int x, int lower, int upper)
{
    /* missing code */
}
```

Which of the following can be used to replace `/* missing code */` so that `between` will work as intended?

- (A) `return (x <= lower) && (x >= upper);`
- (B) `return (x <= lower) || (x >= upper);`
- (C) `return lower <= x <= upper;`
- (D) `return (x >= lower) && (x <= upper);`
- (E) `return (x >= lower) || (x <= upper);`

Unit 3: Boolean (Part A)

10. Consider the following method, `biggest`, which is intended to return the greatest of three integers. It does not always work as intended.

```
public static int biggest(int a, int b, int c)
{
    if ((a > b) && (a > c))
    {
        return a;
    }
    else if ((b > a) && (b > c))
    {
        return b;
    }
    else
    {
        return c;
    }
}
```

Which of the following best describes the error in the method?

- (A) `biggest` always returns the value of `a`.
- (B) `biggest` may not work correctly when `c` has the greatest value.
- (C) `biggest` may not work correctly when `a` and `b` have equal values.
- (D) `biggest` may not work correctly when `a` and `c` have equal values.
- (E) `biggest` may not work correctly when `b` and `c` have equal values.

Unit 3: Boolean (Part A)

11. A teacher put three bonus questions on a test and awarded 5 extra points to anyone who answered all three bonus questions correctly and no extra points otherwise. Assume that the `boolean` variables `bonusOne`, `bonusTwo`, and `bonusThree` indicate whether a student has answered the particular question correctly. Each variable was assigned `true` if the answer was correct and `false` if the answer was incorrect.

Which of the following code segments will properly update the variable `grade` based on a student's performance on the bonus questions?

I. `if (bonusOne && bonusTwo && bonusThree)`

`grade += 5;`

II. `if (bonusOne || bonusTwo || bonusThree)`

`grade += 5;`

III. `if (bonusOne)`

`grade += 5;`

`if (bonusTwo)`

`grade += 5;`

`if (bonusThree)`

`grade += 5;`

- (A) I only
(B) II only
(C) III only
(D) I and III
(E) II and III

Unit 3: Boolean (Part A)

12. Consider the following class definitions.

```
public class Person
{
    private String name;
    public String getName()
    { return name; }
}
public class Book
{
    private String author;
    private String title;
    private Person borrower;
    public Book(String a, String t)
    {
        author = a;
        title = t;
        borrower = null;
    }
    public void printDetails()
    {
        System.out.print("Author: " + author + " Title: " + title);
        if ( /* missing condition */ )
        {
            System.out.println(" Borrower: " + borrower.getName());
        }
    }
    public void setBorrower(Person b)
    { borrower = b; }
}
```

Which of the following can replace `/* missing condition */` so that the `printDetails` method CANNOT cause a run-time error?

- I. `!borrower.equals(null)`
- II. `borrower != null`
- III. `borrower.getName() != null`

- (A) I only
- (B) II only
- (C) III only
- (D) I and II
- (E) II and III

13. Assume that `a`, `b`, and `c` are boolean variables that have been properly declared and initialized. Which of the following boolean expressions is equivalent to `!(a && b) || c` ?

Unit 3: Boolean (Part A)

- (A) `a && b && c`
- (B) `a || b || c`
- (C) `!a && !b || c`
- (D) `!a && !b && c`
- (E) `!a || !b || c`

Unit 3: Boolean (Part A)

14. Consider the following class declaration

```
public class SomeClass
{
    private int num;

    public SomeClass(int n)
    {
        num = n;
    }

    public void increment(int more)
    {
        num = num + more;
    }

    public int getNum()
    {
        return num;
    }
}
```

The following code segment appears in another class.

```
SomeClass one = new SomeClass(100);
```

Unit 3: Boolean (Part A)

```
SomeClass two = new SomeClass(100);
```

```
SomeClass three = one;
```

```
one.increment(200);
```

```
System.out.println(one.getNum() + " " + two.getNum() + " " +  
                    three.getNum());
```

What is printed as a result of executing the code segment?

- (A) 100 100 100
- (B) 300 100 100
- (C) 300 100 300
- (D) 300 300 100
- (E) 300 300 300

Unit 3: Boolean (Part A)

15. Consider the following code segment, which is intended to simulate a random process. The code is intended to set the value of the variable `event` to exactly one of the values 1, 2, or 3, depending on the probability of an event occurring. The value of `event` should be set to 1 if the probability is 70 percent or less. The value of `event` should be set to 2 if the probability is greater than 70 percent but no more than 80 percent. The value of `event` should be set to 3 if the probability is greater than 80 percent. The variable `randomNumber` is used to simulate the probability of the event occurring.

