QUIZIZZ Worksheets

Name

Class

Date

Unit-3 Test-1

Total questions: 35

Worksheet time: 2hrs 14mins

Instructor name: Mr. PRASHANT ATMAKURI

1. Which of the following is FALSE about arrays on Java

a) Length of array can be changed after creation of array

b) A java array is always an object

c) Arrays in Java are always allocated on heap

2.

```
int[] arr = {10, 5, 1, 20, 6, 25};
int sum = 0;
for (int k = 0; k <= arr.length; k++)
{
   sum += arr[k];
}
System.out.println("The sum is " + sum);
```

Consider the following code segment, which is intended to print the sum of all elements of an array. A runtime error occurs when the code segment is executed. Which of the following changes should be made so that the code segment works as intended?

a) The for loop header should be replaced with for (int k = 1; k <= arr.length - 1; k++).

b) The for loop header should be replaced with for (int k = 0; k <= arr.length; k--).

c) The statement in the body of the for loop should be replaced with sum += arr[k - 1].

d) The statement in the body of the for loop should be replaced with sum += arr[0].

e) The for loop header should be replaced with for (int k = 0; k < arr.length; k++).

3.
```
public int countTarget(int[] arr, int target)
{
  int count = 0;
  for (int j = 0; j <= arr.length; j++)  // line 4
  {
    if (arr[j] == target)
    {
      count++;
    }
  }
  return count;
}
```

The method countTarget below is intended to return the number of times the value targ appears in the array arr. The method may not work as intended. Which of the following changes, if any, can be made to line 4 so that the method will work as intended?

a) No change is necessary; the method works correctly as is.

b) Changing `j <= arr.length;` to `j < arr.length +`

c) Changing `j <= arr.length;` to `j < arr.length;`

d) Changing `j <= arr.length;` to `j < arr.length -`

e) Changing `int j = 0;` to `int j = 1;`

4.
```
public static void addOneToEverything(int[] numbers)
{
  for (int j = 0; j < numbers.length; j++)
  {
    numbers[j]++;
  }
}
```

I.
```
for (int num : numbers)
{
  num++;
}
```

II.
```
for (int num : numbers)
{
  num[j]++;
}
```

III.
```
for (int num : numbers)
{
  numbers[num]++;
}
```

Consider the following method. Which of the following code segments, if any, can be used to replace the body of the method so that numbers will contain the same values?

a) I only

b) I and III only

c) None of the code segments will return an equivalent result.

d) I, II, and III

e) II and III only

5.

```
public void changeIt(int[] list, int num)
{
  list = new int[5];
  num = 0;

  for (int x = 0; x < list.length; x++)
    list[x] = 0;
}

public void start()
{
  int[] nums = {1, 2, 3, 4, 5};
  int value = 6;

  changeIt(nums, value);

  for (int k = 0; k < nums.length; k++)
    System.out.print(nums[k] + " ");

  System.out.print(value);
}
```

Consider the following two methods that appear within a single class. What is printed as a result of the call start()?

a) 0 0 0 0 0 6

b) 0 0 0 0 0 0

c) changeIt will throw an exception.

d) 1 2 3 4 5 0

e) 1 2 3 4 5 6

6.

I.
```
int counter = 0;
int i = -1;
while (i <= nums.length - 2)
{
   i++;
   if (nums[i] < 0)
   {
      counter++;
   }
}
```

II.
```
int counter = 0;
for (int i = 1; i < nums.length; i++)
{
   if (nums[i] < 0)
   {
      counter++;
   }
}
```

III.
```
int counter = 0;
for (int i : nums)
{
   if (nums[i] < 0)
   {
      counter++;
   }
}
```

Consider an integer array nums, which has been properly declared and initialized with one or more values. Which of the following code segments counts the number of negative values found in nums and stores the count in counter ?

a) I and III only

b) I and II only

c) I and III only

d) I only

e) II only

7.
```
public static int countPeaks(int[] data)
{
  int numPeaks = 0;
  for ( /* missing loop header */ )
  {
    if (data[p - 1] < data[p] && data[p] > data[p + 1])
    {
      numPeaks++;
    }
  }
  return numPeaks;
}
```

