chap 4: Iteration

1. 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.

2. 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.

chap 4: Iteration

3. 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.

4. 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?

chap 4: Iteration

(A) `0`

(B) `bound - 1`

(C) `bound`

(D) `bound + 1`

(E) An unknown number of times

5. 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++;
}
```

chap 4: Iteration

6. 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

chap 4: Iteration

7. 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.

8. 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) ?

chap 4: Iteration

(A) n*k
(B) n!
(C) n^k
(D) 2^k
(E) k^n

9. 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.

chap 4: Iteration

10. Consider the following code segment.

```
int num = 1;
int count = 0;
while (num <= 10)
{
     if (num % 2 == 0 && num % 3 == 0)
     {
         count++;
     }
     num++;
}
```

What value is stored in the variable `count` as a result of executing the code segment?

(A) 1

(B) 3

(C) 5

(D) 7

(E) 8

11. Consider the following code segment.

```
String str = "a black cat sat on a table";
int counter = 0;
for (int i = 0; i < str.length() - 1; i++)
{
     if (str.substring(i, i + 1).equals("a") &&
         !str.substring(i + 1, i + 2).equals("b"))
     {
         counter++;
     }
}
System.out.println(counter);
```

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

(A) 1

(B) 2

(C) 3

(D) 5

(E) 6

chap 4: Iteration

12. Consider the following code segments.

I.
```
int k = 1;
while (k < 20)
{
  if (k % 3 == 1)
    System.out.print( k + " ");

  k = k + 3;
}
```

II.
```
for (int k = 1; k < 20; k++)
{
  if (k % 3 == 1)
    System.out.print( k + " ");
}
```

III.
```
for (int k = 1; k < 20; k = k + 3)
{
  System.out.print( k + " ");
}
```

Which of the code segments above will produce the following output?

1 4 7 10 13 16 19

chap 4: Iteration

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

13. Consider the following two code segments. Code segment II is a revision of code segment I in which the loop header has been changed.

I.

```
for (int k = 1; k <= 5; k++)
{
     System.out.print(k);
}
```

II.

```
for (int k = 5; k >= 1; k--)
{
     System.out.print(k);
}
```

Which of the following best explains how the output changes from code segment I to code segment II?

(A) Both code segments produce the same output, because they both iterate four times.
(B) Both code segments produce the same output, because they both iterate five times.
(C) Code segment I prints more values than code segment II does, because it iterates for one additional value of `k`.
(D) Code segment II prints more values than code segment I, because it iterates for one additional value of `k`.
(E) The code segments print the same values but in a different order, because code segment I iterates from `1` to `5` and code segment II iterates from `5` to `1`.

14. Consider the following code segment.

```
int count = 5;
while (count < 100)
{
     count = count * 2;
}
count = count + 1;
```

What will be the value of `count` as a result of executing the code segment?

(A) `100`
(B) `101`
(C) `160`
(D) `161`
(E) `321`

chap 4: Iteration

15. Consider the following method.

```
//* Precondition: num > 0 */
public static int doWhat(int num)
{
  int var = 0;

  for (int loop = 1; loop <= num; loop = loop + 2)
  {
    var += loop;
  }

  return var;
}
```

Which of the following best describes the value returned from a call to doWhat ?

(A) num

(B) The sum of all integers between 1 and num, inclusive

(C) The sum of all even integers between 1 and num, inclusive

(D) The sum of all odd integers between 1 and num, inclusive

(E) No value is returned because of an infinite loop.

16. Consider the following code segment.

```
for (int x = 0; x <= 4; x++)      // Line 1
{
     for (int y = 0; y < 4; y++)   // Line 3
     {
         System.out.print("a");
     }
     System.out.println();
}
```

Which of the following best explains the effect of simultaneously changing `x <= 4` to `x < 4` in line 1 and `y < 4` to `y <= 4` in line 3 ?

(A) `"a"` will be printed fewer times because while each output line will have the same length as before, the number of lines printed will decrease by 1.

(B) `"a"` will be printed more times because while the number of output lines will be the same as before, the length of each output line will increase by 1.

(C) `"a"` will be printed the same number of times because while the number of output lines will decrease by 1, the length of each line will increase by 1.

(D) `"a"` will be printed more times because both the number of output lines and the length of each line will increase by 1.

(E) The output of the code segment will not change in any way.

chap 4: Iteration

17. Consider the following code segment.