```
int event = 0;
if (randomNumber <= 0.70)
{
    event = 1;
}
if (randomNumber <= 0.80)
{
    event = 2;
}
else
{
    event = 3;
}
```

The code does not work as intended. Assume that the variable `randomNumber` has been properly declared and initialized. Which of the following initializations for `randomNumber` will demonstrate that the code segment will not work as intended?

- (A) `randomNumber = 0.70;`
 - (B) `randomNumber = 0.80;`
 - (C) `randomNumber = 0.85;`
 - (D) `randomNumber = 0.90;`
 - (E) `randomNumber = 1.00;`
16. Consider the following Boolean expressions.

A && B

I.

!A && !B

II.

Which of the following best describes the relationship between values produced by expression I and expression II?

- (A) Expression I and expression II evaluate to different values for all values of `A` and `B`.
- (B) Expression I and expression II evaluate to the same value for all values of `A` and `B`.
- (C) Expression I and expression II evaluate to the same value only when `A` and `B` are the same.
- (D) Expression I and expression II evaluate to the same value only when `A` and `B` differ.
- (E) Expression I and expression II evaluate to the same value whenever `A` is true.

Unit 3: Boolean (Part A)

17. Consider the following two code segments where the `int` variable `choice` has been properly declared and initialized.

Code Segment A

```
if (choice > 10)
{
    System.out.println("blue");
}
else if (choice < 5)
{
    System.out.println("red");
}
else
{
    System.out.println("yellow");
}
```

Code Segment B

```
if (choice > 10)
{
    System.out.println("blue");
}
if (choice < 5)
{
    System.out.println("red");
}
else
{
    System.out.println("yellow");
}
```

Assume that both code segments initialize `choice` to the same integer value. Which of the following best describes the conditions on the initial value of the variable `choice` that will cause the two code segments to produce different output?

- (A) `choice < 5`
- (B) `choice >= 5` and `choice <= 10`
- (C) `choice > 10`
- (D) `choice == 5` or `choice == 10`
- (E) There is no value for `choice` that will cause the two code segments to produce different output.

Unit 3: Boolean (Part A)

18. Consider the following code segments, which are each intended to convert grades from a 100-point scale to a 4.0-point scale and print the result. A grade of 90 or above should yield a 4.0, a grade of 80 to 89 should yield a 3.0, a grade of 70 to 79 should yield a 2.0, and any grade lower than 70 should yield a 0.0.

Assume that `grade` is an `int` variable that has been properly declared and initialized.

Code Segment I

```
double points = 0.0;
if (grade > 89)
{
    points += 4.0;
}
else if (grade > 79)
{
    points += 3.0;
}
else if (grade > 69)
{
    points += 2.0;
}
else
{
    points += 0.0;
}
System.out.println(points);
```

Code Segment II

```
double points = 0.0;
if (grade > 89)
{
    points += 4.0;
}
if (grade > 79)
{
    grade += 3.0;
}
if (grade > 69)
{
    points += 2.0;
}
if (grade < 70)
{
    points += 0.0;
}
System.out.println(points);
```

Which of the following statements correctly compares the values printed by the two methods?

Unit 3: Boolean (Part A)

- (A) The two code segments print the same value only when `grade` is below 80.
- (B) The two code segments print the same value only when `grade` is 90 or above or `grade` is below 80.
- (C) The two code segments print the same value only when `grade` is 90 or above.
- (D) Both code segments print the same value for all possible values of `grade`.
- (E) The two code segments print different values for all possible values of `grade`.
19. Consider the following incomplete method, which is intended to return `true` if the value of `y` is between the values of the other two parameters and `false` otherwise.

```
/** Precondition: x, y, and z have 3 different values. */
public static boolean compareThree(int x, int y, int z)
{
    return /* missing condition */ ;
}
```

The following table shows the results of several calls to `compareThree`.

Call	Result
<code>compareThree(4, 5, 6)</code>	<code>true</code>
<code>compareThree(6, 5, 4)</code>	<code>true</code>
<code>compareThree(5, 4, 6)</code>	<code>false</code>
<code>compareThree(3, 4, 4)</code>	violates precondition

Which of the following can be used to replace `/* missing condition */` so that `compareThree` will work as intended when called with parameters that satisfy its precondition?

- (A) `(x > y) && (x > z)`
- (B) `(x > y) && (y > z)`
- (C) `(x > y) || (y > z)`
- (D) `(x > y) == (y > z)`
- (E) `(x > y) != (y > z)`

Unit 3: Boolean (Part A)

20. Consider the following code segment in which the `int` variable `x` has been properly declared and initialized.

```
if (x % 2 == 1)
{
    System.out.println("YES");
}
else
{
    System.out.println("NO");
}
```

Assuming that `x` is initialized to the same positive integer value as the original, which of the following code segments will produce the same output as the original code segment?

```
if (x % 2 == 1)
{
    System.out.println("YES");
}
if (x % 2 == 0)
{
    System.out.println("NO");
}
```

I.

```
if (x % 2 == 1)
{
    System.out.println("YES");
}
else if (x % 2 == 0)
{
    System.out.println("NO");
}
else
{
    System.out.println("NONE");
}
```

II.

```
boolean test = x % 2 == 0;
if (test)
{
    System.out.println("YES");
}
```

III.

Unit 3: Boolean (Part A)

```
else
{
    System.out.println("NO");
}
```

- (A) I only
 - (B) II only
 - (C) III only
 - (D) I and II only
 - (E) I, II, and III
21. Assume `obj1` and `obj2` are object references. Which of the following best describes when the expression `obj1 == obj2` is true?
- (A) When `obj1` and `obj2` are defined within the same method
 - (B) When `obj1` and `obj2` are instances of the same class
 - (C) When `obj1` and `obj2` refer to objects that contain the same data
 - (D) When `obj1` and `obj2` refer to the same object
 - (E) When `obj1` and `obj2` are private class variables defined in the same class

22. Consider the following declarations.

