1.) What is the underlying data structure of all the collections?

Ans: ArrayList- array of Object class.

LinkedList- l inear data structure.

HashMap- array and LinkedList.

HashSet- HashTable.

TreeSet- TreeMap.

Stack- Last In First Out (LIFO).

Queue- First In First Out (FIFO).

2.) Advantages and Disadvantages of ArrayList?

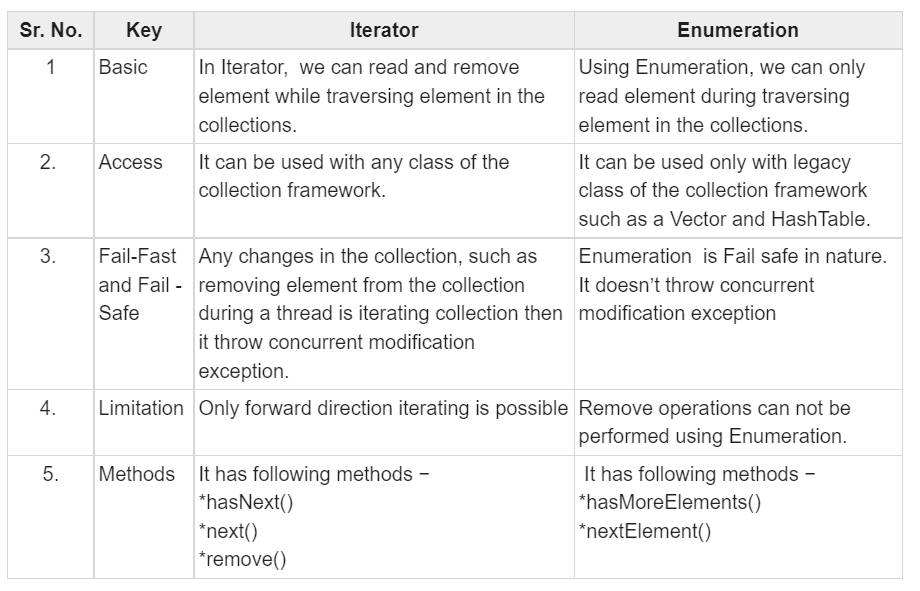
**Advantages:**

* ArrayList is variable length.
* Add any type of data into ArrayList.
* Traverse in both directions.
* ArrayList allows multiple null values.
* ArrayList allows to add duplicate elements.
* When ArrayList exceeds its capacity, then its size increases by 50%.
* Retrieval is faster in ArrayList

**Disadvantages**

* Never used Arraylist when you have to do lot of insertion and deletion as this will force arraylist to adjust the elements present in the list instead used linkedlist.
* When you want to add elements in the last or first and the arraylist size is 0, in this case you can use linkedlist to add specifically on top and bottom of list.
* You can't maintain uniqueness.
* Arraylist dont work on primitive values.

3.) Difference between Iterator and Enumeration?



4.) What are the 10 security breaches in OWASP?

1. **Broken Access Control**

Access control enforces policy such that users cannot act outside of their intended permissions.

### 2. Cryptographic Failures

Many web applications and APIs do not properly protect sensitive data with strong encryption. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes.

### 3. Injection

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query.

### 4. Insecure Design

Pre-coding activities are critical for the design of secure software. The design phase of you development lifecycle should gather security requirements and model threats, and development time should be budgeted to allow for these requirements to be met.

### 5. Security Misconfiguration

Your software is only as secure as you configure it to be. Using ad hoc configuration standards can lead to default accounts being left in place, open cloud storage, misconfigured HTTP headers

### 6. Vulnerable and Outdated Components

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application.

### 7. Identification and Authentication Failures

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys

### 8. Software and Data Integrity Failures

Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations.

### 9. Security Logging and Monitoring Failures

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

### 10. Server-Side Request Forgery

Server-Side Request Forgery (SSRF) flaws occur whenever a web application fetches a remote resource without validating the user-supplied URL.

5.) What is the difference between Comparable and Comparator?

|  |  |
| --- | --- |
| Comparable | Comparator |
| Comparable provides a **single sorting sequence**. In other words, we can sort the collection on the basis of a single element such as id, name, and price. | The Comparator provides **multiple sorting sequences**. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc |
| Comparable **affects the original class**, i.e., the actual class is modified. | Comparator **doesn't affect the original class**, i.e., the actual class is not modified. |
| Comparable provides **compareTo() method** to sort elements. | Comparator provides **compare() method** to sort elements. |
| Comparable is present in **java.lang** package | A Comparator is present in the **java.util** package. |
| We can sort the list elements of Comparable type by **Collections.sort(List)** method | We can sort the list elements of Comparator type by **Collections.sort(List, Comparator)** method. |