1. What is an Array List?

**Java Array List** is perhaps the simplest and one of the most used data structure implementation classes of the Java API Library. It is a part of the Java Collection Framework under the java.util package. On one hand, it behaves like a normal array, providing all the benefits of it and, on the other; it is a generic re-sizable collection implementation of the List interface. Java Array List is especially used for managing a large number of objects.

It provides us dynamic arrays in Java. Though, it may be slower than standard arrays but can be helpful in programs where lots of manipulation in the array is needed.

Most of the developers **choose Array list over Array** as it’s a very good alternative of traditional java arrays. The issue with arrays is that they are of fixed length so if it is full we cannot add any more elements to it, likewise if there are number of elements gets removed from it the memory consumption would be the same as it doesn’t shrink. On the other Array List can dynamically grow and shrink as per the need. Apart from these benefits Array List class enables us to use predefined methods of it which makes our task easy. Let’s see the Array List example first then we will discuss its methods and their usage.

2. What is Iterator?

**Iterator in Java** is nothing but a traversing object, made specifically for Collection objects like List and Set. we have already aware about different kind of traversing methods like for-loop ,while loop, do-while, for each lop etc, they all are  index based traversing but as we know Java is purely [object oriented](http://javarevisited.blogspot.com/2012/03/10-object-oriented-design-principles.html)language there is always possible ways of doing things using objects so Iterator is a way to traverse as well as access the data from the collection. Even with traversing with object we have Enumeration, Iterator and List Iterator in Java.

Iterator is used for iterating (looping) various collection classes such as **Hash Map**, **Array List**, **Linked List** etc  
Iterator Pattern is used "to access the elements of an aggregate object sequentially without exposing its underlying implementation".

3. Characteristics of Array List

* Array List inherits Abstract List class and implements List interface.
* Array List is initialized by a size; however the size can increase if collection grows or shrunk if objects are removed from the collection.
* Java Array List allows us to randomly access the list.
* Array List cannot be used for primitive types, like int, char, etc. We need a wrapper class for such cases (see [this](http://www.geeksforgeeks.org/array-vs-arraylist-in-java/) for details).
* Array List in Java can be seen as similar to [vector in C++](http://quiz.geeksforgeeks.org/vector-sequence-containers-the-c-standard-template-library-stl-set-1/).
* The constant factor is low compared to that for the Linked List implementation.

4. Advantages of using Array List

1. Array List is dynamically grows list if internal array out of space then its size increases by double.

2. Previous to Java version 5 storing primitives type was not allowed only object storage was allowed in Array List. After introducing Auto-boxing with Java 5 it allows us to store primitive type which get converted to Objects.

3. Array List generics are allowed so that one that one can find what kind of objects Array List going to store.

4. To get the length you will have to call size method.

5. There is an overhead related to managing the size of the internal array and more if you try to access its element using casting object.

6. Array List is dynamically allocated elements.

5. Applications of Array List In Real Life

A restaurant menu!   
Array (   
Lunch Menu (Spinach Artichoke Dip, Shrimp Fondue, Asian Dumplings, Southwestern Spring Rolls, Key West Coconut Shrimp, Fire Wings),   
Dinner Menu (Shrimp Fondue, Baja Chicken Quesadilla, Fried Mozzarella, Four Way Sampler)   
)

When bottles or juice packs are manufactured they bear a fixed number and are ordered serially... and kept in a crate

6. What is a linked list?

Linked List is an implementation of List interface. Java Linked List class uses doubly linked list to store the elements. It provides a linked-list data structure. It inherits the Abstract List class and implements List and Deque interfaces.

Operations that index into the list will traverse the list from the beginning or the end, whichever is closer to the specified index.

The major benefit of linked lists is that you do not specify a fixed size for your list. The more elements you add to the chain, the bigger the chain gets.

7. Characteristics of linked list

* Java Linked List class can contain duplicate elements.
* Java Linked List class maintains insertion order.
* Java Linked List class is non synchronized.
* In Java Linked List class, manipulation is fast because no shifting needs to be occurred.
* Java Linked List class can be used as list, stack or queue.

8. Advantages of using Linked List

* Linked lists are a dynamic data structure, which can grow and be pruned, allocating and de allocating memory while the program is running.
* Insertion and deletion node operations are easily implemented in a linked list.
* Dynamic data structures such as stacks and queues can be implemented using a linked list.
* **Efficient Memory Utilization** ,i.e. no need to pre-allocate memory
* Faster Access time, can be expanded in **constant time without memory overhead**
* Linear Data Structures such as Stack, Queue can be **easily implemented** using Linked list

9. Applications of Linked List

Consider the history section of web browsers, where it creates a linked list of web-pages visited, so that when you check history (traversal of a list) or press back button, the previous node's data is fetched

Another real life example could a be queue/line of persons standing for food in mess, insertion is done at one end and deletion at other

A simple real life example is a Train, here each coach is connected to its previous and next coach (Except first and last). In terms of programming consider coach body as node value and connectors as links to previous and next nodes.

10. Difference between Linked list and Array list

Array List internally uses **dynamic array** to store the elements.

Linked List internally uses **doubly linked list** to store the elements.

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

Manipulation with Linked List is **faster** than Array List because it uses doubly linked list so no bit shifting is required in memory.

Array List is **better for storing and accessing** data.

Linked List is **better for manipulating** data.

Array List search operation is pretty fast compared to the Linked List search operation.