List Easy Multiple Choice Questions

These problems are easier than most of those that you will usually see on the AP CS A exam.

1: Which index is the last element in a list called nums at?

(A) nums.length (B) nums.length – 1 (C) nums.size() (D) nums.size() - 1

2: Which of the following is a reason to use an ArrayList instead of an array?

(A) An array has faster access to its elements than a list does.  
(B) An array knows it length, but a list doesn't know its length.  
(C) An ArrayList can allocate more space than it needs.

3: Which of the following is a reason to use an ArrayList instead of an array?

(A) An ArrayList can grow or shrink as needed, while an array is always the same size.  
(B) You can use a for-each loop on an ArrayList, but not in an array.  
(C) You can store objects in an ArrayList, but not in an array.  
4: Which of the following is the correct way to get the first value in a list called nums?

(A) nums[0]  
(B) nums[1]  
(C) nums.first()  
(D) nums.get(0)  
(E) nums.get(1)  
5: Which of the following is the correct way to set the second value in a list called nums to 5?

(A) nums[1] = 5;  
(B) nums[2] = 5;  
(C) nums.set(5, 1);  
(D) nums.set(1, 5);  
(E) nums.set(2, 5);

6: Which of the following is the correct way to remove the value 3 from the list nums=[5,3,2,1]?

(A) nums.remove(3); (B) nums.remove(new Integer(3));  
(C) nums.remove(1); (D) nums.remove(2)  
7: Which of the following is the correct way to add 2 between the 1 and 3 in the following list nums = [1, 3, 4]?

(A) nums.add(2, 0);  
(B) nums.add(2, 1);  
(C) nums.add(0, 2);  
(D) nums.add(1, 2);  
(E) nums.add(2, 2);  
8: Which of the following is *false* about an interface?

(A) It is a type of class.  
(B) The methods in an interface will be public and abstract.  
(C) It is like a contract in that the class that implements the interface must provide the methods defined in the interface.  
(D) You can create an object of an interface type.  
9: What will print when the following code executes?

List<Integer> list1 = **new** ArrayList<Integer>();

list1.add(**new** Integer(1));

list1.add(**new** Integer(2));

list1.add(**new** Integer(3));

list1.remove(1);

System.out.println(list1);

(A) [2, 3]  
(B) [1, 2, 3]  
(C) [1, 2]  
(D) [1, 3]

10: What will print when the following code executes?

List<String> list1 = **new** ArrayList<String>();

list1.add("Anaya");

list1.add("Layla");

list1.add("Sharrie");

list1.set(0, "Destini");

list1.add(0, "Sarah");

System.out.println(list1);

(A) ["Sarah", "Destini", "Layla", "Sharrie"]  
(B) ["Sarah", "Destini", "Anaya", "Layla", "Sharrie"]  
(C) ["Sarah", "Layla", "Sharrie"]

(D) ["Destini", "Layla", "Sharrie", "Sarah"]

List Medium Multiple Choice Questions

These problems are like those you will see on the AP CS A exam.

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

List<Integer> list1 = **new** ArrayList<Integer>();

list1.add(**new** Integer(1));

list1.add(**new** Integer(2));

list1.add(**new** Integer(3));

list1.set(2, **new** Integer(4));

list1.add(2, **new** Integer(5));

list1.add(**new** Integer(6));

System.out.println(list1);

Top of Form

(A) [1, 2, 3, 4, 5]  
(B) [1, 2, 4, 5, 6]  
(C) [1, 2, 5, 4, 6]  
(D) [1, 5, 2, 4, 6]  
2: Given the following code and assume that nums initially contains [0, 0, 4, 2, 5, 0, 3], what will nums contain as a result of executing numQuest?

