List interface

- The List interface provides a way to store the ordered collection.
- It is a child interface of Collection.
- It is an ordered collection of objects in which duplicate values can be stored. List preserves the insertion order, it allows positional access and insertion of elements.

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

```
import java.util.*;
public class ArrayListExam1 {

    public static void main(String[] args) {
        int a;
        ArrayList<String> list=new ArrayList<String>();//Creating arraylist

         list.add("red");//Adding object in arraylist
         list.add("green");
         list.add("yellow");
         list.add("orange");

         System.out.println(list);

         //Traversing list through for-each loop
         System.out.println("Traversing list through for each loop ");
         for(String color:list)
           System.out.println(color);


        //Traversing list through Iterator

         System.out.println("Traversing list through Iterator ");
         Iterator itr=list.iterator();//getting the Iterator
         while(itr.hasNext()){//check if iterator has the elements
```

```
                 System.out.println(itr.next());//printing the element and move to
next
                }

         }

}
```

Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse/ backword direction.

```
package reversarraylist;
import java.util.*;
public class Reverse {

         public static void main(String[] args) {
                 // Let us create a list of strings
      List<String>  mylist = new ArrayList<String>();
      mylist.add("sidhi");
      mylist.add("vidhi");
      mylist.add("ridhi");
      mylist.add("nidhi");

      System.out.println("Original list  ");
                Iterator itr=mylist.iterator();//getting the Iterator
                while(itr.hasNext()){//check if iterator has the elements
                 System.out.println(itr.next());//printing the element and move to
next
                }
                 Collections.reverse(mylist);



          System.out.println("reversed list ");
                        Iterator itr1=mylist.iterator();//getting the Iterator
                        while(itr1.hasNext()){//check if iterator has the elements
```

```
                  System.out.println(itr1.next());//printing the element and move to next
      }
}

      }
```

Write a Java program to create List containing list of items of an Employee class and use ListIterator interface to print items present in the list.

```
package arraylist;

import java.util.*;

class Employee{
  public int id;
  public String name;
  public String city;

  public Employee(){}

  //Parameterized Constructor
  public Employee(int id, String name,String city)
  {

    this.id = id;
    this.name = name;
    this.city=city;

  }

  public int getId() {
  return id;
  }

  public String getName() {
  return name;
```

```
   }

   public String getAddress() {
   return city;
   }
}

public class DisplayArrayList {

        public static void main(String[] args) {

                Employee e1=new Employee(23, "Nidhi","Vashi");

                System.out.print("ID, Name and City of the employee are : ");
      System.out.println(e1.getId()+" "+e1.getName()+" " +e1.getAddress());

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

            list.add(new Employee(1, "Nidhi","Vashi"));
            list.add(new Employee(2, "Ridhi","Nerul"));
            list.add(new Employee(3, "Sidhi","Belapur"));
            list.add(new Employee(4, "Vidhi","Thane"));

              for (Employee s : list) //Iterates as long as there are elements in the
list. An object s is created of type Employee class.
              {
                 System.out.print("ID, Name and City of the employee are : ");
                 System.out.println(s.getId()+" "+s.getName()+" " +s.getAddress());
              }

        }

}
```