```
int valueOne, valueTwo;
```

Assume that `valueOne` and `valueTwo` have been initialized. Which of the following evaluates to `true` if `valueOne` and `valueTwo` contain the same value?

- (A) `valueOne.equals((Object) valueTwo)`
 - (B) `valueOne == valueTwo`
 - (C) `valueOne.compareTo((Object) valueTwo) == 0`
 - (D) `valueOne.compareTo(valueTwo) == 0`
 - (E) `valueOne.equals(valueTwo)`
23. Which of the following best describes the value of the Boolean expression shown below?
- ```
a && !(b || a)
```
- (A) The value is always `true`.
  - (B) The value is always `false`.
  - (C) The value is `true` when `a` has the value `false`, and is `false` otherwise.
  - (D) The value is `true` when `b` has the value `false`, and is `false` otherwise.
  - (E) The value is `true` when either `a` or `b` has the value `true`, and is `false` otherwise.

**Unit 3: Boolean (Part A)**

24. Assume that `a`, `b`, and `c` are variables of type `int`. Consider the following three conditions.

I. `(a == b) && (a == c) && (b == c)`

II. `(a == b) || (a == c) || (b == c)`

III. `((a - b) * (a - c) * (b - c) == 0)`

Assume that subtraction and multiplication never overflow. Which of the conditions above is (are) always true if at least two of `a`, `b`, and `c` are equal?

- (A) I only
- (B) II only
- (C) III only
- (D) I and II
- (E) II and III

25. Assume that `x` and `y` are `boolean` variables and have been properly initialized.

$$(x \ \&\& \ y) \ || \ !(x \ \&\& \ y)$$

The result of evaluating the expression above is best described as

- (A) always true
- (B) always false
- (C) true only when `x` is true and `y` is true
- (D) true only when `x` and `y` have the same value
- (E) true only when `x` and `y` have different values

26. Consider the following code segment.

```
int x = /* some integer value */ ;
```

```
int y = /* some integer value */ ;
```

```
boolean result = (x < y);
```

```
result = ((x >= y) && !result);
```

Which of the following best describes the conditions under which the value of `result` will be true after the code segment is executed?

**Unit 3: Boolean (Part A)**

- (A) Only when  $x < y$
- (B) Only when  $x \geq y$
- (C) Only when  $x$  and  $y$  are equal
- (D) The value will always be true.
- (E) The value will never be true.

27. Consider the following method.

```
public void conditionalTest(int a, int b)
{
 if ((a > 0) && (b > 0))
 {
 if (a > b)
 System.out.println("A");
 else
 System.out.println("B");
 }
 else if ((b < 0) || (a < 0))
 System.out.println("C");
 else
 System.out.println("D");
}
```

What is printed as a result of the call `conditionalTest(3, -2)`?

- (A) A
- (B) B
- (C) C
- (D) D
- (E) Nothing is printed.

**Unit 3: Boolean (Part A)**

28. Consider the following statement. Assume that `a` and `b` are properly declared and initialized `boolean` variables.

```
boolean c = (a && b) || (!a && b);
```

Under which of the following conditions will `c` be assigned the value `false` ?

- (A) Always
  - (B) Never
  - (C) When `a` and `b` have the same value
  - (D) When `a` has the value `false`
  - (E) When `b` has the value `false`
29. Consider the following code segment.

```
int x = 7;
int y = 3;

if ((x < 10) && (y < 0))
 System.out.println("Value is: " + x * y);
else
 System.out.println("Value is: " + x / y);
```

What is printed as a result of executing the code segment?

- (A) Value is: 21
- (B) Value is: 2.3333333
- (C) Value is: 2
- (D) Value is: 0
- (E) Value is: 1

**Unit 3: Boolean (Part A)**

30. Consider the following method.

```
public int pick(boolean test, int x, int y)
{
 if (test)
 return x;
 else
 return y;
}
```

What value is returned by the following method call?

`pick(false, pick(true, 0, 1), pick(true, 6, 7))`

- (A) 0
- (B) 1
- (C) 3
- (D) 6
- (E) 7

**Unit 3: Boolean (Part A)**

31. Consider the following method.

```
public void test(int x)
{
 int y;

 if (x % 2 == 0)
 y = 3;
 else if (x > 9)
 y = 5;
 else
 y = 1;

 System.out.println("y = " + y);
}
```

Which of the following test data sets would test each possible output for the method?

- (A) 8, 9, 12
- (B) 7, 9, 11
- (C) 8, 9, 11
- (D) 8, 11, 13
- (E) 7, 9, 10

32. Consider the following code segment.

```
String alpha = new String("APCS");
String beta = new String("APCS");
String delta = alpha;
System.out.println(alpha.equals(beta));
System.out.println(alpha == beta);
System.out.println(alpha == delta);
```

What is printed as a result of executing the code segment?

**Unit 3: Boolean (Part A)**

- (A) false  
false
- (B) false  
true
- (C) true  
false  
false
- (D) true  
false  
true
- (E) true  
true  
true
33. Assume that object references `one`, `two`, and `three` have been declared and instantiated to be of the same type. Assume also that `one == two` evaluates to `true` and that `two.equals(three)` evaluates to `false`.

Consider the following code segment.

```
if (one.equals(two))
{
 System.out.println("one dot equals two");
}
if (one.equals(three))
{
 System.out.println("one dot equals three");
}
if (two == three)
{
 System.out.println("two equals equals three");
}
```

What, if anything, is printed as a result of executing the code segment?

- (A) one dot equals two
- (B) one dot equals two  
one dot equals three
- (C) one dot equals three  
two equals equals three  
one dot equals two
- (D) one dot equals three  
two equals equals three
- (E) Nothing is printed.

**Unit 3: Boolean (Part A)**

34. Consider the following code segment.