```
for (int k = 1; k <= 7; k += 2)
{
     System.out.print(k);
}
```

Which of the following code segments will produce the same output as the code segment above?

(A)
```
for (int k = 0; k < 7; k += 2)
{
     System.out.print(k);
}
```

(B)
```
for (int k = 0; k <= 7; k += 2)
{
     System.out.print(k);
}
```

(C)
```
for (int k = 0; k <= 8; k += 2)
{
     System.out.print(k + 1);
}
```

(D)
```
for (int k = 1; k < 7; k += 2)
{
     System.out.print(k + 1);
}
```

(E)
```
for (int k = 1; k <= 8; k += 2)
{
     System.out.print(k);
}
```

18. Consider the following code segment.

```
int value = 15;
while (value < 28)
{
  System.out.println(value);
  value++;
}
```

What are the first and last numbers output by the code segment?

chap 4: Iteration

(A)

First	Last
15	27

(B)

First	Last
15	28

(C)

First	Last
16	27

(D)

First	Last
16	28

(E)

First	Last
16	29

chap 4: Iteration

19. Consider the following code segment.

```
int count = 0;
for (int k = 0; k < 10; k++)
{
     count++;
}
System.out.println(count);
```

Which of the following code segments will produce the same output as the code segment above?

(A)
```
int count = 0;
for (int k = 1; k < 10; k++)
{
     count++;
}
System.out.println(count);
```

(B)
```
int count = 1;
for (int k = 1; k <= 10; k++)
{
     count++;
}
System.out.println(count);
```

(C)
```
int count = 1;
for (int k = 0; k <= 9; k++)
{
     count++;
}
System.out.println(count);
```

(D)
```
int count = 0;
for (int k = 9; k >= 0; k--)
{
     count++;
}
System.out.println(count);
```

(E)
```
int count = 0;
for (int k = 10; k >= 0; k--)
{
     count++;
}
System.out.println(count);
```

chap 4: Iteration

20. Consider the following code segment.

```
/* missing loop header */
{
     for (int k = 0; k < 4; k++)
     {
         System.out.print(k);
     }
     System.out.println();
}
```

The code segment is intended to produce the following output.

```
0123
0123
0123
```

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

```
I.  for (int j = 0; j < 3; j++)
II. for (int j = 1; j < 3; j++)
III. for (int j = 1; j <= 3; j++)
```

(A) I only

(B) II only

(C) III only

(D) I and II

(E) I and III

21. Consider the following code segment.

```
for (int num = 0; num < 10; num += 2)
{
     for (int val = 0; val < 5; val++)
     {
         System.out.println("hop");
     }
}
```

How many times will `System.out.println("hop")` be executed?

(A) 0

(B) 5

(C) 10

(D) 25

(E) 50

chap 4: Iteration

22. Consider the following code segment.

```
int counter = 0;
for (int x = 10; x > 0; x--)
{
     for (int y = x; y <= x; y++)
     {
         counter++;  // line 6
     }
}
```

How many times will the statement in line 6 be executed as a result of running the code segment?

(A) 0
(B) 1
(C) 10
(D) 11
(E) 20

23. Consider the following code segment.

```
int outerMax = 10;
int innerMax = 5;
for (int outer = 0; outer < outerMax; outer++)
{
     for (int inner = 0; inner <= innerMax; inner++)
     {
         System.out.println(outer + inner);
     }
}
```

How many values will be printed when the code segment is executed?

(A) 45
(B) 50
(C) 55
(D) 60
(E) 66

chap 4: Iteration

24. Consider the following code segment.

```
int count = 0;

for (int x = 0; x < 4; x++)
{
  for (int y = x; y < 4; y++)
  {
    count++;
  }
}
System.out.println(count);
```

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

(A) 4
(B) 8
(C) 10
(D) 16
(E) 20

25. The following method is intended to print the number of digits in the parameter `num`.

