

Bardia Ardakanian

9831072

Q1:

1.true

2. false. In Java, all objects are dynamically allocated on Heap.

3.false

1. Generic parameters

2. `List<MyType> fixed = Arrays.asList(new MyType[100]);`

3. anonymous objects

4. object addresses

5.

Q2:

1.

- **ArrayList** internally uses a dynamic array to store the elements. **LinkedList** internally uses a doubly **linked list** to store the elements.

- Manipulation with **ArrayList** is slow because it internally uses an array. If any element is removed from the array, all the bits are shifted in memory.

2.

Linked list. Because In LinkedList, the manipulation is fast because no shifting is required.

3.

**Iterable** is one of **the** main interfaces of **the** collection classes in **Java**. **The** Collection interface extends **Iterable** and hence all child classes of Collection also implement **Iterable**. **Iterable** has only one method that produces an **Iterator**: ... This **Iterator** can then be used to iterate over **the** elements in **the Iterable**.

An object that implements this interface allows it to be the target of the “foreach” statement. The for-each loop is **used** for iterating over arrays, collections etc.

Sources:

Javapoint.com

Tutorialspoint.com

Q3:

```
public class Main
{
    public static void main(String[] args) {
        int[] array = {14, 19, 8, 7, 48, 16, 63, 27};
        int n = array.length;
        for (int i = 0 ; i < n; i++)
        {
            int j = i;
            while (j>0 && array[j-1]>array[j])
            {
                int key = array[j];
                array[j] = array[j-1];
                array[j-1] = key;
                j--;
            }

        }

        for (int i : array)
        {
            System.out.print(i + " ");
        }
    }
}
```

output : 7 8 14 16 19 27 48 63