**Module 2**

1. **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/ backward direction**

# Code:-

**package** mod2;

**import** java.util.ArrayList;

**import** java.util.List;b

**import** java.util.ListIterator;

**public class** List\_Iterator\_2 {

**public static void** main(String[] args) {

// **TODO** Auto-generated method stub List<Integer> list=**new** ArrayList<Integer>(); list.add(1);

list.add(2);

list.add(9);

list.add(4);

System.***out***.println("add"); ListIterator<Integer> listIterator =

list.listIterator();

System.***out***.println("Element in Forword direction");

**while**(listIterator.hasNext()) {

System.***out***.println(listIterator.next());

}

System.***out***.println("Element is Backword Direction");

**while**(listIterator.hasPrevious()) {

System.***out***.println(listIterator.previous());

}

}

}

**Output:-**

1. **Write a Java program to create a Set containing list of items of type String and print the items in the list using the Iterator interface. Also print the list in reverse/ backward direction**

# Code:-

**package** module\_2;

**import** java.util.ArrayList; **import** java.util.Collections; **import** java.util.Iterator; **import** java.util.Set;

**import** java.util.TreeSet;

**public class** List {

**public static void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println("lkj"); Set<String> langSet = **new** TreeSet<String>(); langSet.add("C++");

langSet.add("Java");

Iterator<String> iterator = langSet.iterator(); System.***out***.println("Forward Direction:"); **while** (iterator.hasNext()) {

System.***out***.println(iterator.next());

}

ArrayList<String> list = **new** ArrayList<String>(langSet); Collections.*reverse*(list);

System.***out***.println("Reverse Direction:");

**for** (String lang : list) { System.***out***.println(lang);

}

}

}

**Output:-**

1. **Write a Java program using Set interface containing list of items and perform the following operations:**
   1. **Add items in the set.**
   2. **Insert items of one set into another set.**
   3. **Remove items from the set**
   4. **Search the specified item in the set**

**Code**:-

package module\_2; import java.util.TreeSet; public class Set {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("lkj"); TreeSet<String> subjectSet = new TreeSet<String>(); subjectSet.add("DBMS");

subjectSet.add("HTML"); subjectSet.add("Java");

TreeSet<String> subjectSet2 = new TreeSet<String>(); subjectSet2.add("Java");

subjectSet2.add("Python"); subjectSet2.add("C++"); subjectSet.addAll(subjectSet2); System.out.println("Combined Set:");

for (String e : subjectSet) { System.out.println(e);

}

String removeSubject = "Python";

boolean result = subjectSet.contains(removeSubject); if (result) {

subjectSet.remove(removeSubject);

}

System.out.println("After Removing Python:"); for (String e : subjectSet) {

System.out.println(e);

}

}

}

**Output**:-

# Code:-

package module\_2; import java.util.TreeSet;

public class Set {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("lj"); TreeSet<String> subjectSet = new TreeSet<String>(); subjectSet.add("DBMS");

subjectSet.add("HTML"); subjectSet.add("Java");

TreeSet<String> subjectSet2 = new TreeSet<String>(); subjectSet2.add("Java");

subjectSet2.add("Python"); subjectSet2.add("C++"); subjectSet.addAll(subjectSet2); System.out.println("Combined Set:"); for (String e : subjectSet) {

System.out.println(e);

}

String removeSubject = "Python";

boolean result = subjectSet.contains(removeSubject); if (result) {

subjectSet.remove(removeSubject);

}

System.out.println("After Removing Python:"); for (String e : subjectSet) {

System.out.println(e);

}

}

}

**Output**:-

1. **Write a Java program using Map interface containing list of items having keys and associated values and perform the following operations:**
   1. **Add items in the map.**
   2. **Remove items from the map**
   3. **Search specific key from the map**
   4. **Get value of the specified key**
   5. **Insert map elements of one map into another map.**
   6. **Print all keys and values of the map.**

# Code:

**package** mod2;

**import** java.util.HashMap; **import** java.util.Map; **public class** Map\_5 {

**public static void** main(String[] args) {

// **TODO** Auto-generated method stub

Map<Integer,String> map=**new** HashMap<Integer,String>(); map.put(1,"ABC");

map.put(2,"XYZ");

map.put(3,"UVW");

map.forEach((K,v)-> System.***out***.println("Key:"+K+",Value:"+v));

**if**(map.containsKey(2)) {

map.remove(2);

}

System.***out***.println("Map after removing Key 2:"); map.forEach((K,v)->System.***out***.println("Key:"+K+",Value:"+v));

String value=map.get(1); System.***out***.println("Value at Key1:"+value);

Map<Integer,String>map1=**new** HashMap<Integer,String>(); map1.put(3,"OUV");

map1.put(4, "PQR");

map.putAll(map1);

System.***out***.println("Map after adding another Map"); map.forEach((K,v)->System.***out***.println("Key:"+K+",Value:"+v));

}

}

# Output:-

1. **WAP using Lambda Expression with multiple parameters to print addition of two numbers.**

Code:-

**package** mod2;

**interface** Arithmetic{

**public int** add(**int** a,**int** b);

}

**public class** Lambda\_7 {

**public static void** main(String[] args) { System.***out***.println(""); Arithmetic arth=(a,b)-> {**return**(a+b); };

**int** sum=arth.add(10,20); System.***out***.println("Addition of Two numbers:"+sum);

}

}

Output:-

1. **WAP using Lambda Expression to calculate following:**
2. **Convert Fahrenheit to Celsius**
3. **Convert Kilometers to Meters**

# Code:

**package** mod2;

**interface** TemperatureInterface{

**public void** ftoc(**double** farenheit);

}

**interface** DistanceInterface{

**public void** ktom(**double** km);

}

**public class** Lambda\_8 {

**public static void** main(String[] args) { System.***out***.println("ljlj"); TemperatureInterface ti=(farenheit)->{

System.***out***.println((farenheit-32.0)\*(5.0/9.0)+": Degree Celcius");

};

DistanceInterface di= (kilometer) ->{ System.***out***.println(kilometer\*1000+" meters");

};

ti.ftoc(97.6);

di.ktom(4.0);

}

}

**Output:**