

Unit 07 MCQ: ArrayList**This quiz has 18 questions.**

1. Consider the following statement, which is intended to create an `ArrayList` named `a` to store only elements of type `Thing`. Assume that the `Thing` class has been properly defined and includes a no-parameter constructor.

```
ArrayList<Thing> a = /* missing code */;
```

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

- (A) `new Thing()`
- (B) `new ArrayList<Thing>()`
- (C) `new ArrayList(Thing)`
- (D) `new ArrayList(<Thing>)`
- (E) `new ArrayList<>(Thing)`

2. Consider the following statement, which is intended to create an `ArrayList` named `numbers` that can be used to store `Integer` values.

```
ArrayList<Integer> numbers=/* missing code */;
```

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

- I. `new Integer()`
 - II. `new ArrayList<Integer>`
 - III. `new ArrayList<Integer>()`
- (A) III only
 - (B) I and II only
 - (C) I and III only
 - (D) II and III only
 - (E) I, II, and III

3. Consider the following statement, which is intended to create an `ArrayList` named `arrList` to store elements only of type `String`.

```
/* missing code */ =new ArrayList<String>();
```

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

- (A) `ArrayList arrList()`
- (B) `ArrayList arrList`
- (C) `ArrayList<> arrList`
- (D) `ArrayList arrList<String>`
- (E) `ArrayList<String> arrList`

4. Consider the following code segment.

```
ArrayList<Integer> nums =  
    new ArrayList<Integer>();  
  
nums.add(3);  
nums.add(2);  
nums.add(1);  
nums.add(0);  
nums.add(0, 4);  
nums.set(3, 2);  
nums.remove(3);  
nums.add(2, 0);
```

Which of the following represents the contents of `nums` after the code segment has been executed?

- (A) `[2, 4, 3, 2, 0]`
- (B) `[3, 2, 0, 1, 0]`
- (C) `[4, 2, 0, 2, 0]`
- (D) `[4, 3, 0, 2, 0]`
- (E) `[4, 3, 0, 3, 0]`

Unit 07 MCQ: ArrayList

5. Consider the following code segment.

```
ArrayList<String> syllables =  
    new ArrayList<String>();  
  
syllables.add("LA");  
syllables.add(0, "DI");  
syllables.set(1, "TU");  
syllables.add("DA");  
syllables.add(2, syllables.get(0));  
syllables.remove(1);  
System.out.println(syllables.toString());
```

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

- Ⓐ [DI, DA, DI]
- Ⓑ [DI, DI, DA]
- Ⓒ [LA, LA, DA]
- Ⓓ [TU, DI, DA]
- Ⓔ [TU, TU, DA]

6. Consider the following code segment.

```
ArrayList<Integer> vals =  
    new ArrayList<Integer>();  
vals.add(vals.size(), vals.size());  
vals.add(vals.size() - 1, vals.size() + 1);  
vals.add(vals.size() - 2, vals.size() + 2);  
System.out.println(vals.toString());
```

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

- Ⓐ [0, 1, 2]
- Ⓑ [0, 2, 4]
- Ⓒ [1, 2, 3]
- Ⓓ [2, 1, 0]
- Ⓔ [4, 2, 0]

7. Consider the following code segment.

```
ArrayList<Integer> myList =  
    new ArrayList<Integer>();  
  
for (int i = 0; i < 4; i++) {  
    myList.add(i + 1);  
}  
  
for (int i = 0; i < 4; i++) {  
    if (i % 2 == 0) {  
        System.out.print(myList.get(i) + " ");  
    }  
}
```

What output is produced as a result of executing the code segment?

- Ⓐ 0 1 2 3
- Ⓑ 1 2 3 4
- Ⓒ 0 2
- Ⓓ 1 3
- Ⓔ 2 4

8. Consider the following code segment.

```
ArrayList<String> words =  
    new ArrayList<String>();  
words.add("mat");  
words.add("new");  
words.add("open");  
words.add("pet");  
int i = 0;  
while (i < words.size()) {  
    words.remove(i);  
    i++;  
}  
System.out.println(words.toString());
```

What is printed when the code segment is executed?