```
if (a < b || c != d)
{
 System.out.println("dog");
}
else
{
 System.out.println("cat");
}
```

Assume that the `int` variables `a`, `b`, `c`, and `d` have been properly declared and initialized. Which of the following code segments produces the same output as the given code segment for all values of `a`, `b`, `c`, and `d` ?



**Unit 3: Boolean (Part A)**

- ```
if (a < b && c != d)
{
    System.out.println("dog");
}
(A) else
{
    System.out.println("cat");
}
if (a < b && c != d)
{
    System.out.println("cat");
}
(B) else
{
    System.out.println("dog");
}
if (a > b && c == d)
{
    System.out.println("cat");
}
(C) else
{
    System.out.println("dog");
}
if (a >= b || c == d)
{
    System.out.println("cat");
}
(D) else
{
    System.out.println("dog");
}
if (a >= b && c == d)
{
    System.out.println("cat");
}
(E) else
{
    System.out.println("dog");
}
```

35. Assume that `a`, `b`, `c`, and `d` have been declared and initialized with `int` values.

```
!((a >= b) && !(c < d))
```

Which of the following is equivalent to the expression above?

Unit 3: Boolean (Part A)

- (A) `(a < b) || (c < d)`
- (B) `(a < b) || (c >= d)`
- (C) `(a < b) && (c < d)`
- (D) `(a >= b) || (c < d)`
- (E) `(a >= b) && (c < d)`

36. Consider the following method.

```
public String exercise(int input)
{
    if (input < 10)
    {
        return "alpha";
    }
    if (input < 5)
    {
        return "beta";
    }
    if (input < 1)
    {
        return "gamma";
    }
    return "delta";
}
```

Assume that the `int` variable `x` has been initialized in another method in the same class. Which of the following describes the conditions under which the method call `exercise(x)` will return "gamma" ?

- (A) The method will never return "gamma".
 - (B) The method will return "gamma" if `x` is less than 1.
 - (C) The method will return "gamma" if `x` is between 1 and 4, inclusive.
 - (D) The method will return "gamma" if `x` is between 5 and 9, inclusive.
 - (E) The method will always return "gamma".
37. Consider the following statement, which assigns a value to `b1`.

```
boolean b1 = true && (17 % 3 == 1);
```

Which of the following assigns the same value to `b2` as the value stored in `b1` ?

- (A) `boolean b2 = false || (17 % 3 == 2);`
- (B) `boolean b2 = false && (17 % 3 == 2);`
- (C) `boolean b2 = true || (17 % 3 == 1);`
- (D) `boolean b2 = (true || false) && true;`
- (E) `boolean b2 = (true && false) || true;`

Unit 3: Boolean (Part A)

38. The question refer to the code from the GridWorld case study.

Consider the following code segment.

```
Location loc1 = new Location(3, 3);  
  
Location loc2 = new Location(3, 2);  
  
if (loc1.equals(loc2.getAdjacentLocation(Location.EAST)))  
    System.out.print("aaa");  
  
if (loc1.getRow() == loc2.getRow())  
    System.out.print("XXX");  
  
if (loc1.getDirectionToward(loc2) == Location.EAST)  
    System.out.print("555");
```

What will be printed as a result of executing the code segment?

- (A) aaaXXX555
 - (B) aaaXXX
 - (C) XXX555
 - (D) 555
 - (E) aaa
39. Assume that `x` and `y` are variables of type `int`. Which of the following Java expressions never results in a division by zero?
- (A) `(y / x) == 0`
 - (B) `((y / x) == 0) && (x != 0)`
 - (C) `((y / x) == 0) || (x != 0)`
 - (D) `(x != 0) && ((y / x) == 0)`
 - (E) `(x != 0) || ((y / x) == 0)`

Unit 3: Boolean (Part A)

40. Consider the following code segment.

```
int num = /* initial value not shown */;
boolean b1 = true;
if (num > 0)
{
    if (num >= 100)
    {
        b1 = false;
    }
}
else
{
    if (num >= -100)
    {
        b1 = false;
    }
}
```

Which of the following statements assigns the same value to `b2` as the code segment assigns to `b1` for all values of `num` ?

- (A) `boolean b2 = (num > -100) && (num < 100);`
- (B) `boolean b2 = (num > -100) || (num < 100);`
- (C) `boolean b2 = (num < -100) || (num > 100);`
- (D) `boolean b2 = (num < -100) && (num > 0 || num < 100);`
- (E) `boolean b2 = (num < -100) || (num > 0 && num < 100);`

41. Consider the following code segment.

```
int x = 3;
int y = -1;
if (x - 2 > y)
{
    x -= y;
}
if (y + 3 >= x)
{
    y += x;
}
System.out.print("x = " + x + " y = " + y);
```

What is printed as a result of the execution of the code segment?

Unit 3: Boolean (Part A)

- (A) $x = -1$ $y = -1$
- (B) $x = 2$ $y = 1$
- (C) $x = 3$ $y = 2$
- (D) $x = 4$ $y = -1$
- (E) $x = 4$ $y = 3$

Unit 3: Boolean (Part A)

42. Consider the following incomplete method.