```
public int numDigits(int num)
{
     int count = 0;
     while (/* missing condition */)
     {
         count++;
         num = num / 10;
     }
     return count;
}
```

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

(A) `count != 0`
(B) `count > 0`
(C) `num >= 0`
(D) `num != 0`
(E) `num == 0`

chap 4: Iteration

26. Consider the following method.

```
public int mystery(int num)

{

int x = num;

while (x > 0)

{

if (x / 10 % 2 == 0)

return x;

x = x / 10;

}

return x;

}
```

What value is returned as a result of the call mystery(1034) ?

(A) 4
(B) 10
(C) 34
(D) 103
(E) 1034

chap 4: Iteration

27. Consider the following code segment.

```
for (int k = 0; k < 20; k = k + 2)
{
  if (k % 3 == 1)
  {
    System.out.print(k + "  ");
  }
}
```

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

(A) 4 16

(B) 4 10 16

(C) 0 6 12 18

(D) 1 4 7 10 13 16 19

(E) 0 2 4 6 8 10 12 14 16 18

28. Consider the following code segment, which is intended to print the sum of all the odd integers from 0 up to and including 101.

```
int r = 0;
int sum = 0;
/* missing loop header */
{
     if (r % 2 == 1)
     {
          sum += r;
     }
     r++;
}
System.out.println(sum);
```

Which of the following could replace `/* missing loop header */` to ensure that the code segment will work as intended?

(A) `while (r <= 100)`

(B) `while (sum <= 100)`

(C) `while (r < 101)`

(D) `while (r <= 101)`

(E) `while (sum <= 101)`

chap 4: Iteration

29. Consider the following code segment.

```
int x = 1;
while ( /* condition */ )
{
  if (x % 2 == 0)
  {
    System.out.print(x + " ");
  }
  x = x + 2;
}
```

The following conditions have been proposed to replace /* *condition* */ in the code segment.

I. x < 0

II. x <= 1

III. x < 10

For which of the conditions will nothing be printed?

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

chap 4: Iteration

30. Consider the following code segment.

```
int a = 24;
int b = 30;
while (b != 0)
{
 int r = a % b;
 a = b;
 b = r;
}

System.out.println(a);
```

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

(A) 0
(B) 6
(C) 12
(D) 24
(E) 30

31. Consider the following code segment. Assume that `num3 > num2 > 0`.

```
int num1 = 0;
int num2 = /* initial value not shown */;
int num3 = /* initial value not shown */;
while (num2 < num3)
{
     num1 += num2;
     num2++;
}
```

Which of the following best describes the contents of `num1` as a result of executing the code segment?

chap 4: Iteration

(A) The product of `num2` and `num3`

(B) The product of `num2` and `num3 - 1`

(C) The sum of `num2` and `num3`

(D) The sum of all integers from `num2` to `num3`, inclusive

(E) The sum of all integers from `num2` to `num3 - 1`, inclusive

32. Consider the following code segment.

```
for (int outer = 0; outer < 3; outer++)
{
     for (/* missing loop header */)
     {
         System.out.print(outer + "" + inner + "_");
     }
}
```

Which of the following can be used as a replacement for `/* missing loop header */` so that the code segment produces the output `00_01_02_11_12_22_` ?

(A) `int inner = 0; inner < 3; inner++`

(B) `int inner = 1; inner < 3; inner++`

(C) `int inner = outer - 1; inner < 3; inner++`

(D) `int inner = outer; inner < 3; inner++`

(E) `int inner = outer + 1; inner < 3; inner++`

33. Consider the following code segment.

```
int count = 0;
for (int x = 1; x <= 3; x++)
{
     /* missing loop header */
     {
         count++;
     }
}
System.out.println(count);
```

Which of the following should be used to replace `/* missing loop header */` so that the code segment will print `6` as the value of `count` ?

(A) `for (int y = 0; y <= 2; y++)`

(B) `for (int y = 0; y < 3; y++)`

(C) `for (int y = 2; y >= 0; y--)`

(D) `for (int y = 3; y > 0; y--)`

(E) `for (int y = 0; y < x; y++)`

chap 4: Iteration

34. Consider the following code segment.