**private** List<Integer> nums;

*// precondition: nums.size() > 0;*

*// nums contains Integer objects*

**public** void numQuest()

{

int k = 0;

Integer zero = **new** Integer(0);

**while** (k < nums.size())

{

**if** (nums.get(k).equals(zero))

nums.remove(k);

**else**

k++;

}

}

Top of Form

(A) [0, 4, 2, 5, 3]  
(B) [3, 5, 2, 4, 0, 0, 0]  
(C) [0, 0, 0, 4, 2, 5, 3]  
(D) [4, 2, 5, 3]  
(E) [0, 0, 4, 2, 5, 0, 3]  
3: Which of the following best describes the behavior of process1 and process2 (shown below)?

**public** **static** List<Integer> process1(int n)

{

List<Integer> someList = **new** ArrayList<Integer>();

**for** (int k = 0; k < n; k++)

someList.add(k);

**return** someList;

}

**public** **static** List<Integer> process2(int n)

{

List<Integer> someList = **new** ArrayList<Integer>();

**for** (int k = 0; k < n; k++)

someList.add(k, k);

**return** someList;

}

Top of Form

(A) Both methods produce the same result, and process1 is faster than process2.  
(B) The two methods produce different results and take the same amount of time.  
(C) The two methods produce different results, and process1 is faster than process2.  
(D) The two methods produce different results, and process2 is faster than process1.  
(E) Both methods produce the same result and take the same amount of time.  
8-12-4: What is printed as a result of executing the following code segment?

List<Integer> aList = **new** ArrayList<Integer>();

aList.add(**new** Integer(1));

aList.add(**new** Integer(2));

aList.add(1, **new** Integer(5));

aList.set(1, **new** Integer(4));

aList.add(**new** Integer(6));

aList.add(**new** Integer(3));

System.out.println(aList);

Top of Form

(A) [1, 2, 5, 4, 6, 3]  
(B) [6, 5, 4, 3, 2, 1]  
(C) [1, 2, 3, 4, 5, 6]  
(D) [1, 4, 2, 6, 3]  
(E) [1, 2, 4, 6, 3]  
5: What is printed as a result of executing the following code segment?

List<Integer> aList = **new** ArrayList<Integer>();

aList.add(**new** Integer(1));

aList.add(**new** Integer(2));

aList.remove(1);

aList.add(1, **new** Integer(3));

aList.set(1, **new** Integer(4));

aList.add(**new** Integer(5));

System.out.println(list);

Top of Form

(A) [1, 2, 3, 4, 5]  
(B) [1, 4, 5]  
(C) [1, 4, 3, 5]  
(D) [2, 4, 5]  
(E) [2, 4, 3, 5]  
6: What is printed as a result of executing the following code segment?

List<String> list1 = **new** ArrayList<String>();

list1.add("a");

list1.add("b");

list1.add(0,"c");

list1.add(1, "d");

list1.set(2, "e");

list1.add("f");

System.out.println(list1);

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

Top of Form

(A) [c, d, e, b]  
(B) [c, d, e, b, f]  
(C) [c, a, e, b, f]  
(D) [c, d, e, a, b, f]  
(E) [c, a, e, d, b, f]  
7: Given the list nums = [4, 2, 3, 4, 5] what is the result after executing nums.remove(4)?

Top of Form

(A) [2, 3, 4, 5]  
(B) [2, 3, 5]  
(C) [4, 2, 3, 5]  
(D) [4, 2, 3, 4]  
8: What is printed as a result of executing the following code segment?

List<String> list1 = **new** ArrayList<String>();

list1.add("a");

list1.add("b");

list1.add(0,"c");

list1.set(1, "d");

list1.set(0, "e");

list1.add("b");

System.out.println(list1);

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

Top of Form

(A) [e, d, b]  
(B) [e, d, b, b]  
(C) [e, d, a, b, b]  
(D) [e, d, a, b]  
9: Assume that numList has been initialized with the following Integer objects: [0, 1, 2, 3, 4]. What is the value of numList after mystery(5) executes?

**private** List<Integer> numList;

**public** void mystery(int n)

{

**for** (int i = 0; i < n; i++)

{

Integer obj = numList.remove(0);

numList.add(obj);

}

}

Top of Form

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

Bottom of Form

10: Assume that numList has been initialized with the following Integer objects: [5, 7, 8, 12]. Which of the following shows the values in numList after a call to mystery(11)?