```
public int someProcess(int n)
{
    /* body of someProcess */
}
```

The following table shows several examples of input values and the results that should be produced by calling `someProcess`.

Input Value n	Value Returned by <code>someProcess(n)</code>
3	30
6	60
7	7
8	80
9	90
11	11
12	120
14	14
16	160

Which of the following code segments could be used to replace */* body of someProcess */* so that the method will produce the results shown in the table?

I. `if ((n % 3 == 0) && (n % 4 == 0))`

```
    return n * 10;
```

```
else
```

```
    return n;
```

Unit 3: Boolean (Part A)

II. `if ((n % 3 == 0) || (n % 4 == 0))`

`return n * 10;`

`return n;`

III. `if (n % 3 == 0)`

`if (n % 4 == 0)`

`return n * 10;`

`return n;`

(A) I only

(B) II only

(C) III only

(D) I and III

(E) II and III

Unit 3: Boolean (Part A)

43. Consider the following method.

```
public String inRangeMessage(int value)
{
    if (value < 0 || value > 100)
        return "Not in range";
    else
        return "In range";
}
```

Consider the following code segments that could be used to replace the body of `inRangeMessage`.

- I. `if (value < 0)`

```
{
    if (value > 100)
        return "Not in range";
    else
        return "In range";
}
else
    return "In range";
```

- II. `if (value < 0)`

```
    return "Not in range";
else if (value > 100)
    return "Not in range";
else
```


Unit 3: Boolean (Part A)

```
return "In range";
```

III. `if (value >= 0)`

```
return "In range";
```

```
else if (value <= 100)
```

```
return "In range";
```

```
else
```

```
return "Not in range";
```

Which of the replacements will have the same behavior as the original version of `inRangeMessage` ?

- (A) I only
- (B) II only
- (C) III only
- (D) I and III
- (E) II and III

44. The following method is intended to return `true` if and only if the parameter `val` is a multiple of 4 but is not a multiple of 100 unless it is also a multiple of 400. The method does not always work correctly.

```
public boolean isLeapYear(int val)
{
    if ((val % 4) == 0)
    {
        return true;
    }
    else
    {
        return (val % 400) == 0;
    }
}
```

Which of the following method calls will return an incorrect response?

- (A) `isLeapYear(1900)`
- (B) `isLeapYear(1984)`
- (C) `isLeapYear(2000)`
- (D) `isLeapYear(2001)`
- (E) `isLeapYear(2010)`

Unit 3: Boolean (Part A)

45. Consider the following instance variables and method that appear in a class representing student information.

```
private int assignmentsCompleted;

private double testAverage;

public boolean isPassing()

{ /* implementation not shown */ }
```

A student can pass a programming course if at least one of the following conditions is met.

- The student has a test average that is greater than or equal to 90.
- The student has a test average that is greater than or equal to 75 and has at least 4 completed assignments.

Consider the following proposed implementations of the `isPassing` method.

I. `if (testAverage >= 90)`

```
return true;
```

```
if (testAverage >= 75 && assignmentsCompleted >= 4)
```

```
return true;
```

```
return false;
```

II. `boolean pass = false;`

```
if (testAverage >= 90)
```

```
pass = true;
```

```
if (testAverage >= 75 && assignmentsCompleted >= 4)
```

```
pass = true;
```

```
return pass;
```

III. `return (testAverage >= 90) ||`

Unit 3: Boolean (Part A)

```
(testAverage >= 75 && assignmentsCompleted >= 4);
```

Which of the implementations will correctly implement method `isPassing`?

- (A) I only
- (B) II only
- (C) I and III only
- (D) II and III only
- (E) I, II, and III

46. Consider the following data field and method.

```
private int[] seq;

// precondition: seq.length > 0

public int lenIncreasing()
{
    int k = 1;

    while ((k < seq.length) && (seq[k - 1] < seq[k]))
    {
        k++;
    }

    // assertion

    return k;
}
```

Which of the following assertions is true when execution reaches the line `// assertion` in `lenIncreasing`?

- (A) $(k == \text{seq.length}) \ \&\& \ (\text{seq}[k - 1] \geq \text{seq}[k])$
- (B) $(k == \text{seq.length}) \ || \ (\text{seq}[k - 1] \geq \text{seq}[k])$
- (C) $(k < \text{seq.length}) \ \&\& \ (\text{seq}[k - 1] < \text{seq}[k])$
- (D) $(k < \text{seq.length}) \ || \ (\text{seq}[k - 1] < \text{seq}[k])$
- (E) $(k == \text{seq.length}) \ || \ (\text{seq}[k - 1] == \text{seq}[k])$

Unit 3: Boolean (Part A)

47. Consider the following code segment.

```
int k = 0;
while (k < 10)
{
    System.out.print((k % 3) + " ");
    if ((k % 3) == 0)
        k = k + 2;
    else
        k++;
}
```

What is printed as a result of executing the code segment?

- (A) 0 2 1 0 2
- (B) 0 2 0 2 0 2
- (C) 0 2 1 0 2 1 0
- (D) 0 2 0 2 0 2 0
- (E) 0 1 2 1 2 1 2

Unit 3: Boolean (Part A)

48. Consider the following method.