```
int x = 1;
while ( /* missing code */ )
{
  System.out.print(x + " ");
  x = x + 2;
}
```

Consider the following possible replacements for /* missing code */.

I. x < 6

II. x != 6

III. x < 7

Which of the proposed replacements for /* missing code */ will cause the code segment to print only the values 1 3 5?

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

35. Consider the following code segment.

```
int k = 0;
/* missing loop header */
{
     k++;
     System.out.print(k + " ");
}
```

Which of the following can be used as a replacement for `/* missing loop header */` so that the code segment prints out the string `"1 2 3 4 "`?

(A) `while (k < 3)`
(B) `while (k < 4)`
(C) `while (k < 5)`
(D) `while (k <= 4)`
(E) `while (k <= 5)`

chap 4: Iteration

36. Consider the following code segment.

```
int val = 48;
int div = 6;
while ((val % 2 == 0) && div > 0)
{
     if (val % div == 0)
     {
         System.out.print(val + " ");
     }
     val /= 2;
     div--;
}
```

What is printed when the code segment is executed?

(A) `48 12 6`

(B) `48 12 6 3`

(C) `48 12 6 3 1`

(D) `48 24 12 6`

(E) `48 24 12 6 3`

Directions: Select the choice that best fits each statement. The following question(s) refer to the following method

```
public static int mystery(int n)
{
  int x = 1;
  int y = 1;

  // Point A

  while (n > 2)
  {
    x = x + y;

    // Point B

    y = x - y;
    n--;
  }

  // Point C

  return x;
}
```

chap 4: Iteration

37. What value is returned as a result of the call mystery (6)?

(A) 1
(B) 5
(C) 6
(D) 8
(E) 13

38. Which of the following is true of method mystery ?

(A) x will sometimes be 1 at // Point B.
(B) x will never be 1 at // Point C.
(C) n will never be greater than 2 at // Point A.
(D) n will sometimes be greater than 2 at // Point C.
(E) n will always be greater than 2 at // Point B.

---

39. Consider the following output.

```
1  1  1  1  1
2  2  2  2
3  3  3
4  4
5
```

Which of the following code segments will produce this output?

chap 4: Iteration

(A)

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

(B)

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

(C)

```
for (int j = 1; j <= 5; j++)
{
  for (int k = 5; k >= 1; k--)
  {
    System.out.print(j + "  ");
  }
  System.out.println();
}
```

(D)

```
for (int j = 1; j <= 5; j++)
{
  for (int k = 5; k >= j; k--)
  {
    System.out.print(j + "  ");
  }
  System.out.println();
}
```

(E)

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

chap 4: Iteration

40. Consider the following incomplete method, which is intended to return the number of integers that evenly divide the integer inputVal. Assume that inputVal is greater than 0.

```
public static int numDivisors(int inputVal)
{
  int count = 0;
  for (int k = 1; k <= inputVal; k++)
  {
    if ( /* condition */ )
    {
      count++;
    }
  }
  return count;
}
```

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

(A) inputVal % k == 0

(B) k % inputVal == 0

(C) inputVal % k != 0

(D) inputVal / k == 0

(E) k / inputVal > 0

41. Consider the following code segment.

```
for (int outer = 1; outer <= 6; outer++)
{
  for (int inner = outer; inner <= 6; inner++)
  {
    if (inner % 2 == 0)
    {
      System.out.print(inner + "  ");
    }
  }
  System.out.println();
}
```

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

chap 4: Iteration

(A)
```
2   4   6
4   6
6
```

(B)
```
2   4   6
2   4   6
2   4   6
```

(C)
```
2   4   6
2   4   6
4   6
4   6
6
6
```

(D)
```
2   4   6
2   4   6
2   4   6
2   4   6
2   4   6
2   4   6
```

(E)
```
2   4
2   4
4
4
```

42. Consider the following output.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

Which of the following code segments will produce the output shown above?

chap 4: Iteration

(A)
```
for (int j = 1; j <= 6; j++)
{
  for (int k = 1; k < j; k++)
    System.out.print(" " + k);
  System.out.println();
}
```

(B)
```
for (int j = 1; j <= 6; j++)
{
  for (int k = 1; k <= j; k++)
    System.out.print(" " + j);
  System.out.println();
}
```

(C)
```
for (int j = 1; j <= 6; j++)
{
  for (int k = 1; k <= j; k++)
    System.out.print(" " + k);
  System.out.println();
}
```

(D)
```
for (int j = 1; j < 6; j++)
{
  for (int k = 1; k <= j; k++)
    System.out.print(" " + k);
  System.out.println();
}
```

(E)
```
for (int j = 1; j < 6; j++)
{
  for (int k = 1; k < j; k++)
    System.out.print(" " + k);
  System.out.println();
}
```

43. Consider the following code segment.

