mcq1_4

1. In the code segment below, assume that the `int` variable `n` has been properly declared and initialized. The code segment is intended to print a value that is `1` more than twice the value of `n`.

```
/* missing code */
System.out.print(result);
```

Which of the following can be used to replace `/*` *missing code* `*/` so that the code segment works as intended?

```
  I. int result = 2 * n;
     result = result + 1;
 II. int result = n + 1;
     result = result * 2;
III. int result = (n + 1) * 2;
```

(A) I only ✓

(B) II only

(C) III only

(D) I and III

(E) II and III

2. Consider the following code segment.

```
int a = 5;
int b = 8;
int c = 3;
System.out.println(a + b / c * 2);
```

What is printed as a result of executing this code?

(A) `2`

(B) `6`

(C) `8`

(D) `9` ✓

(E) `14`

3. In the code segment below, assume that the `int` variables `a` and `b` have been properly declared and initialized.

```
int c = a;
int d = b;
c += 3;
d--;
double num = c;
num /= d;
```

Which of the following best describes the behavior of the code segment?

(A) The code segment stores the value of `(a + 3) / b` in the variable `num`.

(B) The code segment stores the value of `(a + 3) / (b - 1)` in the variable `num`. ✓

(C) The code segment stores the value of `(a + 3) / (b - 2)` in the variable `num`.

(D) The code segment stores the value of `(a + 3) / (1 - b)` in the variable `num`.

(E) The code segment causes a runtime error in the last line of code because `num` is type `double` and `d` is type `int`.

4. Consider the following code segment, which is intended to find the average of two positive integers, `x` and `y`.

```
int x;
int y;
int sum = x + y;
double average = (double) (sum / 2);
```

Which of the following best describes the error, if any, in the code segment?

(A) There is no error, and the code works as intended.

(B) In the expression `(double) (sum / 2)`, the cast to `double` is applied too late, so the average will be less than the expected result for even values of `sum`.

(C) In the expression `(double) (sum / 2)`, the cast to `double` is applied too late, so the average will be greater than the expected result for even values of `sum`.

(D) In the expression `(double) (sum / 2)`, the cast to `double` is applied too late, so the average will be less than the expected result for odd values of `sum`. ✓

(E) In the expression `(double) (sum / 2)`, the cast to `double` is applied too late, so the average will be greater than the expected result for odd values of `sum`.

mcq1_4

5. Consider the following static method.

```
public static int calculate(int x)

{

  x = x + x;

  x = x + x;

  x = x + x;



  return x;

}
```

Which of the following can be used to replace the body of calculate so that the modified version of calculate will return the same result as the original version for all x ?

(A) return 3 + x;

(B) return 3 * x;

(C) return 4 * x;

(D) return 6 * x;

[✓ (E) return 8 * x;]

mcq1_4

6. Consider the following static method.

```
public static int calculate(int x)

{

  x = x + x;

  x = x + x;

  x = x + x;

  return x;

}
```

Which of the following can be used to replace the body of calculate so that the modified version of calculate will return the same result as the original version for all x ?

(A) return 2 * x;

(B) return 4 * x;

(C) return 8 * x; ✓

(D) return 3 * calculate(x);

(E) return x + calculate(x - 1);

7. Consider the following code segment.

```
double num = 9 / 4;
System.out.print(num);
System.out.print(" ");
System.out.print((int) num);
```

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

(A) `2 2`

(B) `2.0 2` ✓

(C) `2.0 2.0`

(D) `2.25 2`

(E) `2.25 2.0`

8. Which of the following expressions evaluate to `3.5` ?

```
  I. (double) 2 / 4 + 3
 II. (double) (2 / 4) + 3
III. (double) (2 / 4 + 3)
```

[✓ (A) I only]

(B) III only

(C) I and II only

(D) II and III only

(E) I, II, and III

9. Consider the following code segment.

```
double x = (int) (5.5 - 2.5);
double y = (int) 5.5 - 2.5;
System.out.println(x - y);
```

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

(A) `-1.0`

(B) `-0.5`

(C) `0.0`

[✓ (D) `0.5`]

(E) `1.0`

10. Consider the following code segment.

```
int w = 1;
int x = w / 2;
double y = 3;
int z = (int) (x + y);
```

Which of the following best describes the results of compiling the code segment?