```
public static int mystery(boolean a, boolean b, boolean c)
{
    int answer = 7;

    if (!a)
    {
        answer += 1;
    }

    if (b)
    {
        answer += 2;
    }

    if (c)
    {
        answer += 4;
    }

    return answer;
}
```

Which of the following method calls will return the value 11 ?

- (A) `mystery(true, true, true)`
- (B) `mystery(true, false, true)`
- (C) `mystery(false, true, false)`
- (D) `mystery(false, false, true)`
- (E) `mystery(false, false, false)`

Unit 3: Boolean (Part A)

49. Consider the following method.

```
public static int mystery(int[] arr)
{
    int count = 0;
    int curr = arr[arr.length - 1];

    for (int value : arr)
    {
        if (value > curr)
        {
            count = count + 1;
        }
        else
        {
            count = count - 1;
        }
        curr = value;
    }

    return count;
}
```

The following code segment appears in another method of the same class.

```
int[] arr = {4, 14, 15, 3, 14, 18, 19};
System.out.println(mystery(arr));
```

What is printed as a result of executing the code segment?

- (A) -7
- (B) -6
- (C) 3
- (D) 5
- (E) 7

Unit 3: Boolean (Part A)

50. Consider the following code segment.

```
if (false && true || false)
{
    if (false || true && false)
    {
        System.out.print("First");
    }
    else
    {
        System.out.print("Second");
    }
}
if (true || true && false)
{
    System.out.print("Third");
}
```

What is printed as a result of executing the code segment?

- (A) First
- (B) Second
- (C) Third
- (D) FirstThird
- (E) SecondThird

Unit 3: Boolean (Part A)

51. Consider the following code segment.

```
int start = 4;
int end = 5;
boolean keepGoing = true;
if (start < end && keepGoing)
{
    if (end > 0)
    {
        start += 2;
        end++;
    }
    else
    {
        end += 3;
    }
}
if (start < end)
{
    if (end == 0)
    {
        end += 2;
        start++;
    }
    else
    {
        end += 4;
    }
}
```

What is the value of `end` after the code segment is executed?

- (A) 5
- (B) 6
- (C) 9
- (D) 10
- (E) 16

Unit 3: Boolean (Part A)

52. Consider the following code segment.

```
int x = 7;
int y = 4;
boolean a = false;
boolean b = false;
if (x > y)
{
    if (x % y >= 3)
    {
        a = true;
        x -= y;
    }
    else
    {
        x += y;
    }
}
if (x < y)
{
    if (y % x >= 3)
    {
        b = true;
        x -= y;
    }
    else
    {
        x += y;
    }
}
```

What are the values of `a`, `b`, and `x` after the code segment has been executed?

- (A) `a = true`, `b = true`, `x = -1`
- (B) `a = true`, `b = false`, `x = 3`
- (C) `a = true`, `b = false`, `x = 7`
- (D) `a = false`, `b = true`, `x = 3`
- (E) `a = false`, `b = false`, `x = 11`

Unit 3: Boolean (Part A)

53. Consider the following method.

```
public void numberCheck(int maxNum)
{
    int typeA = 0;
    int typeB = 0;
    int typeC = 0;

    for (int k = 1; k <= maxNum; k++)
    {
        if (k % 2 == 0 && k % 5 == 0)
            typeA++;
        if (k % 2 == 0)
            typeB++;
        if (k % 5 == 0)
            typeC++;
    }

    System.out.println(typeA + " " + typeB + " " + typeC);
}
```

What is printed as a result of the call `numberCheck(50)` ?

- (A) 5 20 5
- (B) 5 20 10
- (C) 5 25 5
- (D) 5 25 10
- (E) 30 25 10

Unit 3: Boolean (Part A)

54. The price per box of ink pens advertised in an office supply catalog is based on the number of boxes ordered. The following table shows the pricing.

Number of Boxes	Price per Box
1 up to but not including 5	\$5.00
5 up to but not including 10	\$3.00
10 or more	\$1.50

The following incomplete method is intended to return the total cost of an order based on the value of the parameter numBoxes.

```
/** Precondition: numBoxes > 0 */
public static double getCost(int numBoxes)
{
    double totalCost = 0.0;

    /* missing code */

    return totalCost;
}
```

Which of the following code segments can be used to replace */* missing code */* so that method getCost will work as intended?

```
I. if (numBoxes >= 10)
{
    totalCost = numBoxes * 1.50;
}
if (numBoxes >= 5)
{
    totalCost = numBoxes * 3.00;
}
if (numBoxes > 0)
{
    totalCost = numBoxes * 5.00;
}
```

Unit 3: Boolean (Part A)

```
II.  if (numBoxes >= 10)
    {
        totalCost = numBoxes * 1.50;
    }
    else if (numBoxes >= 5)
    {
        totalCost = numBoxes * 3.00;
    }
    else
    {
        totalCost = numBoxes * 5.00;
    }
```

```
III. if (numBoxes > 0)
    {
        totalCost = numBoxes * 5.00;
    }
    else if (numBoxes >= 5)
    {
        totalCost = numBoxes * 3.00;
    }
    else if (numBoxes >= 10)
    {
        totalCost = numBoxes * 1.50;
    }
```

- (A) I only
- (B) II only
- (C) III only
- (D) I and II
- (E) II and III

55. Consider the following code segment.

```
int a = 10;
int b = 5 * 2;
System.out.print(a == b);
```

What is printed as a result of executing the code segment?

- (A) 5
- (B) 10
- (C) 10 == 10
- (D) true
- (E) false

Unit 3: Boolean (Part A)

56. Consider the following code segment. Assume `num` is a properly declared and initialized `int` variable.

