

1) 2pts This question is about the Java ArrayList concepts of size and capacity.

Circle the **best** definition of size, put a box around the best definition of capacity:

- A) Has zero elements
- B) Has ten elements
- C) The maximum number of elements ever inserted into the ArrayList
- D) The maximum amount of memory one can allocate on a given computer
- E) The maximum number of elements that can be put in an ArrayList before ensureCapacity must be called
- F) The number of actual elements that can be accessed in an ArrayList

2. WARNING: The following code may not behave as the programmer intended! This is a 2 part problem. This code is supposed to print out the number of zeros in an array.

A) 1pt What value is returned by this code when the input is {2, 0, 0, 3, 0}

```
public static double countZeros(int[] foo)
{
    double count = 0;
    for (int i = 0; i < foo.length; i++)
    {
        if (foo[i] == 0)
            count++;
        else
            return count;
    }
    return count;
}
```

A) 0 B) 1 C) 2 D) 3 E) none of the above

B) 1pt In one SHORT sentence, describe how you would fix this code to perform its job.

remove this code
One cannot know how many total 0s there are in the array without looping over the WHOLE array. Return AFTER the loop!

Below is MOST of a method to add an element to an ArrayList that is being kept in inverseSorted order – that is from largest (at index 0) to smallest. Select the appropriate code that could be used to control the inserting of a new Double value into the ArrayList

```
public static void insertInDecreasingOrder(ArrayList foo, Double d)
{
    if (foo.size() == 0)
        foo.add(d);
    else
    {
        for (int i = 0; i < foo.size(); i++)
        {
            <<< 3 SELECT THE IF STATEMENT FRAGMENT TO GO HERE >>>
            {
                <<< 4 SELECT THE IF BODY TO GO HERE >>>
            }
        }
        foo.add(d);
    }
}
```

3. 1pt (Hint: You can assume that two Double values can be compared with >, <, == or !=)

- A) if ((Double)foo.get(i) > d)
- ☒ B) if ((Double)foo.get(i) < d)
- C) if ((Double)foo.get(i) == d)
- D) if ((Double)foo.get(i) != d)

inserts over and over forever!

loop left to right until I find something smaller than me. Insert me in THAT place.

4. 2pts

- A) foo.add(i,d); (1pt)
- ☒ B) foo.add(i,d); return;
- C) foo.add(d);
- D) foo.add(d); (1pt) return;

2pt

i	foo.get(i)
0	9
1	8
2	6

insert 7 at index 2
BREAK

5. (2pts) Draw the ArrayList after the following code is executed. Draw the **complete internal array** (e.g. with appropriate capacity) as we have been doing in class on similar problems. At each place there is an arrow show the contents of the ArrayList.

```
ArrayList foo = new ArrayList();
for (int i = 0; i < 5; i++)
    foo.add(new Integer(i));
foo.add(1, new Integer(20));
foo.remove(2);
foo.ensureCapacity(11);
foo.set(4, new Integer(33));
```

index 2