[✓ (A) The code segment compiles without error.]

(B) The code segment does not compile, because the `int` variable `x` cannot be assigned the result of the operation `w / 2`.

(C) The code segment does not compile, because the integer value `3` cannot be assigned to the `double` variable `y`.

(D) The code segment does not compile, because the operands of the addition operator cannot be of different types `int` and `double`.

(E) The code segment does not compile because the result of the addition operation is of type `double` and cannot be cast to an `int`.

mcq1_4

11. Consider the following code segment.

```
String oldStr = "ABCDEF";
String newStr = oldStr.substring(1, 3) + oldStr.substring(4);
System.out.println(newStr);
```

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

(A) `ABCD`

(B) `BCDE`

(C) `BCEF` ✓

(D) `BCDEF`

(E) `ABCDEF`

mcq1_4

12. Consider the following class.

```
public class SomeMethods

{
public void one(int first)

{ / * implementation not shown * / }

public void one(int first, int second)

{ / * implementation not shown * / }

public void one(int first, String second)

{ / * implementation not shown * / }

}
```

Which of the following methods can be added to the SomeMethods class without causing a compile-time error?

```
  I.  public void one(int value)
      { / * implementation not shown * / }

 II.  public void one (String first, int second)

      { / * implementation not shown * / }

III.  public void one (int first, int second, int third)

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

(A) I only

(B) I and II only

(C) I and III only

[✓ (D) II and III only]

(E) I, II, and III

13. Consider the following methods, which appear in the same class.

```
public void slope(int x1, int y1, int x2, int y2)
{
     int xChange = x2 - x1;
     int yChange = y2 - y1;
     printFraction(yChange, xChange);
}

public void printFraction(int numerator, int denominator)
{
     System.out.print(numerator + "/" + denominator);
}
```

Assume that the method call `slope(1, 2, 5, 10)` appears in a method in the same class. What is printed as a result of the method call?

(A) `8/4` ✓

(B) `5/1`

(C) `4/8`

(D) `2/1`

(E) `1/5`

14. Consider the following method, which is intended to calculate and return the expression $\sqrt{\frac{(x+y)^2}{|a-b|}}$.

```
public double calculate(double x, double y, double a, double b)
{
     return /* missing code */;
}
```

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

(A) `Math.sqrt(x ^ 2, y ^ 2, a - b)`

(B) `Math.sqrt((x + y) ^ 2) / Math.abs(a, b)`

(C) `Math.sqrt((x + y) ^ 2 / Math.abs(a - b))`

(D) `Math.sqrt(Math.pow(x + y, 2) / Math.abs(a, b))`

(E) `Math.sqrt(Math.pow(x + y, 2) / Math.abs(a - b))` ✓

mcq1_4

15. Consider the following method.

```
public double myMethod(int a, boolean b)
{  /* implementation not shown */  }
```

Which of the following lines of code, if located in a method in the same class as `myMethod`, will compile without error?

(A) `int result = myMethod(2, false);`

(B) `int result = myMethod(2.5, true);`

(C) `double result = myMethod(0, false);` ✓

(D) `double result = myMethod(true, 10);`

(E) `double result = myMethod(2.5, true);`

16. Consider the following code segment.

```
String temp = "comp";
System.out.print(temp.substring(0) + " " +
            temp.substring(1) + " " +
            temp.substring(2) + " " +
            temp.substring(3));
```

What is printed when the code segment is executed?

(A) `comp`

(B) `c o m p`

(C) `comp com co c`

(D) `comp omp mp p` ✓

(E) `comp comp comp comp`

17. The following statement assigns an integer value to `x`.

```
int x = (int)(Math.random() * 5) + 10;
```

Consider the statement that would result if the positions of `5` and `10` were swapped in the preceding statement and the resulting integer were assigned to `y`.

