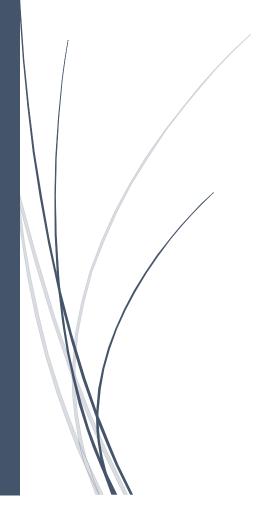
## 5/21/2018

# OOP Assignment 4 JAVA CODE



Alina Raza FA17-BCS-072

### QUESTION 1

Updating specific array element by given element and can search an element in an array list.

```
JAVA CODE:
import java.util.*;
 public class Test {
 public static void main(String[] args) {
// Creae a list and add some colors to the list
List<String> list Strings = new ArrayList<String>();
list_Strings.add("Red");
list_Strings.add("Green");
list_Strings.add("Orange");
list_Strings.add("White");
list Strings.add("Black");
// Print the list
System.out.println(list_Strings);
// Update the third element with "Yellow"
list_Strings.set(2, "Yellow");
 // Print the list again
 System.out.println(list_Strings); }}
Output:
[Red, Green, Orange, White, Black]
```

[Red, Green, Yellow, White, Black]

```
QUESTION 2 (A)
Sorting a given array list.
JAVA CODE:
import java.util.*;
public class ArrayListOfInteger {
       public static void main(String args[]){
         ArrayList<Integer> arraylist = new ArrayList<Integer>();
         arraylist.add(11);
         arraylist.add(2);
         arraylist.add(7);
         arraylist.add(3);
         /* ArrayList before the sorting*/
                                               System.out.println("Before Sorting:");
         for(int counter: arraylist){
                                                     System.out.println(counter);}
         /* Sorting of arraylist using Collections.sort*/
                                                              Collections.sort(arraylist);
System.out.println("After Sorting:");
for(int counter: arraylist){
System.out.println(counter);}}}
Output:
Before Sorting:
11
2
7
3
After Sorting:
2
3
7
```

11

```
QUESTION 2 (B)
```

Cloning an arraylist to another arraylist

```
JAVA CODE:
import java.util.ArrayList;
public class Details {
 public static void main(String a[]){
  ArrayList<String> al = new ArrayList<String>();
  //Adding elements to the ArrayList
  al.add("Apple");
  al.add("Orange");
  al.add("Mango");
  al.add("Grapes");
  System.out.println("ArrayList: "+al);
  ArrayList<String> al2 = (ArrayList<String>)al.clone();
  System.out.println("Shallow copy of ArrayList: "+ al2);
  //add and remove on original ArrayList
  al.add("Fig");
  al.remove("Orange");
  //Display of both ArrayLists after add & remove
  System.out.println("Original ArrayList:"+al);
  System.out.println("Cloned ArrayList:"+al2);
 }
}
Output:
```

ArrayList: [Apple, Orange, Mango, Grapes]

Shallow copy of ArrayList: [Apple, Orange, Mango, Grapes]

Original ArrayList:[Apple, Mango, Grapes, Fig]

Cloned ArrayList:[Apple, Orange, Mango, Grapes]

#### **QUESTION 3**

Retrieve but does not remove, the last element of a linked list

```
JAVA CODE:
import java.util.*;
public class Exercise21 {
public static void main(String[] args) {
// create an empty linked list
LinkedList <String> c1 = new LinkedList <String> ();
      c1.add("Red");
     c1.add("Green");
     c1.add("Black");
     c1.add("White");
     c1.add("Pink");
                           System.out.println("Original linked list: " + c1);
  // Retrieve but does not remove, the last element of a linked list
    String x = c1.peekLast();
  System.out.println("Last element in the list: " + x); System.out.println("Original linked list: " + c1); }}
Output:
Original linked list: [Red, Green, Black, White, Pink]
Last element in the list: Pink
Original linked list: [Red, Green, Black, White, Pink]
```

### **QUESTION 4**

Convert a priority queue to an array containing all of the elements of the queue

```
JAVA CODE:
import java.util.*;
 public class Example10 {
 public static void main(String[] args) {
 // Create Priority Queue
      PriorityQueue<String> pq1 = new PriorityQueue<String>();
 // use add() method to add values in the Priority Queue
     pq1.add("Red");
     pq1.add("Green");
     pq1.add("Black");
     pq1.add("White");
  System.out.println("Original Priority Queue: "+pq1);
 //Convert a linked list to array list
 List<String> array_list = new ArrayList<String>(pq1);
 System.out.println("Array containing all of the elements in the queue: "+array list);
 }
}
Output:
Original Priority Queue: [Black, Red, Green, White]
Array containing all of the elements in the queue: [Black, Red, Green, White]
```