Consider the following method, which is intended to return the number of *local maximu* values in an array. Local maximum values are array elements that are greater than both adjacent array elements. The first and last elements of an array have only a single adjacent element, so neither the first nor the last array element is counted by this metho For example, an array containing the values {3, 9, 7, 4, 10, 12, 3, 8} has two local maximum values: 9 and 12. Which of the following can replace /* *missing loop header* * so the method countPeaks works as intended?

a) `int p = 1; p < data.length - 1; p++`

b) `int p = data.length - 1; p > 0; p`

c) `int p = 0; p < data.length; p++`

d) `int p = 1; p < data.length; p-`

e) `int p = 0; p < data.length - 1; p++`

8.
```
myArray[0]=myArray[3];
myArray[3]=myArray[0];
System.out.println(myArray[3]*-2);
```

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

a) `ArrayIndexOutOfBoundsException`

b) 0

c) 32

d) 18

e) 8

9.
```
String[] people = {"Hank", "George", "Carol", "Johnsie"};
int d = 0;

for (int i=1; i<people.length; i++)
{
    d+=people.length;
}
System.out.println(d);
```

Consider the following code. What is printed as a result of executing this code?

a) 18
b) 12
c) A syntax error is encountered.
d) 22
e) 16

10. Consider the following method.

```
// Change each number evenly divisible by 7 to a 7
public static int[] change7(int[] numbers)
{
    for (int num : numbers)
    {
        if (num%7 == 0)
        {
            num=7;
        }
    }
    return numbers;
}
```

What will happen when the following call to **change7** is made?

```
int[] numbers = {15, 20, 21, 23, 12, 21};
change7(numbers);
```

a) **numbers** will contain {15, 20, 21, 23, 12, 7};
b) **numbers** will contain {15, 20, 7, 23, 12
c) **numbers** will contain {15, 20, 21, 23, 12, 21};
d) **numbers** will contain {15, 20, 7, 23, 12,

11. What are the legal indexes for the array `ar`, given the following declaration:
`int[] ar = {2, 4, 6, 8 }`

a) `0, 2, 4, 6`
b) 2, 4, 6, 8
c) `1, 2, 3, 4`
d) `0, 1, 2, 3`

12. Which of the following statements outputs the fifth value in the quizzes array?

a) System.out.println(quizzes[4]);
b) quizzes[4];
c) quizzes[5];
d) System.out.println(quizzes[5]);

13.
```
public int test(int v)
{
      for (int i = 0; i < a.length; i++)
      {
         if (a[i] == v)
            return i;
      }
      return -1;
}
```

Given the following method, what would test return if a = {0, 2, 3, 4} and v = 1?

a) 0
b) 2
c) -1
d) 1

14.
```
public int test(int v)
{
      for (int i = 0; i < a.length; i++)
      {
         if (a[i] == v)
            return i;
      }
      return -1;
}
```

Given the following method, what would test return if a = {0, 2, 3, 4} and v = 3?

a) 0
b) -1
c) 2
d) 1

15.
```
double[][] values = {{1.2, 9.0, 3.2},
                     {9.2, 0.5, -1.2},
                     {7.3, 7.9, 4.8}} ;
```

Given the code, what is in values[2][1] ?

a) 9.2
b) 7.3
c) 7.9
d) 0.5

16.
```
double[][] things = {{1.2, 9.0, 8.7, 1.0},
                     {9.2, 0.5, 0.0, 5.2},
                     {7.3, 7.9, 1.2, 3.9}} ;
```

Given the code, what is the value of things.length ?

a) 12
b) 4
c) 3
d) 7

17.
```
int[][] a = {{2, 4, 6, 8},
             {1, 2, 3, 4}};
```

How would you get the value "6" out of the following array?

a) d) a[2][0]
b) a [1][3]
c) c) a[0][2]
d) a [0][3]
e) e) a[3][1]