```
int y = (int)(Math.random() * 10) + 5;
```

Which of the following are true statements about how the possible values assigned to `y` differ from the possible values assigned to `x` ?

I. There are more possible values of `x` than there are possible values of `y`.
II. There are more possible values of `y` than there are possible values of `x`.
III. The value assigned to `x` can sometimes be the same as the value assigned to `y`.

mcq1_4

(A) I only

(B) II only

(C) III only

(D) I and III

[✓ (E) II and III]

18. Consider the following code segment.

```
String str = "CompSci";
System.out.println(str.substring(0, 3));
int num = str.length();
```

What is the value of `num` when the code segment is executed?

(A) `3`

(B) `4`

(C) `5`

(D) `6`

[✓ (E) `7`]

19. Consider the following code segment.

```
String str = "0";
str += str + 0 + 8;
System.out.println(str);
```

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

(A) `8`

(B) `08`

(C) `008`

[✓ (D) `0008`]

(E) Nothing is printed, because numerical values cannot be added to a `String` object.

20. Consider the following code segment.

```
int one = 1;
int two = 2;
String zee = "Z";
System.out.println(one + two + zee);
```

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

(A) 12Z

(B) 3Z ✓

(C) 12zee

(D) 3zee

(E) onetwozee

21. 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 ✓

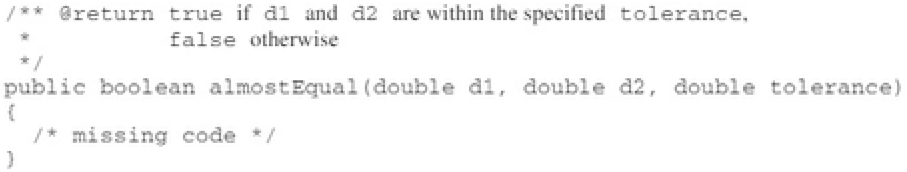
(C) C

(D) D

(E) Nothing is printed.

mcq1_4

22. 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?

(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; ✓

23. 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);

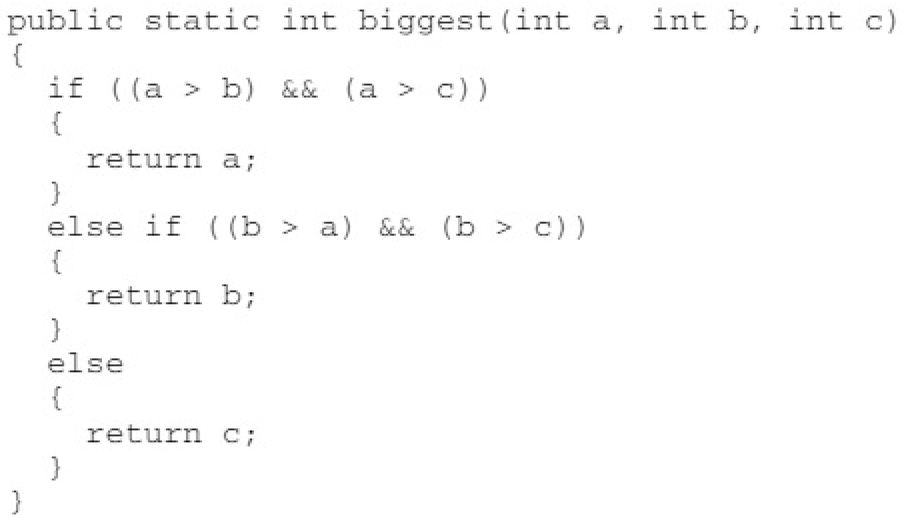
(B) return (x <= lower) || (x >= upper);

(C) return lower <= x <= upper;

(D) return (x >= lower) && (x <= upper); ✓

(E) return (x >= lower) || (x <= upper);

24. 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.

25. 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

mcq1_4

26. 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

27. 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` ?

(A) `a && b && c`

(B) `a || b || c`

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

(D) `!a && !b && c`

[✓ (E) `!a || !b || c`]

28. 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)

mcq1_4

29. 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);
```

mcq1_4

```
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

mcq1_4

30. 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;`

mcq1_4

31. Consider the following two code segments. Assume that the `int` variables `m` and `n` have been properly declared and initialized and are both greater than `0`.