- Ⓐ []
- Ⓑ [new, pet]
- Ⓒ [open, pet]
- Ⓓ [new, open, pet]
- Ⓔ [mat, new, open, pet]

Unit 07 MCQ: ArrayList

9. Consider the following code segment.

```
ArrayList<String> arrList =
    new ArrayList<String>();
arrList.add("A");
arrList.add("B");
arrList.add("C");
arrList.add("D");
for (int i = 0; i < arrList.size(); i++) {
    System.out.print(arrList.remove(i));
}
```

What is printed when the code segment is executed?

- (A) AC
- (B) BD
- (C) ABC
- (D) ABCD
- (E) Nothing is printed.

10. Consider the following method definition. The method `isReversed` is intended to return `true` if `firstList` and `secondList` contain the same elements but in reverse order, and to return `false` otherwise.

```
/** Precondition:
 * firstList.size() == secondList.size() */
public static boolean
isReversed(ArrayList<Integer> firstList,
            ArrayList<Integer> secondList)
{
    for (int j=0; j<firstList.size()/2; j++)
    {
        if (firstList.get(j) !=
            secondList.get(secondList.size()-1-j))
        {
            return false;
        }
    }
    return true;
}
```

The method does not always work as intended. For which of the following inputs does the method NOT return the correct value?

- (A) When `firstList` is {1, 3, 3, 1} and `secondList` is {1, 3, 3, 1}
- (B) When `firstList` is {1, 3, 3, 1} and `secondList` is {3, 1, 1, 3}
- (C) When `firstList` is {1, 3, 5, 7} and `secondList` is {5, 5, 3, 1}
- (D) When `firstList` is {1, 3, 5, 7} and `secondList` is {7, 5, 3, 1}
- (E) When `firstList` is {1, 3, 5, 7} and `secondList` is {7, 5, 3, 3}

11. In the code segment below, `myList` is an `ArrayList` of integers. The code segment is intended to remove all elements with the value 0 from `myList`.

```
int j = 0;
while (j < myList.size()) {
    if (myList.get(j) == 0) {
        myList.remove(j);
    }
    j++;
}
```

The code segment does not always work as intended. For which of the following lists does the code segment NOT produce the correct result?

- (A) {0, 1, 2, 3}
- (B) {0, 1, 0, 2}
- (C) {1, 0, 0, 2}
- (D) {1, 2, 3, 0}
- (E) {1, 2, 3, 4}

12. In the code segment below, `numList` is an `ArrayList` of integers that is sorted in descending order. The code segment is intended to insert the integer value `val` into `numList` so that `numList` is still sorted in descending order.

```
int j = 0;
while (val != numList.get(j)) {
    j++;
}
numList.add(j, val);
```

The code segment does not always work as intended. Assuming that `numList` has been initialized to {3, 2, 1, 0}, for which value of `val` does the code segment NOT produce the expected result?

- (A) 4
- (B) 3
- (C) 2
- (D) 1
- (E) 0

Unit 07 MCQ: ArrayList

13. Consider the method `sequentialSearch`, which takes an `ArrayList` of `Integer` elements and an `int` value as parameters and returns the index of the first appearance of the target value in the list or `-1` if the target value does not appear in the list.

```
public static int
sequentialSearch(
    ArrayList<Integer>
    elements, int target)
{
    for (int j = 0; j < elements.size(); j++)
    {
        if(elements.get(j) == target) {
            return j;
        }
    }
    return -1;
}
```

Which of the following explains how replacing the `for` loop with:

`for (int j=(elements.size()-1); j>=0; j--)` will affect the behavior of `sequentialSearch`?

- Ⓐ The modification has no effect: the modified method will continue to return the index of the first appearance of the target value in the list, or `-1` if the target value does not appear in the list.
- Ⓑ The modified method will return the index of the last appearance of the target value in the list, or `-1` if the target value does not appear in the list.
- Ⓒ The modified method will throw an `IndexOutOfBoundsException`.
- Ⓓ The modified method will return `-1` regardless of the inputs.
- Ⓔ The modified method will not compile.

14. Consider the following search method.

```
public static int search(int[] arr,
                        int target)
{
    int result = -1;
    for(int j = 0; j < arr.length; j++) {
        if(arr[j] == target) {
            result = j; // Line 8
        }
    }
    return result;
}
```

Which of the following describes the effect of replacing the statement in line 8 of the method with `result = arr[j];`?