18.
```
int[][] a = {{2, 4, 6, 8},
             {1, 2, 3, 4}};
```

What would the statement a[1].length return?

a) 2
b) 3
c) 1
d) 4

19.
```
int[][] items = {{0, 1, 3, 4},
                 {4, 3, 99, 0},
                 {3, 2, 7, 70}} ;
```

Given the code, which of the following fragments prints out every element of *items*?

a)
```
for ( int row=0; row < items.length; row++ ){
    System.out.println();
    for ( int col=0; col < items[row].length; col++ )
        System.out.print( items[row][col] + " ");
}
```

b)
```
for ( int row=0; row < items.length; row++ )
    System.out.println();
    for ( int col=0; col < items.length; col
        System.out.print( items[row][col] +

}
```

c)
```
for ( int row=0; row < items.length; row++ ){
    for ( int row=0; row < items[row].length; row++ )
        System.out.print( items[row][col] + " ");
    System.out.println();
}
```

d)
```
for ( int row=0; row < items.length; row++ ){
    System.out.println();
    for ( int col=0; col < items[col].length; c
        System.out.print( items[row][col] + " "
}
```

20.

```
{
    double [][] arr = new double [] []
            {{5.0, 91.5, 8.9, 9.6},
            {7.0, 16.3, 7.4, 3.5}};
  for (int j = 0; j < arr.length; j++)
  {
      for (int k = 0; k < arr[0].length; k++)
    {
      if (arr[j][k] > j + k)
      {
        System.out.println("Hi");
      }
    }
  }
}
```

How many times will "Hi" be printed?

a) 8

b) 7

c) 16

d) 6

21. Which while loop will print only the bottom right half of the array
If this is the array: This is printed:
{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
3
5 6
7 8 9

a)
```
int b = 0;
while(b < a.length)
{
  int c = a[b].length -1;
  while(c >= a.length - b)
  {
    System.out.print(a[b][c] + " ");
    c--;
  }
  b++;
  System.out.println();
}
```

b)
```
int b = 0;
while(b < a.length)
{
  int c = 0;
  while(c < a[0].length)
  {
    System.out.print(a[b][c] + "
    c++;
  }
  b++;
  System.out.println();
}
```

c)
```
int b = 0;
while(b < a.length)
{
  int c = 0;
  while(c < a[0].length)
  {
    if(c >= a.length - b)
      System.out.print(a[b][c] + " ");
    c++;
  }
  b++;
  System.out.println();
}
```

d)
```
int b = 0;
while(b < a.length)
{
  int c = 0;
  while(c < a[0].length)
  {
    if(c >= a.length - b - 1)
      System.out.print(a[b][c] + "
    c++;
  }
  b++;
  System.out.println();
}
```

22. Consider the following code.

```
public static void main(String args[])
```

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

    if (numArray.length==4)
    {

        for (int i=0; i<numArray[0].length; i++)
        {
            System.out.print(numArray[i][2]);
        }
    }
    else
    {
        for(int i=2; i>-1; i--)
        {
            System.out.print(numArray[2][i]);
        }
    }
}
```

What is printed as a result of executing the code?

a) 548

b) out of bounds exception

c) 987

d) 767

e) 246

23. Consider the following code segment.

```
int[][] heights = new int[3][6];
heights[0][0] = 10;
heights[1][0] = 20;
heights[2][0] = 30;
double total = 0;
int count = 0;

for (int r=0; r < heights.length; r++)
{
    for (int c=0; c < heights[r].length; c++)
    {
        total += heights[r][c];
        count++;
    }
}
System.out.println((int) (total/count) + ", " +
    (int) (total/heights.length) + ", " +
    (int) (total/heights[0].length));
```

Which of the following is the output printed to the console after this segment of code is executed?

a) 3, 3, 3

b) 20, 20, 10

c) 10, 10, 10

d) 3, 20,10

e) 20, 20, 20

24. Consider the following code.

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

for (int i=0; i<matrix.length; i++)
{
    for (int j=i; j<matrix[0].length; j++)
    {
        System.out.print(matrix[i][j]);
    }
    System.out.println();
}
```