```
 I. for (int i = 0; i < m * n; i++)
    {
       System.out.print("A");
    }
II. for (int j = 1; j <= m; j++)
    {
       for (int k = 1; k < n; k++)
       {
          System.out.print("B");
       }
    }
```

Assume that the initial values of `m` and `n` are the same in code segment I as they are in code segment II. Which of the following correctly compares the number of times that `"A"` and `"B"` are printed when each code segment is executed?

(A) `"A"` is printed `m` fewer times than `"B"`.

(B) `"A"` is printed `n` fewer times than `"B"`.

[✓ (C) `"A"` is printed `m` more times than `"B"`.]

(D) `"A"` is printed `n` more times than `"B"`.

(E) `"A"` and `"B"` are printed the same number of times.

32. Consider the following code segment.

```
int j = 1;
while (j < 5)
{
     int k = 1;
     while (k < 5)
     {
          System.out.println(k);
          k++;
     }
     j++;
}
```

Which of the following best explains the effect, if any, of changing the first line of code to `int j = 0;` ?

(A) There will be one more value printed because the outer loop will iterate one additional time.

[✓ (B) There will be four more values printed because the outer loop will iterate one additional time.]

(C) There will be one less value printed because the outer loop will iterate one fewer time.

(D) There will be four fewer values printed because the outer loop will iterate one fewer time.

(E) There will be no change to the output of the code segment.

mcq1_4

33. Consider the following method definition. The method `printAllCharacters` is intended to print out every character in `str`, starting with the character at index `0`.

```
public static void printAllCharacters(String str)
{
     for (int x = 0; x < str.length(); x++)    // Line 3
     {
         System.out.print(str.substring(x, x + 1));
     }
}
```

The following statement is found in the same class as the `printAllCharacters` method.

```
printAllCharacters("ABCDEFG");
```

Which choice best describes the difference, if any, in the behavior of this statement that will result from changing `x < str.length()` to `x <= str.length()` in line 3 of the method?

(A) The method call will print fewer characters than it did before the change because the loop will iterate fewer times.

(B) The method call will print more characters than it did before the change because the loop will iterate more times.

(C) The method call, which worked correctly before the change, will now cause a run-time error because it attempts to access a character at index `7` in a string whose last element is at index `6`. ✓

(D) The method call, which worked correctly before the change, will now cause a run-time error because it attempts to access a character at index `8` in a string whose last element is at index `7`.

(E) The behavior of the code segment will remain unchanged.

34. Consider the following method.

```
/** Precondition: bound >= 0 */
public int sum(int bound)
{
     int answer = 0;
     for (int i = 0; i < bound; i++)
     {
         answer += bound;
     }
     return answer;
}
```

Assume that `sum` is called with a parameter that satisfies the precondition and that it executes without error. How many times is the test expression `i < bound` in the `for` loop header evaluated?

mcq1_4

(A) `0`

(B) `bound - 1`

(C) `bound`

[✓ (D) `bound + 1`]

(E) An unknown number of times

35. Which of the following code segments produces the output `"987654321"` ?

(A)

```
int num = 10;
while (num > 0)
{
     System.out.print(num);
     num--;
}
```

(B)

```
int num = 10;
while (num >= 0)
{
     System.out.print(num);
     num--;
}
```

[✓ (C)

```
int num = 10;
while (num > 1)
{
     num--;
     System.out.print(num);
}
```
]

(D)

```
int num = 10;
while (num >= 1)
{
     num--;
     System.out.print(num);
}
```

(E)

```
int num = 0;
while (num <= 9)
{
     System.out.print(10 - num);
     num++;
}
```

mcq1_4

36. Consider the following method.

```
/** Precondition: Strings one and two have the same length. */
public static String combine(String one, String two)
{
    String res = "";
    for (int k = 0; k < one.length(); k++)
    {
        if (one.substring(k, k + 1).equals(two.substring(k, k + 1)))
        {
           res += one.substring(k, k + 1);
        }
        else
        {
           res += "0";
        }
    }
    return res;
}
```

What is returned as a result of the call `combine("10110", "01100")` ?

(A) `"00000"`

[✓ (B) `"00100"`]

(C) `"00101"`

(D) `"10110"`

(E) `"11011"`

mcq1_4

37. Consider the following methods.