```
int num = 1;
while (num < 5)
{
     System.out.print("A");
     num += 2;
}
```

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

(A) `A`

(B) `AA`

(C) `AAA`

(D) `AAAA`

(E) `AAAAA`

chap 4: Iteration

44. Consider the following code segment.

```
for (int j = 0; j < 3; j++)
{
     for (int k = 0; k < 4; k++)
     {
         System.out.println("Fun");
     }
}
```

Which of the following best explains how changing the outer `for` loop header to `for (int j = 0; j <= 3; j++)` affects the output of the code segment?

(A) The output of the code segment will be unchanged.

(B) The string `"Fun"` will be printed more times because the outer loop will execute more times.

(C) The string `"Fun"` will be printed more times because the inner loop will execute more times in each iteration of the outer loop.

(D) The string `"Fun"` will be printed fewer times because the outer loop will execute fewer times.

(E) The string `"Fun"` will be printed fewer times because the inner loop will execute fewer times in each iteration of the outer loop.

45. Which of the following code segments will print all multiples of 5 that are greater than 0 and less than 100 ?

```
I. for (int k = 1; k < 100; k++)
{
   if (k % 5 == 0)
   {
      System.out.print(k + " ");
   }
}

II. for (int k = 1; k < 100; k++)
{
   if (k / 5 == 0)
   {
      System.out.print(k + " ");
   }
}

III. int k = 5;
while (k < 100)
{
   System.out.print(k + " ");
   k = k + 5;
}
```

chap 4: Iteration

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

46. Consider the following code segment.

```
for (int r = 3; r > 0; r--)
{
  int c;

  for (c = 1; c < r; c++)
  {
    System.out.print("-");
  }
  for (c = r ; c <= 3; c++)
  {
    System.out.print("*");
  }

  System.out.println();
}
```

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

(A)
```
--*
-**
***
```

(B)
```
*--
**-
***
```

(C)
```
***
-**
--*
```

(D)
```
***
**-
*--
```

(E)
```
--*
***
--*
```

chap 4: Iteration

47. Consider the following code segment.

```
int num = 2574;
int result = 0;

while (num > 0)
{
  result = result * 10 + num % 10;
  num /= 10;
}
System.out.println(result);
```

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

(A) 2
(B) 4
(C) 18
(D) 2574
(E) 4752

48. Consider the following code segment.

```
int sum = 0;
int k = 1;
while (sum < 12 || k < 4)
  sum += k;

System.out.println(sum);
```

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

(A) 6
(B) 10
(C) 12
(D) 15
(E) Nothing is printed due to an infinite loop.

chap 4: Iteration

49. Consider the following code segment.

```
int val = 1;
while (val <= 6)
{
     for (int k = 0; k <= 2; k++)
     {
         System.out.println("Surprise!");
     }
     val++;
}
```

How many times is the string `"Surprise!"` printed as a result of executing the code segment?

(A) 3
(B) 6
(C) 12
(D) 15
(E) 18

50. Consider the following code segment.

```
for (int k = 1; k <= 100; k++)
  if ((k % 4) == 0)
    System.out.println(k);
```

Which of the following code segments will produce the same output as the code segment above?

(A)
```
for (int k = 1; k <= 25; k++)
  System.out.println(k);
```

(B)
```
for (int k = 1; k <= 100; k = k + 4)
  System.out.println(k);
```

(C)
```
for (int k = 1; k <= 100; k++)
  System.out.println(k % 4);
```

(D)
```
for (int k = 4; k <= 25; k = 4 * k)
  System.out.println(k);
```

(E)
```
for (int k = 4; k <= 100; k = k + 4)
  System.out.println(k);
```

chap 4: Iteration

51. The question refer to the following code segment.

