**List exercise:**

**Check your understanding:**

**Part1**

1. 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.add(2, **new** Integer(4));

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

System.out.println(list1);

Top of Form

(A) [1, 2, 3, 4, 5]  
(B) [1, 4, 2, 3, 5]  
(C) [1, 2, 4, 3, 5]  
(D) [1, 2, 4, 5]

2. What will print when the following code executes?

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

list1.add("Anaya");

list1.add("Layla");

list1.add("Sharrie");

list1.add(1, "Sarah");

System.out.println(list1);

Top of Form

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

3. What will print when the following code executes?

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

list1.add(5);

list1.add(4);

list1.add(3);

list1.add(1, 2);

System.out.println(list1);

Top of Form

(A) [5, 4, 3, 2]  
(B) [5, 4, 1, 3]  
(C) [2, 5, 4, 3]  
(D) [5, 2, 4, 3]

4. What will print when the following code executes?

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

list1.add(1);

list1.add(3);

list1.add(2);

list1.add(1);

System.out.println(list1);

Top of Form

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

Bottom of Form

Bottom of Form

1. C  add(2, new Integer(4)) will put the 4 at index 2, but first move the 3 to index 3.Bottom of Form
2. Answer A The add(1, "Sarah") will move any current items to the right and then put "Sarah" at index 1.
3. D This adds the 2 to index 1, but first moves all other values past that index to the right.
4. 4. BThe add method adds each object to the end of the list and lists can hold duplicate objects.

Part 2

1: 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.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: 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(1, "Destini");

list1.add(1, "Sarah");

System.out.println(list1);

Top of Form

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

3: What will print when the following code executes?

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

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

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

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

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

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

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

System.out.println(numList);

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]

4: 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);

Top of Form

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

5: 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(2);

System.out.println(list1);

Top of Form

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

1. C The set will change the item at index 2 to 4. The add of 5 at index 2 will move everything else to the right and insert 5. The last add will be at the end of the list.Bottom of Form
2. D The list is first ["Anaya", "Layla", "Sharrie"] and then changes to ["Anaya", Destini", "Sharrie"] and then to ["Anaya", "Sarah", "Destini", "Sharrie"]
3. D. add without an index adds at the end, set will replace the item at that index, add with an index will move all current values at that index or beyond to the right.
4. D The item at index 1 is removed and the 3 is moved left.
5. C The 3 (at index 2) is removed

Part 3 **Check your understanding**

**Looping list**

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

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

**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++;

}

}

Top of Form

(A) [0, 4, 2, 5, 3]  
(B) [3, 5, 2, 4, 0, 0, 0, 0]  
(C) [0, 0, 0, 0, 4, 2, 5, 3]  
(D) [4, 2, 5, 3]  
How many times it loop through?

What is the result if change 3 to be 0?

A Correct! Incrementing the index each time through the loop will miss when there are two zeros in a row.

(6)

[0, 4, 2, 5, 0]

What is the corresponding answer on the right ? t ? . 