```
if (num > 0)
{
    if (num % 2 == 0)
    {
        System.out.println("A");
    }
    else
    {
        System.out.println("B");
    }
}
```

Which of the following best describes the result of executing the code segment?

- (A) When `num` is a negative odd integer, "B" is printed; otherwise, "A" is printed.
- (B) When `num` is a negative even integer, "B" is printed; otherwise, nothing is printed.
- (C) When `num` is a positive even integer, "A" is printed; otherwise, "B" is printed.
- (D) When `num` is a positive even integer, "A" is printed; when `num` is a positive odd integer, "B" is printed; otherwise, nothing is printed.
- (E) When `num` is a positive odd integer, "A" is printed; when `num` is a positive even integer, "B" is printed; otherwise, nothing is printed.

57. Consider the following code segment.

```
double a = 1.1;

double b = 1.2;

if ((a + b) * (a - b) != (a * a) - (b * b))
{
    System.out.println("Mathematical error!");
}
```

Which of the following best describes why the phrase "Mathematical error!" would be printed?

(Remember that mathematically $(a + b) * (a - b) = a^2 - b^2$.)

Unit 3: Boolean (Part A)

- (A) Precedence rules make the if condition true.
- (B) Associativity rules make the if condition true.
- (C) Roundoff error makes the if condition true.
- (D) Overflow makes the if condition true.
- (E) A compiler bug or hardware error has occurred.

58. Consider the following method.

```
public static void printSome(int num1, int num2)
{
    for (int i = 0; i < num1; i++)
    {
        {
            System.out.print(i + " ");
        }
    }
}
```

Which of the following method calls will cause "0 10 " to be printed?

- (A) `printSome(0, 20)`
- (B) `printSome(5, 10)`
- (C) `printSome(10, 5)`
- (D) `printSome(20, 5)`
- (E) `printSome(25, 5)`

Unit 3: Boolean (Part A)

59. Consider the following method.

```
public static void whatIsIt(int a, int b)
{
    if ((a < b) && (a > 0))
    {
        System.out.println("W");
    }
    else if (a == b)
    {
        if (b > 0)
        {
            System.out.println("X");
        }
        else if (b < 0)
        {
            System.out.println("Y");
        }

        {
            System.out.println("Z");
        }
    }
}
```

Which of the following method calls will cause "W" to be printed?

- I. `whatIsIt(10, 10)`
 - II. `whatIsIt(-5, 5)`
 - III. `whatIsIt(7, 10)`
- (A) I only
(B) II only
(C) III only
(D) I and III
(E) II and III

Unit 3: Boolean (Part A)

60. Consider the following method.

```
public static void message(int a, int b, int c)
{
    if (a < 10)
    {
        if (b < 10)
        {
            System.out.print("X");
        }
        System.out.print("Y");
    }
    if (c < 10)
    {
        if (b > 10)
        {
            System.out.print("Y");
        }
        else
        {
            System.out.print("Z");
        }
    }
}
```

What is printed as a result of the call `message(5, 15, 5)` ?

- (A) XY
- (B) XYZ
- (C) Y
- (D) YY
- (E) Z

Unit 3: Boolean (Part A)

61. Consider the following method, which returns an `int` based on its parameter `x`.

```
public static int puzzle(int x)
{
    if (x > 20)
    {
        x -= 2;
    }
    else if (x % 2 == 0) // Line 7
    {
        x += 4;
    }
    return x;
}
```

Consider a modification to the method that eliminates the `else` from line 7 so that line 7 becomes

```
if (x % 2 == 0) // Modified line 7
```

For which of the following values of `x` would the return values of the original method and the modified method differ?

- (A) 0
 - (B) 5
 - (C) 14
 - (D) 22
 - (E) 25
62. Assume that `a` and `b` are variables of type `int`. The expression

```
!(a < b) && !(a > b)
```

is equivalent to which of the following?

- (A) `true`
- (B) `false`
- (C) `a == b`
- (D) `a != b`
- (E) `!(a < b) && (a > b)`

Unit 3: Boolean (Part A)

63. Consider the following method that is intended to modify its parameter `nameList` by replacing all occurrences of `name` with `newValue`.

```
public void replace(ArrayList<String> nameList,  
                   String name, String newValue)  
{  
    for (int j = 0; j < nameList.size(); j++)  
    {  
        if ( /* expression */ )  
        {  
            nameList.set(j, newValue);  
        }  
    }  
}
```

Which of the following can be used to replace `/* expression */` so that `replace` will work as intended?

- (A) `nameList.get(j).equals(name)`
- (B) `nameList.get(j) == name`
- (C) `nameList.remove(j)`
- (D) `nameList[j] == name`
- (E) `nameList[j].equals(name)`

Unit 3: Boolean (Part A)

64. A school that does not have air conditioning has published a policy to close school when the outside temperature reaches or exceeds 95°F. The following code segment is intended to print a message indicating whether or not the school is open, based on the temperature. Assume that the variable `degrees` has been properly declared and initialized with the outside temperature.

```
if (degrees > 95)
{
    System.out.println("School will be closed due to extreme heat");
}
else
{
    System.out.println("School is open");
}
```

Which of the following initializations for `degrees`, if any, will demonstrate that the code segment may not work as intended?

- (A) `degrees = 90;`
 - (B) `degrees = 94;`
 - (C) `degrees = 95;`
 - (D) `degrees = 96;`
 - (E) The code will work as intended for all values of `degrees`.
65. Consider the following code segment.

