

Q. 1) What is the Collection Framework?

- Java Collection Framework (JCF) is a set of classes and interfaces used to store and manipulate groups of objects.
- It provides a standard architecture for managing data.
- Introduced in Java 1.2 (in `java.util` package).
- It works only with objects (primitives must use wrapper classes).
- Supports dynamic memory allocation (unlike arrays).
- Uses Generics for type safety.
- Provides built-in algorithms like sorting and searching.
- Improves performance and reusability of code.
- Provides a common way to access data using Iterator.

Q. 2) What are the interfaces of the Collection Framework?

1 List

- Ordered collection.
- Allows duplicate elements.
- Elements are accessed by index.
- Example implementations: `ArrayList`, `LinkedList`, `Vector`, `Stack`.

2 set

- Does NOT allow duplicate elements.
- No indexing.
- Example implementations: `HashSet`, `LinkedHashSet`, `TreeSet`.

3 Queue

- Follows FIFO (First In First Out).
- Used for processing elements before removal.
- Example implementations: `PriorityQueue`, `LinkedList`.

Q.3) What is the difference between Collection and Collections?

Collection

- It is an interface.
- Belongs to `java.util` package.
- It is the root interface of the collection framework (except `Map`).
- Used to represent a group of objects.

Collections

- It is a utility class.
- Belongs to `java.util` package.
- Provides static methods to operate on collection objects.

Q. 4) Difference between List, Set, and Map.

List

- Stores elements in ordered sequence
- Allows duplicate elements
- Maintains insertion order
- Allows index-based access

Set

- Stores unique elements only
- Does NOT allow duplicates
- No index-based access

Map

- Stores data in key-value pairs
- Keys must be unique
- Values can be duplicate
- No indexing

Q .5) What is ArrayList?

- Allowed duplicates
- Insertion order preserve.
- By default size is 10.
- Size increases 1.5 times.
- Datastructure growable.
- Have three constructor :-
 - default constructor
 - Initial capacity constructors
 - `Collection(c)`.

Q. 6) Write a program to count occurrences of an element in an array.

```
public class CountOccur {  
  
    //calculating occurrence of 10 in an array  
    public static void main(String[] args){  
  
        int[] a={1,2,3,4,5,6,7,8,9,10};  
        int key=10;  
        int count=0;  
        for (int i=0;i<a.length;i++){  
  
            if (a[i]==key){  
                count++;  
            }  
        }  
        System.out.println(count);  
    }  
}
```

Output:- count is :1

Q. 7) Write a program to separate even and odd numbers from an array.

```
import java.util.Scanner;  
  
public class SeparateEvOd {  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter size of array: ");  
        int n = sc.nextInt();  
  
        int[] arr = new int[n];  
  
        System.out.println("Enter array elements:");  
        for (int i = 0; i < n; i++) {  
            arr[i] = sc.nextInt();  
        }  
  
        System.out.println("Even numbers:");  
        for (int i = 0; i < n; i++) {  
            if (arr[i] % 2 == 0) {  
                System.out.print(arr[i] + " ");  
            }  
        }  
  
        System.out.println("\nOdd numbers:");  
        for (int i = 0; i < n; i++) {
```

```
        if (arr[i] % 2 != 0) {  
            System.out.print(arr[i] + " ");  
        }  
    }  
}
```

Output:-

Enter size of array: 6

Enter array elements:

1

2

3

3

3

2

Even numbers:

2 2

Odd numbers:

1 3 3 3

Q. 8)