

Java Programming Assignment-11

Part A-Theory Questions

1)What is the Collection Framework.

Ans:The Collection Framework in Java is a set of classes and interfaces used to store, manage, and manipulate groups of objects.

What does Collection Framework provide.

It provides:

Interfaces – rules (like List, Set, Map)

Classes – ready-made implementations (like ArrayList, HashSet, HashMap)

Methods – add, remove, search, sort, etc.

1. List

Allows duplicate values

Maintains insertion order

Example: ArrayList, LinkedList

2. Set

No duplicate values

No guaranteed order (mostly)

Example: HashSet, TreeSet

3. Map

Stores data in key–value pairs

Keys are unique

Example: HashMap, TreeMap

Advantages of Collection Framework

Dynamic size .

Ready-made methods (add, remove, search)

Reduces coding effort .

Improves performance .

2)What are the interfaces of the Collection Framework.

Ans:Main interfaces of the Java Collection Framework.

1)Iterator

Root interface.

Allows looping using for-each loop.

2)Collection

Parent interface of most collections.

Defines common methods like add(), remove(), size().

3>List

Allows duplicate values.

Maintains insertion order.

Implementations: ArrayList, LinkedList, Vector.

4)Set

No duplicates.

Order depends on implementation.

Implementations: HashSet, LinkedHashSet, TreeSet.

5)Queue

Used for FIFO operations.

Special methods like offer(), poll(), peek().

Implementations: PriorityQueue, ArrayDeque.

3)What are difference between Collection and Collection.

Ans:

Collection (interface)

It is an interface.

Part of java.util.

Collections (class)

It is a utility class.

Part of java.util.

Used to store a group of objects.	Contains static methods.
Parent interface of List, Set, Queue.	Used to operate on Collection object
Defines methods like:	Common methods:
add().	sort().
remove().	reverse().
size().	shuffle().
iterator().	max(),min().

4)Difference between List,Set and Map.

Ans.

Difference between List, Set, and Map (For Exam)

List

Allows duplicate elements.

Maintains insertion order.

Elements are accessed using index.

Examples: ArrayList, LinkedList.

Set.

Does not allow duplicate elements.

Order depends on implementation.

No index-based access.

Examples: HashSet, LinkedHashSet, TreeSet.

Map

Stores elements in key–value pairs.

Keys are unique, values can be duplicate.

Accessed using key.

Does not extend Collection.

Examples: HashMap, TreeMap, LinkedHashMap.

5)What is ArrayList.

Ans:ArrayList is a resizable array implementation of the List interface in Java.

ArrayList are:

Part of java.util package.

Implements List interface.

Allows duplicate elements.

Maintains insertion order.

Can store multiple data types (using generics).

Not synchronized (not thread-safe by default).

ArrayList Features:

Size grows and shrinks dynamically.

Faster random access (because index-based).

Slower in insertion/deletion in middle.

Part B-Practical Questions

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

Ans:

```
import java.util.Scanner;  
  
public class CountOccurrence{  
  
    public static void main(String[] args)  
    {  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter size:");  
        int n=sc.nextInt();  
        int count=0;  
        int[] a=new int[n];
```

```

System.out.println("Enter array :");

for(int i=0;i<n;i++)
{
    a[i]=sc.nextInt();
}

System.out.println("Enter number for count occurrence:");

int data=sc.nextInt();

for(int i=0;i<n;i++)
{
    if(a[i]==data)
        count++;
}

System.out.println("Count is:"+count);
}
}

```

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

Ans:

```

import.java.util.Scanner;

public class EvenOddArray{

    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter size:");

        int n=sc.nextInt();

        int[] a=new int[n];

        System.out.println("Enter array :");

```

```

for(int i=0;i<n;i++)
{
    a[i]=sc.nextInt();
}

System.out.println("Even numbers:");
for(int i=0;i<n;i++)
{
    if(a[i]%2==0)
        System.out.println(a[i]);
}

System.out.println("Odd numbers:");
for(int i=0;i<n;i++)
{
    if(a[i]%2!=0)
        System.out.println(a[i]);
}
}

```

3)Write a program to find intersection of two arrays.

Ans:

```

import java.util.Scanner;

public class IntersectionArray {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter size of first array: ");
        int n1 = sc.nextInt();

        int n2 = sc.nextInt();

```

```

int[] a = new int[n1];

System.out.println("Enter elements of first array:");

for (int i = 0; i < n1; i++) {

    a[i] = sc.nextInt();

}

System.out.print("Enter size of second array: ");

int n2 = sc.nextInt();

int[] b = new int[n2];

System.out.println("Enter elements of second array:");

for (int i = 0; i < n2; i++) {

    b[i] = sc.nextInt();

}

System.out.println("Intersection of two arrays:");

for (int i = 0; i < n1; i++) {

    for (int j = 0; j < n2; j++) {

        if (a[i] == b[j]) {

            System.out.print(a[i] + " ");

        }

    }

}

}

```

4)Write a program to find union of two arrays.

Ans:

```

import java.util.Scanner;

public class UnionArray {

```

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter size of first array: ");  
    int n1 = sc.nextInt();  
    int[] a = new int[n1];  
    System.out.println("Enter elements of first array:");  
    for (int i = 0; i < n1; i++) {  
        a[i] = sc.nextInt();  
    }  
    System.out.print("Enter size of second array: ");  
    int n2 = sc.nextInt();  
    int[] b = new int[n2];  
    System.out.println("Enter elements of second array:");  
    for (int i = 0; i < n2; i++) {  
        b[i] = sc.nextInt();  
    }  
    System.out.println("Union of two arrays:");  
    for (int i = 0; i < n1; i++) {  
        System.out.print(a[i] + " ");  
    }  
    for (int i = 0; i < n2; i++) {  
        boolean found = false;  
        for (int j = 0; j < n1; j++) {  
            if (b[i] == a[j]) {  
                found = true;  
                break;  
            }  
        }  
        if (found) {  
            System.out.print(b[i] + " ");  
        }  
    }  
}
```

```
    }
}

if (!found) {
    System.out.print(b[i] + " ");
}
}

}

}
```

5) Write a program to check if two arrays are equal.

Ans:

```
class ArrayEqual {

    public static void main(String[] args) {

        int[] a = {1, 2, 3};

        int[] b = {1, 2, 3};

        boolean same = true;

        for (int i = 0; i < a.length; i++) {

            if (a[i] != b[i]) {

                same = false;

                break;
            }
        }

        if (same)

            System.out.println("Arrays are equal");

        else

            System.out.println("Arrays are not equal");
    }
}
```

}