```
int x = 7;
if (x < 7)
{
    x = 2 * x;
}
if (x % 3 == 1)
{
    x = x + 2;
}
System.out.print(3 * x);
```

What is printed as a result of executing the code segment?

- (A) 7
- (B) 9
- (C) 14
- (D) 21
- (E) 27

Unit 3: Boolean (Part A)

66. Consider the following code segment.

```
double regularPrice = 100;
boolean onClearance = true;
boolean hasCoupon = false;
double finalPrice = regularPrice;
if (onClearance)
{
    finalPrice -= finalPrice * 0.25;
}
if (hasCoupon)
{
    finalPrice -= 5.0;
}
System.out.println(finalPrice);
```

What is printed as a result of executing the code segment?

- (A) 20.0
- (B) 25.0
- (C) 70.0
- (D) 75.0
- (E) 95.0

67. Consider the following method.

```
public int someCode(int a, int b, int c)
{
    if ((a < b) && (b < c))
        return a;
    if ((a >= b) && (b >= c))
        return b;
    if ((a == b) || (a == c) || (b == c))
        return c;
}
```

Which of the following best describes why this method does not compile?

Unit 3: Boolean (Part A)

- (A) The reserved word `return` cannot be used in the body of an `if` statement.
- (B) It is possible to reach the end of the method without returning a value.
- (C) The `if` statements must have `else` parts when they contain `return` statements.
- (D) Methods cannot have multiple `return` statements.
- (E) The third `if` statement is not reachable.

68. Consider the following statement.

```
boolean x = (5 < 8) == (5 == 8);
```

What is the value of `x` after the statement has been executed?

- (A) 3
 - (B) 5
 - (C) 8
 - (D) `true`
 - (E) `false`
69. Consider the following Boolean expression in which the `int` variables `x` and `y` have been properly declared and initialized.

```
(x <= 10) == (y > 25)
```

Which of the following values for `x` and `y` will result in the expression evaluating to `true` ?

- (A) `x = 8` and `y = 25`
- (B) `x = 10` and `y = 10`
- (C) `x = 10` and `y = 30`
- (D) `x = 15` and `y = 30`
- (E) `x = 25` and `y = 30`

Unit 3: Boolean (Part A)

70. Vehicles are classified based on their total interior volume. The `classify` method is intended to return a vehicle classification `String` value based on total interior volume, in cubic feet, as shown in the table below.

Vehicle size class	Total interior volume
Minicompact	Less than 85 cubic feet
Subcompact	85 to 99 cubic feet
Compact	100 to 109 cubic feet
Mid-Size	110 to 119 cubic feet
Large	120 cubic feet or more

The `classify` method, which does not work as intended, is shown below.

```
public static String classify(int volume)
{
    String carClass = "";
    if (volume >= 120)
    {
        carClass = "Large";
    }
    else if (volume < 120)
    {
        carClass = "Mid-Size";
    }
    else if (volume < 110)
    {
        carClass = "Compact";
    }
    else if (volume < 100)
    {
        carClass = "Subcompact";
    }
    else
    {
        carClass = "Minicompact";
    }
    return carClass;
}
```

The `classify` method works as intended for some but not all values of the parameter `volume`. For which of the following values of `volume` would the correct value be returned when the `classify` method is executed?

Unit 3: Boolean (Part A)

- (A) 80
- (B) 90
- (C) 105
- (D) 109
- (E) 115

71. Consider the following code segment.

```
int num1 = 0;

int num2 = 3;

while ((num2 != 0) && ((num1 / num2) >= 0))
{
    num1 = num1 + 2;
    num2 = num2 - 1;
}
```

What are the values of num1 and num2 after the while loop completes its execution?

- (A) num1 = 0, num2 = 3
- (B) num1 = 8, num2 = -1
- (C) num1 = 4, num2 = 1
- (D) num1 = 6, num2 = 0
- (E) The loop will never complete its execution because a division by zero will generate an ArithmeticException.

Unit 3: Boolean (Part A)

72. The following categories are used by some researchers to categorize zip codes as urban, suburban, or rural based on population density.
- An urban zip code is a zip code with more than 3,000 people per square mile.
 - A suburban zip code is a zip code with between 1,000 and 3,000 people, inclusive, per square mile.
 - A rural zip code is a zip code with fewer than 1,000 people per square mile.

Consider the following method, which is intended to categorize a zip code as urban, suburban, or rural based on the population density of the area included in the zip code.

```
public static String getCategory(int density)
{
    /* missing code */
}
```

Which of the following code segments can replace `/* missing code */` so the `getCategory` method works as intended?

- I.

```
String cat;
if (density > 3000)
{
    cat = "urban";
}
else if (density > 999)
{
    cat = "suburban";
}
else
{
    cat = "rural";
}
return cat;
```
- II.

```
String cat;
if (density > 3000)
{
    cat = "urban";
}
if (density > 999)
{
    cat = "suburban";
}
cat = "rural";
return cat;
```
- III.

```
if (density > 3000)
{
    return "urban";
}
if (density > 999)
```


Unit 3: Boolean (Part A)

```
{  
    return "suburban";  
}  
return "rural";
```

- (A) I only
- (B) III only
- (C) I and II only
- (D) I and III only
- (E) I, II, and III