ArrayList Test 2

1. Which of the following is a reason to use an ArrayList instead of an array?
2. An ArrayList allows faster access to the kth item than array does.
3. An ArrayList resizes itself as necessary when items are added, but an array does not.
4. An ArrayList always uses less memory than an array.
5. An ArrayList can store objects and an array can only store primitives.

1. Which of the following will correctly instantiate an ArrayList of integer values?
2. List<Integer> list = new ArrayList<Integer>()
3. ArrayList<int> list = new ArrayList<int>()
4. List<int> list = new ArrayList<int>()
5. I only
6. II only
7. III only
8. I and II
9. Consider the following code segment.

ArrayList<String> items = new ArrayList<String> ();

items.add("A");

items.add("B");

items.add("C");

items.add(0, "D");

items.remove(1);

items.add(2, "E");

System.out.println(items);

What of the following represents items after the code above is executed?

1. [A, B, E, C]
2. [D, B, E, C]
3. [D, A, B, E]
4. [D, E, B, C]
5. Assume that an ArrayList of integers named **list** has been instantiated. What is output by the following code segment?

for (int k = 0; k < 10; k+=2)  
{  
 list.add(k);

}

for (int n : list)  
{  
 System.out.print(n + " ");  
}

1. 0 0 0 0 0
2. 0 2 4 6 8
3. 8 6 4 2 0
4. 0 2 4 6 8 10
5. Consider the following method that is intended to modify its parameter list by replacing all occurrences of name with newValue.

public void replace(ArrayList<String> list,   
 String name, String newValue)  
{  
 for (int j = 0; j < list.size(); j++)  
 {  
 if ( /\* expression \*/ )  
 {  
 list.set(j, newValue);  
 }  
 }  
}

Which of the following can be used to replace /\* expression \*/ so that replace works as  
 intended?

1. list[j].equals(name)
2. list[j] == name
3. list.get(j).equals(name)
4. list.get(j) == name

1. Consider the follow method.

public void mystery()

{

for(int i=0; i < list.size(); i++)

{

if(list.get(i) == 1)

list.remove(i);

}

}

Assume that an ArrayList of integers named **list** has been instantiated and initialized with the following Integer objects.

[0, 1, 1, 1, 1, 0, 1, 0, 0, 1];

Which of the following represents **list** after a call to mystery?

1. [0, 1, 1, 0, 0, 0]
2. [0, 1, 0, 0, 0]
3. [0, 0, 1, 1, 0, 0]
4. [0, 0, 0, 0]

**Questions 7-8 refer to the following two classes.**

public class Info   
{  
 private String name;

public Info(String n)   
 {   
 name = n;  
 }

public String getName()   
 {  
 return name;  
 }

public void setName(String n)   
 {  
 name = n;  
 }

}

public class Demo  
{  
 public static void main(String[] args)  
 {  
 ArrayList<Info> list= new ArrayList<Info>();  
  
 list.add(new Info("A"));  
 list.add(new Info("B"));  
 list.add(new Info("C"));  
 list.add(new Info("D"));

for (Info obj : list)  
 {  
 /\* expression 1 \*/   
 }

}  
 }

1. Which of the following can replace /\* expression 1 \*/ so that all of the Info objects in list have their name field assigned a value of "X"?
2. list[i].setName("X")
3. obj = "X"
4. obj.setName("X")
5. obj.getName().setName("X")
6. Which of the following **cannot** be added to the Demo class’s main method so that the entire ArrayList is printed displaying the name field of each Info object in list.
7. for (int k = 0; k < list.size(); k++)

{  
 System.out.println(list.get(k).getName());  
}

1. for (int k = 0; k < list.size(); k++)  
   {  
    Info obj = list.get(k);  
    System.out.println(obj.getName());  
   }
2. for (int k = 0; k < list.size(); k++)  
   {  
    Info obj = list.get(k);  
    System.out.println(obj);  
   }
3. for (Info obj : list)  
   {  
    System.out.println(obj.getName());  
   }

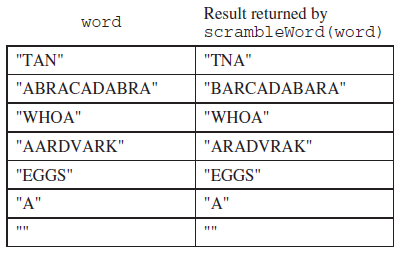
**Free Response**

This question involves reasoning about strings made up of uppercase letters. You will implement two related methods that appear in the same class (not shown). The first method takes a single string parameter and returns a scrambled version of that string. The second method takes a list of strings and modifies the list by scrambling each entry in the list. Any entry that cannot be scrambled is removed from the list.

1. Write the method scrambleWord, which takes a given word and returns a string that contains a scrambled version of the word according to the following rules.

* The scrambling process begins at the first letter of the word and continues from left to right.
* If two consecutive letters consist of an "A" followed by a letter that is not an "A", then the two letters are swapped in the resulting string.
* Once the letters in two adjacent positions have been swapped, neither of those two positions can be involved in a future swap.

The following table shows several examples of words and their scrambled versions.



Complete method scrambleWord below.