```
/** Precondition: a > 0 and b > 0 */
public static int methodOne(int a, int b)
{
     int loopCount = 0;
     for (int i = 0; i < a / b; i++)
     {
         loopCount++;
     }
     return loopCount;
}
/** Precondition: a > 0 and b > 0 */
public static int methodTwo(int a, int b)
{
     int loopCount = 0;
     int i = 0;
     while (i < a)
     {
         loopCount++;
         i += b;
     }
     return loopCount;
}
```

Which of the following best describes the conditions under which `methodOne` and `methodTwo` return the same value?

(A) When `a` and `b` are both even

(B) When `a` and `b` are both odd

(C) When `a` is even and `b` is odd

[✓ (D) When `a % b` is equal to zero]

(E) When `a % b` is equal to one

mcq1_4

38. Consider the following code segments. Code segment 2 is a revision of code segment 1 in which the loop increment has been changed.

Code Segment 1

```
int sum = 0;
for (int k = 1; k <= 30; k++)
{
     sum += k;
}
System.out.println("The sum is: " + sum);
```

Code Segment 2

```
int sum = 0;
for (int k = 1; k <= 30; k = k + 2)
{
     sum += k;
}
System.out.println("The sum is: " + sum);
```

Code segment 1 prints the sum of the integers from 1 through 30, inclusive. Which of the following best explains how the output changes from code segment 1 to code segment 2 ?

(A) Code segment 1 and code segment 2 will produce the same output.

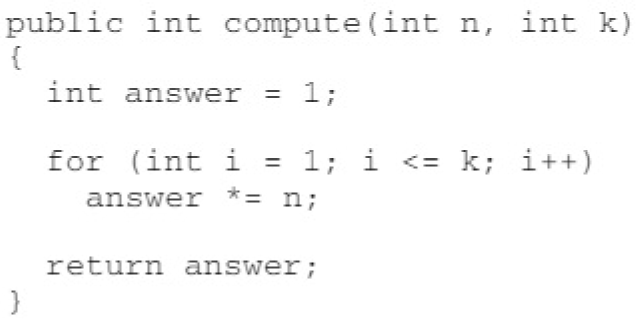
(B) Code segment 2 will print the sum of only the even integers from 1 through 30, inclusive because it starts `sum` at zero, increments `k` by twos, and terminates when `k` exceeds `30`.

(C) Code segment 2 will print the sum of only the odd integers from 1 through 30, inclusive because it starts `k` at one, increments `k` by twos, and terminates when `k` exceeds `30`. ✓

(D) Code segment 2 will print the sum of only the even integers from 1 through 60, inclusive because it starts `sum` at zero, increments `k` by twos, and iterates 30 times.

(E) Code segment 2 will print the sum of only the odd integers from 1 through 60, inclusive because it starts `k` at one, increments `k` by twos, and iterates 30 times.

39. Consider the following method.



```
public int compute(int n, int k)
{
  int answer = 1;

  for (int i = 1; i <= k; i++)
    answer *= n;

  return answer;
}
```

Which of the following represents the value returned as a result of the call compute (n, k) ?

mcq1_4

(A) n*k

(B) n!

[✓ (C) n^k]

(D) 2^k

(E) k^n

40. Consider the following code segment. The code is intended to read nonnegative numbers and compute their product until a negative number is read; however, it does not work as intended. (Assume that the readInt method correctly reads the next number from the input stream.)

```
int k = 0;
int prod = 1;


while (k >= 0)
{
  System.out.print("enter a number: ");
  k = readInt(); // readInt reads the next number from input
  prod = prod * k;
}


System.out.println("product: " + prod);
```

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

(A) The variable prod is incorrectly initialized.

(B) The while condition always evaluates to false.

(C) The while condition always evaluates to true.

[✓ (D) The negative number entered to signal no more input is included in the product.]

(E) If the user enters a zero, the computation of the product will be terminated prematurely.