

1. What does the following code fragment write to the monitor?

```
int sum = 14;
if ( sum < 20 )
    System.out.print("Under ");
else
    System.out.print("Over ");
System.out.println("the limit.");
```

- a. Under
- b. Over
- c. Under the limit.
- d. Over the limit.

2. What does the following code fragment write to the monitor?

```
int sum = 7;
if ( sum > 20 )
{
    System.out.print("You win ");
}
else
{
    System.out.print("You lose ");
}
System.out.println("the prize.");
```

- a. You win
- b. You lose
- c. You win the prize.
- d. You lose the prize.

3. What does the following code fragment write to the monitor?

```
int sum = 21;
if ( sum != 20 )
    System.out.print("You win ");

else
    System.out.print("You lose ");

System.out.println("the prize.");
```

- a. You win
- b. You lose
- c. You win the prize.
- d. You lose the prize.

4. Evaluate (to true or false) each of the following expressions:

$14 \leq 14$     $14 < 14$     $-9 > -25$     $-25 > -9$

- a. true   true   true   true
- b. true   false   false   false
- c. true   false   true   true
- d. true   false   true   false

5. What does the following program output?

```
class question4
{
    public static void main ()
    {
        int totalCost = 6;
        int items = 12;
        System.out.println("cost per item: " + totalCost/items );
    }
}
```

- a. cost per item: .5
- b. cost per item: 0
- c. cost per item: 1
- d. cost per item: +6/12

6. What is the result of evaluating the following expression?

$(1/2 + 3.5) * 2.0$

- a. 8.0
- b. 8
- c. 7.0
- d. 0

7. What does the following code fragment write?

```
int depth = 12 ;
int temp = 42 ;

System.out.print("The water is: ")
if ( depth >= 8 )
    System.out.print("deep ");

if ( temp <= 50 )
    System.out.print("cold ");

System.out.println(" wet.");
```

- a. The water is:
- b. The water is: wet.
- c. The water is: deep cold wet.
- d. The water is wet cold deep.

8. Examine the following code:

```
int count = 7;
while ( count >= 4 )
{
    System.out.print( count + " " );
    count = count - 1;
}
System.out.println( );
```

What does this code print?

- a. 1 2 3 4 5 6 7
- b. 7 6 5 4
- c. 6 5 4 3
- d. 7 6 5 4 3

9. Examine the following code:

```
int count = -2 ;
while ( count < 3 )
{
    System.out.print( count + " " );
    count = count + 1;
}
System.out.println( );
```

What does this code print?

- a. -2 -1 1 2 3 4
- b. -2 -1 1 2 3
- c. -3 -4 -5 -6 -7
- d. -2 -1 0 1 2

10. Examine the following code:

```
int count = 1;
while ( count < 5 )
{
    System.out.print( count + " " );
}
System.out.println( );
```

What does this code print on the monitor?

- a. 1 2 3 4
- b. 1 2 3 4 5
- c. 2 3 4
- d. 1 1 1 1 1 1 1 1 1 1 . . . .

11. Examine the following code fragment:

```
int j = 1;
while ( j < 10 )
{
    System.out.println( j + " " );
    j = j + j%3;
}
```

What is output?

- a. 1 4 7
- b. 1 4 7 10
- c. 1 2 5 8
- d. 1 2 4 5 7 8

12. Fill the blank so that the following fragment prints out 0.2, 0.4, 0.6, 0.8, 1.0,

```
for ( int j = 2; j <= 10; j++ )
    System.out.print( _____ + ", " );
```

- a. j/10
- b. j%10
- c. (j+1.0)/10
- d. j/10.0

13. What does the following print on the monitor?

```
for ( int j = 0; j < 5; j++ )
{
    for ( int k = 0; k < 10; k++ )
        System.out.print( "*" );

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

- |                          |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|--------------------------|
| <input type="radio"/> a. | <input type="radio"/> b. | <input type="radio"/> c. | <input type="radio"/> d. |
| *****                    | *****                    | *****                    | *                        |
| *****                    | *****                    | *****                    | *                        |
| *****                    | *****                    | *****                    | *                        |
| *****                    | *****                    | *****                    | *                        |
| *****                    | *****                    | *****                    | *                        |
|                          | *****                    | *****                    |                          |
|                          | *****                    | *****                    |                          |
|                          | *****                    | *****                    |                          |
|                          | *****                    | *****                    |                          |
|                          | *****                    | *****                    |                          |
|                          | *****                    | *****                    |                          |
|                          | *****                    | *****                    |                          |
|                          | *****                    | *****                    |                          |

14. What are the legal indexes for the array ar, given the following declaration:

```
int[] ar = {2, 4, 6, 8 }
```

- a. 0, 1, 2, 3
- b. 1, 2, 3, 4
- c. 2, 4, 6, 8
- d. 0, 2, 4, 6

15. What is the output of the following code fragment:

```
int[] ar = {2, 4, 6, 8 };  
System.out.println( ar[0] + " " + ar[1] );
```

- a. 2 6
- b. 8
- c. 2 4
- d. 6 8

16. What is the output of the following code fragment:

```
int[] ar = {2, 4, 6, 8 };
```

```
ar[0] = 23;  
ar[3] = ar[1];
```

```
System.out.println( ar[0] + " " + ar[3] );
```

- a. 23 2
- b. 2 8
- c. 31
- d. 23 4

17. What is the output of the following code fragment:

```
int[] y = new int[5];
```

```
y[0] = 34;  
y[1] = 88;  
y[5] = y[0] + y[1]
```

```
System.out.println( y[0] + " " + y[1] + " " + y[5] );
```

- a. 34 88 0
- b. 34 88 122
- c. Run time error
- d. 0 34 88

18. What is the output of the following code fragment:

```
int[] z = new int[9];
```

```
z[0] = 7;  
z[1] = 3;  
z[2] = 4;
```

```
System.out.println( z[0] + z[1] + " " + z[5] );
```

- a. 10 0
- b. 7 3 0
- c. The program is defective and will not compile.
- d. 7 3 4

19. What is the output of the following code fragment:

```
int[] zip = new int[5];
```

```
zip[0] = 7;
```

```
zip[1] = 3;
```

```
zip[2] = 4;
```

```
zip[3] = 1;
```

```
zip[4] = 9;
```

```
System.out.println( zip[ 2 + 1 ] );
```

a. 4 3

b. 3 7

c. 4

d. 1

20. What is the output of the following code fragment:

```
int[] zip = new int[5];
```

```
zip[0] = 7;
```

```
zip[1] = 3;
```

```
zip[2] = 4;
```

```
zip[3] = 1;
```

```
zip[4] = 9;
```

```
int j = 3;
```

```
System.out.println( zip[ j-1 ] );
```

a. 7

b. 3

c. 4

d. 1

21. How many elements are present after the following code fragment has executed?

```
double[] ann = new double[ 7 ];
```

```
double[] bob;
```

```
bob = ann;
```

a. 2

b. 7

c. 14

d. 1

22. What is the output of the following code fragment:

```
int[] egArray = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
```

```
for ( int index= 0 ; index < 5 ; index++ )  
    System.out.print( egArray[ index ] + " " );
```

- a. 2 4 6 8
- b. 2 4 6 8 10
- c. 2 4 6 8 10 1
- d. 2 4 6 8 10 1 3 5 7 9

23. What is the output of the following code fragment:

```
int[] egArray = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
```

```
for ( int index= 0 ; index < egArray.length ; index++ )  
    System.out.print( egArray[ index ] + " " );
```

- a. 2 4 6 8
- b. 2 4 6 8 10
- c. 2 4 6 8 10 1
- d. 2 4 6 8 10 1 3 5 7 9

24. What is the output of the following code fragment:

```
int[] egArray = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
```

```
for ( int index= 0 ; index < egArray.length ; index = index + 2 )  
    System.out.print( egArray[ index ] + " " );
```

- a. 2 4 6 8 10 1 3 5 7 9
- b. 4 8 1 5 9
- c. 2 6 10 3 7
- d. 2 6 10 3 7 0

25. Fill in the blanks of the following code fragment so that the elements of the array are printed in reverse order, starting with the last element.

```
int[] egArray = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
```

```
for ( int index= _____ ; _____ ; _____ )  
    System.out.print( egArray[ index ] + " " );
```

- a. index = 0; index < egArray.length; index--
- b. index = length; index < 0; index--
- c. index = length-1; index > 0; index--
- d. index = egArray.length-1; index >= 0; index--

26. Examine the following program fragment:

```
int[] array = { 1, 4, 3, 6, 8, 2, 5};
int what = array[0];

// scan the array
for ( int index=0; index < array.length; index++ )
{
    if ( array[ index ] > what )
        what = array[ index ];
}
System.out.println( what );
```

What does the fragment write to the monitor?

- a. 1
- b. 5
- c. 1 4 3 6 8 2 5
- d. 8

27. Examine the following program fragment:

```
int[] array = { 1, 4, 3, 6, 8, 2, 5};
int what = array[0];

for ( int index=0; index < array.length; index++ )
{
    if ( array[ index ] < what )
        what = array[ index ];
}
System.out.println( what );
```

What does the fragment write to the monitor?

- a. 1
- b. 5
- c. 1 4 3 6 8 2 5
- d. 8



28. Examine the following program fragment:

```
int[] array = { 1, 4, 3, 6 };
int what = 0;

for ( int index=0; index < array.length; index++ )
{
    what = what + array[ index ] ;
}
System.out.println( what );
```

What does the fragment write to the monitor?

- a. 14
- b. 1
- c. 6
- d. 1 4 3 6

29. Fill in the blank in the following code fragment so that each element of the array is assigned twice the value of its index.

```
int[] array = new int[10];

for ( int index=0; index < array.length; index++ )
{
    _____
}

```

- a. `index = 2*index;`
- b. `array[ 2*index ] = 2*index;`
- c. `array[ index ] = 2*array[ index ];`
- d. `array[ index ] = 2*index;`

30. What does the following method do?

```
void spread ( int[] values )
{
    for ( int index= 1 ; index < values.length ; index++ )
        values[index] = values[0];
}

```

- a. It changes an array by copying the element in slot 0 to all other slots.
- b. It changes an array by making every element the same as the value of its index.
- c. It changes an array by making every element zero.
- d. It makes a change to the formal parameter but does not make a change to the caller's array.