2.8 Collections and Generics

This section will guide you to:

- Create a Java project in your IDE
- Write a program in Java to create collections

This lab has three subsections, namely:

2.8.1 Writing a program in Java to verify implementations of collections
2.8.2 Executing the program and verifying it is working
2.8.3 Pushing the code to your GitHub repositories

Step 2.8.1: Writing a program in Java to verify implementations of collections

There are two ways you can perform this step; you can create a new Java project, or you can create a new Java class in the existing project. It is preferable to create a new Java class in the existing project but feel free to explore the first option. The steps mentioned below will work once you create a project in Java.

- *Open Eclipse*
- *[Right click]* on the **src** folder of the project
- Select *New -> Java Class* -> Enter the filename (follow camelCasing)
- Execute the code below resolving the warning and errors due compatibility-related issues

```
import java.util.*;

public class collectionAssisted {

	public static void main(String[] args) {
		//creating arraylist
		System.out.println("ArrayList");
		ArrayList<String> city=new ArrayList<String>();
		city.add("Bangalore");//
		city.add("Delhi");
		System.out.println(city);

		//creating vector
		System.out.println("\n");
		System.out.println("Vector");
		Vector<Integer> vec = new Vector();
		vec.addElement(15);
		vec.addElement(30);
		System.out.println(vec);

		//creating linkedlist
```

```
        System.out.println("\n");
        System.out.println("LinkedList");
        LinkedList<String> names=new LinkedList<String>();
        names.add("Alex");
        names.add("John");
        Iterator<String> itr=names.iterator();
        while(itr.hasNext()){
         System.out.println(itr.next());

         //creating hashset
         System.out.println("\n");
         System.out.println("HashSet");
         HashSet<Integer> set=new HashSet<Integer>();
         set.add(101);
         set.add(103);
         set.add(102);
         set.add(104);
         System.out.println(set);

         //creating linkedhashset
         System.out.println("\n");
         System.out.println("LinkedHashSet");
         LinkedHashSet<Integer> set2=new LinkedHashSet<Integer>();
         set2.add(11);
         set2.add(13);
         set2.add(12);
         set2.add(14);
         System.out.println(set2);
        }
        }
}
```

Step 2.8.2: Executing the program and verifying whether it is working

Before you execute the program, check for syntactical corrections. If no errors are found, follow the steps mentioned below:

- ***[Right click]*** in the program space
- Select *Run As Java Application*

```
ArrayList
[Bangalore, Delhi]


Vector
[15, 30]


LinkedList
Alex


HashSet
[101, 102, 103, 104]


LinkedHashSet
[11, 13, 12, 14]
John


HashSet
[101, 102, 103, 104]


LinkedHashSet
[11, 13, 12, 14]
```

Step 2.8.3: Pushing the codes to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd java_program

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master