What is printed to the console as a result of executing this code?

a)
```
123456789
34567891
5678912
789123
```

b)
```
1234
345
56
7
```

c)
```
1
3
5
7
```

d) An array index out of bounds error is thr

e)

123456789

234567891

345678912

456789123

25. Consider the following code segment:

```
int[][] sum = new int [4][4];
for (int i = 0; i < num.length; i++)
{
   for (int k = 0; k < num[0].length; k++
   {
      num[i][k] = i * k;
   }
}
```

What are the contents of num after the code segment has executed?

a)

0	1	2	3
1	2	3	4
2	3	4	5
3	4	5	6

b)

0	0	0	0
0	1	2	3
0	2	4	6
0	3	6	9

c)

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

d)

0	0	0	0
1	2	3	4
2	4	6	8
3	6	9	12

e)

0 3 6 9

0 2 4 6

0 1 2 3

0 0 0 0

26. What is the list nums if it is initially [5, 3, 1] and the following code is executed?
nums.add(6);
nums.add(0, 4);
nums.remove(1);

a) [4, 5, 3, 6]
b) [4, 3, 1, 6]
c) [4, 3, 6]
d) [5, 3, 1, 6]

27. What index value is used to locate the last element in the nums ArrayList?

a) nums.length
b) nums.size()-1
c) nums.length()-1
d) nums.size()

28. Which statement below is the correct way to retrieve the first element in the nums ArrayList?

a) nums.get(0)
b) nums(0)
c) nums[0]
d) nums[1]

29. Given the ArrayList nums with the values [3, 7, 6, 0], what code below is the proper way to change the 7 to be a 5?

a) nums.set(5, 2)
b) nums[2] = 5
c) nums.set(7, 5)
d) nums.set(1, 5)

30. `ArrayList<Integer> foo = new ArrayList<Integer>();`

Write a line of code to add the value 23 to the end of ArrayList *foo*.

Ans. _____

31. `ArrayList<Integer> foo = new ArrayList<Integer>();`

Write a line of code to insert the value 42 to the ArrayList *foo* at index 5.

Ans. _____

32. `ArrayList<Integer> foo = new ArrayList<Integer>();`

Write a line of code to retrieve the value at index 9 from the ArrayList *foo*.

Ans. _____

33. `ArrayList<Integer> foo = new ArrayList<Integer>();`

Write a line of code to get the length of ArrayList *foo*.

Ans. _____

34. `ArrayList<Integer> foo = new ArrayList<Integer>();`

Write a line of code to remove the element at index 7 from the ArrayList *foo*.

Ans. _____

35. Which statement is the correct declaration and initialization of an ArrayList of String values?

a) String<ArrayList> name;
name = new String<ArrayList>();

b) ArrayList name;
name = new ArrayList<String>();

c) ArrayList<String> name;
name = new ArrayList<String>();

d) ArrayList<String> name;
name = ArrayList<String>();

Answer Keys

1. a) Length of array can be changed after creation of array
2. e) The for loop header should be replaced with for (int k = 0; k < arr.length; k++).
3. c)
4. c) None of the code segments will return an equivalent result.
5. e) 1 2 3 4 5 6
6. d) I only
7. a)
8. c) 32
9. b) 12
10. c)
11. d) `0, 1, 2, 3`
12. a) System.out.println(quizzes[4]
13. c) -1
14. c) 2
15. c) 7.9
16. c) 3
17. c) c) a[0][2]
18. d) 4
19. a)
20. b) 7
21. d)
22. d) 767
23. d) 3, 20,10
24. a)
25. b)
26. b) [4, 3, 1, 6]
27. b) nums.size()-1
28. a) nums.get(0)
29. d) nums.set(1, 5)
30. foo.add(23);
31. foo.set(5, 42);, foo.set(5,42);
32. foo.get(9);
33. foo.size();
34. foo.remove(7);
35. c) ArrayList<String> name; name = new ArrayList<String> ();