ASSIGNMENT

OOP LAB

Submitted to

Sr.Elsin miss

Submitted by

Aksa Anna Jose

Roll no: 6,S2

8) **Program to create a generic stack and do the push and pop operations.**

```
import java.io.*;

class Stack1 {

static final int MAX = 1000;

int top;

int a[] = new int[MAX];

booleanisEmpty()

{

return (top < 0);

}

stack()

{

top = -1;

}

boolean push(int x)

{

if (top >= (MAX - 1)) {

System.out.println("Stack Overflow");

return false;

}

else {

a[++top] = x;

System.out.println(x + " pushed into stack");

return true;
}

}

int pop()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top--];

return x;

}

}

int peek()

{if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top];

return x;

}

}

}
```

```
classstackop{
public static void main(String args[])
{
stack s = new stack();
s.push(8);
s.push(18);
s.push(28);
System.out.println(s.pop() + " Popped from stack");
s.push(35);
s.push(45);
System.out.println(s.pop() + " Popped from stack");
}
}
```

Output

9) **Using generic method perform Bubble sort.**

Program

```
import java.io.*;
classbubblesort
{
static void bubbleSort(intarr[], int n)
{
```

```
inti, j, temp;

boolean swapped;

for (i = 0; i< n - 1; i++)

{

swapped = false;

for (j = 0; j < n - i - 1; j++)

{

if (arr[j] >arr[j + 1])

{

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

swapped = true;

} }

if (swapped == false)

break;

} }

static void printArray(intarr[], int size)

{

inti;

for (i = 0; i< size; i++)

System.out.print(arr[i] + " ");

System.out.println();

}
```

```
public static void main(String args[])

{

intarr[] = { 60, 90, 70, 10, 110, 50, 30, 150, 40, 20 };

int n = arr.length;

bubbleSort(arr, n);

System.out.println("Sorted array: ");

printArray(arr, n);

}
```

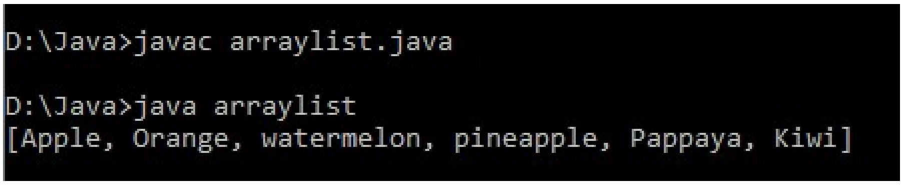
Output

10) **Maintain a list of string using Arraylist from collection framework,perform built-in.**

Program

```
importjava.util.*;

public class arraylist

{

public static void main(String args[])

{

ArrayList<String> list=new ArrayList<String>();

list.add("Apple");

list.add("Orange");

list.add("watermelon");

list.add("pineapple");
list.add("Pappaya");

list.add("Kiwi");

System.out.println(list);

}

}
```

Output

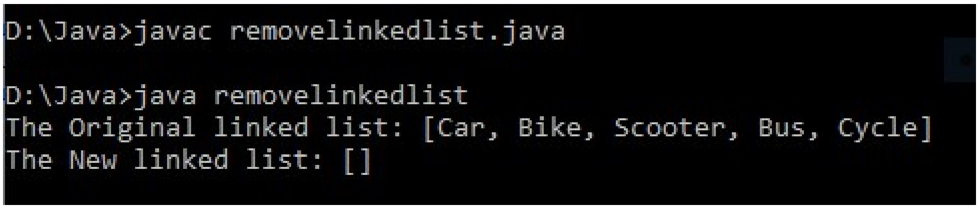


```
D:\Java>javac arraylist.java

D:\Java>java arraylist
[Apple, Orange, watermelon, pineapple, Pappaya, Kiwi]
```

11) **Program to remove all the elements from a linked list.**

```
importjava.util.*;
public class removelinkedlist
 {
public static void main(String args[])
{
LinkedList<String>l_list = new LinkedList<String>();
l_list.add("Car");
l_list.add("Bike");
l_list.add("Scooter");
l_list.add("Bus");
l_list.add("Cycle");
System.out.println("The Original linked list: " + l_list);
l_list.clear();
System.out.println("The New linked list: " + l_list);

}

}
```

Output

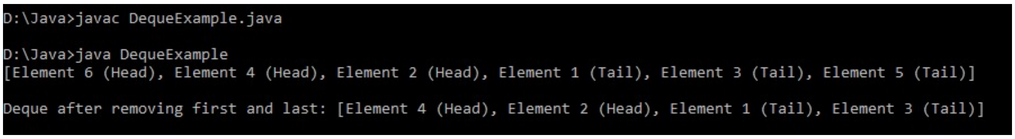


```
D:\Java>javac removelinkedlist.java

D:\Java>java removelinkedlist
The Original linked list: [Car, Bike, Scooter, Bus, Cycle]
The New linked list: []
```

14) **Program to demonstrate the addition and deletion of elements in dequeue.**

```
import java.util.*;

public class DequeExample {

        public static void main(String[] args)

        {

                Deque<String> deque

                        = new LinkedList<String>();

                deque.add("Element 1 (Tail)");



                deque.addFirst("Element 2 (Head)");



                deque.addLast("Element 3 (Tail)");



                deque.push("Element 4 (Head)");
                deque.offer("Element 5 (Tail)");



                deque.offerFirst("Element 6 (Head)");

                System.out.println(deque + "\n");



                deque.removeFirst();
                deque.removeLast();
                System.out.println("Deque after removing "
                                                + "first and last: "
                                                + deque);
        }
}
```

Output



```
D:\Java>javac DequeExample.java

D:\Java>java DequeExample
[Element 6 (Head), Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail), Element 5 (Tail)]

Deque after removing first and last: [Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail)]
```

17) **Program to demonstrate the working of map interface by adding, changing and removing.**

```
import java.util.*;

class HashMap

 {

          public static void main(String args[])

          {

                    Map<String, Integer> hm

                              = new HashMap<String, Integer>();


                    hm.put("a", new Integer(100));

                    hm.put("b", new Integer(200));

                    hm.put("c", new Integer(300));

                    hm.put("d", new Integer(400));


                    for (Map.Entry<String, Integer> me : hm.entrySet()) {

                              System.out.print(me.getKey() + ":");

                              System.out.println(me.getValue());

                    }

          }

}
```

Output