```
int k = a random number such that 1 ≤ k ≤ n ;
for (int p = 2; p <= k; p++)
  for (int r = 1; r < k; r++)
    System.out.println("Hello");
```

What is the maximum number of times that Hello will be printed?

(A) 2
(B) n - 1
(C) n - 2
(D) $(n - 1)^2$
(E) n^2

52. The question refer to the following code segment.

```
int k = a random number such that 1 ≤ k ≤ n ;
for (int p = 2; p <= k; p++)
  for (int r = 1; r < k; r++)
    System.out.println("Hello");
```

What is the minimum number of times that Hello will be printed?

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

chap 4: Iteration

53. Consider the following code segment.

```
for (int outer = 0; outer < n; outer++)
{
  for (int inner = 0; inner <= outer; inner++)
  {
    System.out.print(outer + " ");
  }
}
```

If n has been declared as an integer with the value 4, what is printed as a result of executing the code segment?

(A) 0 1 2 3

(B) 0 0 1 0 1 2

(C) 0 1 2 2 3 3 3

(D) 0 1 1 2 2 2 3 3 3 3

(E) 0 0 1 0 1 2 0 1 2 3

chap 4: Iteration

54. Consider the following method.

```
public int sol(int lim)
{
  int s = 0;

  for (int outer = 1; outer <= lim; outer++)
  {
    for (int inner = outer; inner <= lim; inner++)
    {
      s++;
    }
  }

  return s;
}
```

What value is returned as a result of the call `sol(10)`?

(A) 20
(B) 45
(C) 55
(D) 100
(E) 385

chap 4: Iteration

55. Consider the following method.

```
public String wordPlay(String word)
{
     String str = "";
     for (int k = 0; k < word.length(); k++)
     {
         if (k % 3 == 0)
         {
            str = word.substring(k, k + 1) + str;
         }
     }
     return str;
}
```

The following code segment appears in another method in the same class as `wordPlay`.

```
System.out.println(wordPlay("Computer Science"));
```

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

(A) `C`
(B) `ci tm`
(C) `eeStm`
(D) `ncepC`
(E) `eeSepC`

56. Consider the following code segment.

```
int num = 1;
for (int k = 2; k < 10; k++)
{
     num = 0;
     num = num + k;
}
```

What will be the value of `num` after the loop is executed?

(A) `2`
(B) `9`
(C) `10`
(D) `44`
(E) `45`

chap 4: Iteration

57. Consider the following method.

```
public static String changeStr(String str)
{
     String result = "";
     for (int i = str.length() - 1; i >= str.length() / 2; i -= 2)
     {
         result += str.substring(i, i + 1);
     }
     return result;
}
```

What value is returned as a result of the method call `changeStr("12345")` ?

(A) `"4"`

(B) `"53"`

(C) `"531"`

(D) `"543"`

(E) `"54321"`

58. Consider the following code segment.

```
for (int j = 1; j < 10; j += 2)
{
     System.out.print(j);
}
```

Which of the following code segments will produce the same output as the code segment above?

chap 4: Iteration

(A)
```
int j = 1;
while (j < 10)
{
     j += 2;
     System.out.print(j);
}
```

(B)
```
int j = 1;
while (j < 10)
{
     System.out.print(j);
     j += 2;
}
```

(C)
```
int j = 1;
while (j <= 10)
{
     j += 2;
     System.out.print(j);
}
```

(D)
```
int j = 1;
while (j >= 10)
{
     j += 2;
     System.out.print(j);
}
```

(E)
```
int j = 1;
while (j >= 10)
{
     System.out.print(j);
     j += 2;
}
```

chap 4: Iteration

59. Consider the following code segment.

```
int k = 1;
while (k < 20)
{
  if ((k % 3) == 1)
    System.out.print(k + " ");

  k++;
}
```

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

(A) 2 5 8 11 14 17
(B) 3 6 9 12 15 18
(C) 1 4 7 10 13 16 19
(D) 1 3 5 7 9 11 13 15 17 19
(E) 2 4 6 8 10 12 14 16 18 20