Assignment 2

Theory:

In Java, the List interface is an ordered collection that allows us to store and access elements sequentially.Lists may contain duplicate elements It extends the Collectioninterface.

Classes that Implement List:

● Since List is an interface, we cannot create objects from it.

● In order to use functionalities of the List interface, we can use these classes

● It is used to store a collection of elements where duplicate elements are allowed.Java List

● Positional access — manipulates elements based on their numerical position in the list. This includes methods such as get, set, add, addAll,and remove.

● Search — searches for a specified object in the list and returns its numerical position. Search methods include indexOf and lastIndexOf.

● Iteration — extends Iterator semantics to take advantage of the list's sequential nature. The listIterator methods provide this behavior.

● Range-view — The sublist method performs arbitrary range operations on the list.In Java, we must **import java.util.List**package in order to use List.

// ArrayList implementation of List List<String> list1 = new ArrayList<>();

// LinkedList implementation of List List<String> list2 = new LinkedList<>();

Here, we have created objects list1 and list2 of classes ArrayList and LinkedList.These objects can use the functionalities of the List interface.

Java List

● add() - adds an element to a list

- addAll() - adds all elements of one list to another
- get() - helps to randomly access elements from lists
- iterator() - returns iterator object that can be used to sequentially access elements of lists
- set() - changes elements of lists
- remove() - removes an element from the list remove(“Welcome”)
- removeAll() - removes all the elements from the list
- clear() - removes all the elements from the list (more efficient than removeAll())
- size() - returns the length of lists
- toArray() - converts a list into an array
- contains() - returns true if a list contains specified element

Question 1 : Write a Java program to create a List containing a listof items of type String and use for –each loop to print the items of the list.

Code:

// Write a Java program to create a List containing a list of items of type String and use for -each loop to print the itemsof the list

```
import java.util.*;
public class aditya
{

public static void main(String[] args)
{

List<String>list = new ArrayList<>();

list.add("10th");
list.add("12th");
list.add("bsc.cs");
list.add("MCA");
list.add("Job");

for(String newStr :list)
{
System.out.println(newStr+"");
}
}
}
```

Output:

```
PS C:\Users\MICROSOFT\Desktop\java1>  c:; cd 'c:\Users\MICROSOFT\Desktop\java1'; & 'C:\Program Files\Java\jdk-19\bin\java.exe'
'--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MICROSOFT\AppData\Roaming\Code\User\workspaceStora
ge\731cc6bbbf82f43f7c15da4dce972989\redhat.java\jdt_ws\java1_75f7981c\bin' 'aditya'
10th
12th
bsc.cs
MCA
Job
PS C:\Users\MICROSOFT\Desktop\java1>
```

Question 2 : Write a Java program to create a List containing a listof items and use List iterator interface to print items present in the list. Also print the list in reverse/backward direction.

Code:

```
import java.util.ArrayList; import java.util.List;
import java.util.ListIterator;


public abstract class aditya {


public static void main(String[] args) { List<Integer> l1 = new ArrayList<>(); l1.add(11);
l1.add(22);
l1.add(33);
ListIterator<Integer> L = l1.listIterator(); System.out.println("Traversing in Forward direction");
while(L.hasNext()) {
System.out.println(L.next());


}
System.out.println("Traversing in reverse direction"); while(L.hasPrevious()) {

System.out.println(L.previous());
}
}


}
```

Output:

```
PS C:\Users\MICROSOFT\Desktop\java1>  c:; cd 'c:\Users\MICROSOFT\Desktop\java1'; & 'C:\Program Files\Java\jdk-19\bin\java.exe'
'--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MICROSOFT\AppData\Roaming\Code\User\workspaceStora
ge\731cc6bbbf82f43f7c15da4dce972989\redhat.java\jdt_ws\java1_75f7981c\bin' 'aditya'
Traversing in Forward direction
11
22
33
Traversing in reverse direction
33
22
11
PS C:\Users\MICROSOFT\Desktop\java1>
```