- Ⓐ The modified method will return the index of the first occurrence of `target` in `arr`.
- Ⓑ The modified method will return the index of the last occurrence of `target` in `arr`.
- Ⓒ The modified method will return `target` if `target` appears in `arr` and will return `-1` otherwise.
- Ⓓ The modified method will return `-1` if `target` appears in `arr` and will return `target` otherwise.
- Ⓔ The modified method will return `-1` for all possible inputs.

Unit 07 MCQ: ArrayList

15. Consider the method `seqSearch`, which implements a sequential search algorithm.

```
public int seqSearch(int[] arr, int target)
{
    for (int j = 0; j < arr.length; j++) {
        if (arr[j] == target) {
            return j;
        }
    }
    return -1;
}
```

Consider another method, `seqSearch2`, which modifies `seqSearch` to use an enhanced for loop.

```
public int seqSearch2(int[] arr, int target)
{
    for (int j : arr) {
        if (j == target) {
            return j;
        }
    }
    return -1;
}
```

Which of the following best describes the difference in the behavior of `seqSearch2` relative to `seqSearch` as a result of the modification?

- (A) The modification in `seqSearch2` has no effect: `seqSearch2` will always behave exactly as `seqSearch` does.
- (B) The modification in `seqSearch2` will cause a compilation error.
- (C) The modification in `seqSearch2` will cause an `IndexOutOfBoundsException` to be thrown for some inputs.
- (D) The modification in `seqSearch2` will cause `-1` to be returned for all inputs.
- (E) The modification in `seqSearch2` will cause the value of `target` to be returned instead of the index of `target` in cases where `target` appears in `arr`.

16. Consider the following correct implementation of the insertion sort algorithm.

```
1 public static void insertionSort(
2     int[] e)
3 {
4     for (int j = 1; j < e.length; j++) {
5         int temp = e[j];
6         int possibleIndex = j;
7         while (possibleIndex > 0 &&
8             temp < e[possibleIndex-1])
9             {
10                e[possibleIndex] =
11                    e[possibleIndex-1];
12                possibleIndex--;
13            }
14        elements[possibleIndex] = temp;
15    }
16 }
```

The following declaration and method call appear in a method in the same class as `insertionSort`.

```
int[] arr = {10, 8, 3, 4};
insertionSort(arr);
```

How many times is the statement `possibleIndex--` in line 12 of the method executed as a result of the call to `insertionSort`?

- (A) 0
- (B) 1
- (C) 4
- (D) 5
- (E) 6

Unit 07 MCQ: ArrayList

17. Consider the following correct implementation of the insertion sort algorithm.

```

1 public static void insertionSort(
2     int[] e)
3 {
4     for (int j = 1; j < e.length; j++) {
5         int temp = e[j];
6         int possibleIndex = j;
7         while (possibleIndex > 0 &&
8             temp < e[possibleIndex-1])
9             {
10                e[possibleIndex] =
11                    e[possibleIndex-1];
12                possibleIndex--;
13            }
14        e[possibleIndex] = temp;
15    }
16 }
```

The following declaration and method call appear in a method in the same class as `insertionSort`.

```
int[] nums = {8, 7, 5, 4, 2, 1};
insertionSort(nums);
```

How many times is the statement `e[possibleIndex] = temp;` in line 14 of the method executed as a result of the call to `insertionSort`?

- (A) 3
- (B) 4
- (C) 5
- (D) 6
- (E) 7

18. Consider the following correct implementation of the selection sort algorithm.

```

1 public static void selectionSort(
2     int[] e)
3 {
4     for (int j=0; j<e.length-1; j++) {
5         int minIndex = j;
6         for(int k=j+1; k<e.length; k++) {
7             if(e[k] < e[minIndex]) {
8                 minIndex = k;
9             }
10        }
11        if (j != minIndex) {
12            int temp = e[j];
13            e[j] = e[minIndex];
14            e[minIndex] = temp;
15        }
16    }
17 }
```

The following declaration and method call appear in a method in the same class as `selectionSort`.

```
int[] arr = {30, 40, 10, 50, 20};
selectionSort(arr);
```

How many times is the statement `e[minIndex] = temp;` in line 14 of the method executed as a result of the call to `selectionSort`?

- (A) 1
- (B) 2
- (C) 3
- (D) 4
- (E) 5