**private** List<Integer> numList;

**public** void mystery(int value)

{

int i = 0;

**while** (i < numList.size() && numList.get(i) < value)

{

i++;

}

numList.add(i, value);

}

(A) [5, 7, 8, 12]  
(B) [5, 7, 8, 11, 12]  
(C) [11, 5, 7, 8, 12]  
(D) [5, 7, 8, 12, 11]  
(E) [5, 7, 11, 8, 12]

List Hard Multiple Choice Questions

1: What is in the list nums if it initially contained {5, 3, 1} and the following code is executed?

nums.add(6);

nums.add(0,4);

nums.remove(1);

(A) [5, 3, 1, 6] (B) [4, 3, 1, 6]  
(C) [4, 3, 6]  
(D) [5, 3, 6]  
(E) [4, 5, 3, 6]

2: Assume that nums has been created as an ArrayList object and initially contains the following Integer values: [0, 0, 4, 2, 5, 0, 3, 0]. What will nums contain as a result of executing the following method numQuest?

**private** List<Integer> nums;

*//precondition: nums.size() > 0*

*//nums contains Integer objects*

**public** void numQuest() {

int k = 0;

Integer zero = **new** Integer(0);

**while** (k < nums.size()) {

**if** (nums.get(k).equals(zero))

nums.remove(k);

k++;

}

}

(A) [0, 0, 4, 2, 5, 0, 3, 0]  
(B) [3, 5, 2, 4, 0, 0, 0, 0]  
(C) [0, 0, 0, 0, 4, 2, 5, 3]  
(D) [4, 2, 5, 3]  
(E) [0, 4, 2, 5, 3]

Answer Key:

Easy question

1. D Correct! The last element is at the size of the list minus

2. c Correct! Every time an ArrayList fills up a new array is created that is twice as big. This can lead to extra space that is wasted.

3. A Correct! This is the main advantage to an ArrayList.

4. D Correct! Use the get method to get a value from a list and the first element in a list is at index 0.

5. D Correct! This sets the second value in the list to 5.

6. c Correct! This would remove the value at index 1 which is 3.

7. D Correct! This would add 2 at index 1 which would result in [1, 2, 3, 4]

8. D Correct! You can not create an object of an interface type. This is why you create a ``List`` using the ArrayList class which implements the ``List`` interface.

9. D Correct! This removes the value at index 1 which is 2.

10. A Correct! The list is first ["Anaya", "Layla", "Sharrie"] and then ["Destini, "Layla", "Sharrie"] and finally ["Sarah", "Destini, "Layla", "Sharrie"]

Medium question

1. C Correct! The add method that takes just an object as a parameter adds that object to the end of the list. The set replaces the value at that index with the new value. The add with parameters of an index and an object puts the passed object at that index and moves any existing values by one index to the right (increments the index).
2. D Correct! This shows all zeros removed. Since k is only incremented if a value wasn't removed this will work correctly.
3. E Correct! The method process1 adds to the end of the list each time through the loop. The method process2 also adds to the end of the list each time through the loop. The only difference would be if there were values in the list in process2. Any existing values would be moved to the right. But, there are no existing values in the list at that index or beyond.
4. D Correct! The add with an index of 2 and a value of 5 adds the 5 at index 2 not 1. Remember that the first index is 0.
5. B Correct! The list is [1], then [1, 2], then [1], then [1, 3], then [1, 4], then [1, 4, 5].
6. B Correct! This list is [a], then [a, b], then [c, a, b], then [c, d, a, b], then [c, d, e, b], then [c, d, e, b, f]
7. D Correct! This removes the value at index 4 which is 5.
8. B Correct! The list is [a], [a, b], [c, a, b], [c, d, b], [e, d, b], and then [e, d, b, b]
9. C Correct! Each value is removed one at a time and added to the end of the list which results in the same list.
10. B Correct! This will add the value at the correct location in a list in ascending order.