PROGRAMMING USING JAVA
WEEK 11 ASSIGNMENT

1. **How do you create an instance of Vector? How do you add or insert a new element into a vector? How do you remove an element from a vector? How do you find the size of a vector?**

Program :

```
import java.util.*;
class VectorDemo {
public static void main(String args[]) {
// initial size is 3, increment is 2
Vector<Integer> v = new Vector<Integer>(3, 2);
System.out.println("Initial size: " + v.size());
System.out.println("Initial capacity: " + v.capacity());
v.addElement(1);
v.addElement(2);
v.addElement(3);
v.addElement(4);
System.out.println("Capacity after four additions: " + v.capacity());
v.addElement(5);
System.out.println("Current capacity: " + v.capacity());
v.addElement(6);
v.addElement(7);
System.out.println("Current capacity: " + v.capacity());
v.addElement(9);
v.addElement(10);
System.out.println("Current capacity: " + v.capacity());
v.addElement(11);
v.addElement(12);
System.out.println("First element: " + v.firstElement());
System.out.println("Last element: " + v.lastElement());
if(v.contains(3))
System.out.println("Vector contains 3.");
// Enumerate the elements in the vector.
Enumeration<Integer> vEnum = v.elements();
System.out.println("\nElements in vector:");
while(vEnum.hasMoreElements())
System.out.print(vEnum.nextElement() + " ");
System.out.println();
}
```

Output

```
Initial size: 0
```

Initial capacity: 3
Capacity after four additions: 5
Current capacity: 5
Current capacity: 7
Current capacity: 9

First element: 1
Last element: 12 Vector
contains 3. Elements in
vector:
1 2 3 4 5 6 7 9 10 11 12

2. **How do you create an instance of Stack? How do you add a new element to a stack? How do you remove an element from a stack? How do you find the size of a stack?**

Program :

```
import java.util.*;
import java.io.*;

public class StackDemo {
	public static void main(String args[])
	{
		// Creating an empty Stack
		Stack<Integer> stack = new Stack<Integer>();

		// Use add() method to add elements
		stack.push(10);
		stack.push(15);
		stack.push(30);
		stack.push(20);
		stack.push(5);

		// Displaying the Stack System.out.println("Initial
		Stack: " + stack);

		// Removing elements using pop() method
		System.out.println("Popped element: "
						+ + stack.pop());
		System.out.println("Popped element: "
						+ + stack.pop());

		// Displaying the Stack after pop operation
		System.out.println("Stack after pop operation "
						+ + stack);
	}
```

} **Output :**

```
Initial Stack: [10, 15, 30, 20, 5]
Popped element: 5
Popped element: 20
Stack after pop operation [10, 15, 30]
```