5.Writing a program in Java to verify implementations of collections

ALGORITHM

Step 1: Start

Step 2: Create array list collection

Step 3: Add items using add( )

Step 4: Create linked list collection

Step 5: Add items using add()

Step 6: Create stack collection

Step 7: Add items using push()

Step 8: Stop

SOURCE CODE

```
//implementation of collections

package assistedPracticeProject;
import java.util.*;
public class Practice_Project5
{
	public static void main(String args[])
	{
		//collection - array list
		ArrayList<String> list=new ArrayList<String>();
		System.out.println("\n---PROGRAMMING LANGUAGES---\n");
		//adding objects in array list
		list.add("C");
		list.add("JAVA");
		list.add("C++");
		list.add("Python");
```

```
list.add("C#");

//iterate through the list
Iterator pp1=list.iterator();
while(pp1.hasNext())
{
        System.out.println(pp1.next()); //displaying each list item
}

//collection - linked list
LinkedList<String> ll=new LinkedList<String>();
System.out.println("\n---OPERATING SYSTEM---\n");
ll.add("Linux");
ll.add("Windows");
ll.add("IOS");
ll.add("Android");
Iterator<String> pp2=ll.iterator();
while(pp2.hasNext())
{
        System.out.println(pp2.next());
}

//iterate through stack

Stack<String> stack = new Stack<String>();
System.out.println("\n---INPUT DEVICES---\n");
stack.push("Keyboard");
stack.push("Mouse");
stack.push("Joystick");
```

```
                stack.push("Web Camera");
                stack.push("Scanner");
                Iterator<String> pp3=stack.iterator();
                while(pp3.hasNext())
                {
                        System.out.println(pp3.next());
                }
        }
}
```

OUTPUT