ASSIGNMENT

≈013

Name : L. Jaithra

Reg.No : 192210259

Course Code : CSA0914

Course Name: Programming in Java using Raspberry pi

Submitted To:-

Dr. Hemavathi. R

1 Pseudocode:-

```
PROCEDURE Array list Operations
    DECLARE list AS Arraylist of string
      ADD apple, banana, Cherry, dates to list
      Print list
      SET removeindex to 2
        Remove element at remove index from list
        Print removedelement, list
      SET Searchelement to "data'
       if Search index is not -1 then
        print Search element, search index
       else
         Print 'notfound"
    FOR each element in list
      print element
    END FOR
END program.
```

Program:-

```
import java.util.Arraylist;
public class Arraylistoperations {
    Public static void main(String[] args) {
        Arraylist<String>list = new Arraylist<>();
          list.add("apple");
          list.add("banana");
          list.add ("cherry");
          list.add ("dates');
System.out.println("Initial list: "+ list);
int removeindex = 2;
String removelement = List remove (removeindex);
system.out.println ("remove element: "+ removedelement);
System.out.println("list after removal: "+ list);
String Searchelement = "date"
if (Searchindex ≠ -1){
  System.out.println ("element found at: "+ Searchindex;
}
else {
 System.out.println ("elmement not found');
}
System.out.println (" iterating through the list: ");
for (string element: list) {
  System.out.println(element);
}
}
}
```

Output-

```
Initial list, (apple, banana, Cherry, dates)
removed element: cherry
element 'dates' found at 3
```

2. Pseudocode:-

```
PROCEDURE Hashsetoperations
   Declare nameset as Hashset of strings
      Add john, Alice, bob to nameset
      print nameset
      SET newname to "David"
      ADD newname to nameset
      print newname + nameset
      SET removename to "Bob"
      remove removename from namset
      print removename + nameset
      SET Searchname to "Alice"
      If nameset contains searchname then
         Print "name is present in the set"
      else
         print "name is not present"
    END FOR
  END program.
```

Program:-

```
  import java.util.Hashset;
  public class Hashsetoperations{
     public static void main (String[] args){
       Hashset<string> nameset = new Hashset<>();
          nameset.add ("John");
          nameset.add ("Alice");
          nameset.add ("Bob");
      System.out.println ("Initial set:"+nameset);
       String newname = "David";
      nameset.add (neconame);
      System.out.println ("After adding set:"+nameset);
      string removename = "Bob";
      nameset.remove (removename)
      System.out.println ("After removing:"+ nameset
      string Search name = "Alice"
      if (nameset.Contains(searchname)) {
        System.out.println("name is found");
      }
      else {
        System.out.println("name is not present");
      }
      System.out.println("display all names:");
    }
  }
```

Output:-

```
Initialset: (Bob, John, Alice)
Set after adding david (Bob, John, Alice, David)
Set after removing Bob: [John, Alice, David]
Name Alice is present in the set.
```

3. Pseudocode:-

```
PROCEDURE priorityqueue Example
    Declare employee As class
        Declare name as string
        Declare priority as Integer.
        Constructor Employee (name as string, integer)
          Set this name, priority.
        end constructor
    end class
    Declare pq as priorityqueue of employee.
      Set pq to new priorityqueue (E1, E2) => E2.priority - E1.priority.
      Add John, 3; Alice, 1; Bob, 2; eve, 4 to pq
    Print pq
   End FOR
END PROGRAM.
```

Program:-

```
import java.util.priortyqueues
class Employee{
    string name;
    int priorily;
      public Employee(string name, int priority){
          this.name=name;
          this.priority=priority;
      }
  }

 public class priorityqueueExample{
   public static void main(String[] args){
     priorityqueue<Employee> pq=new priorityqueue<>
      pq.add (new Employee (John, 3));
      pq.add (new Employee("Alice", 1));
      pq.add (new Employee("Bob",2));
      pq.add (new Employee ("Eve", 4));
   System.out.println("Initial priority"+pq);
  Employee highest priority Employee=pq.poll();
     System.out.println("removed employee"+highest
                                                 priority);
     System.out.println("priority queue after highest
   }                                      priority");
 }
```

Output:-

```
Displaying priority Queue
    Eve-priority: 4
     John-priority:3
     Bob-priority:2
    Alice-priority:1
```

4. Pseudocode:-

```
PROCEDURE Hashmapexample
  Declare Studentmap As hashmap of integer to string
    Add 101, John; 102, Alice; 103, Bob; 104, Eve.
    print Studentmap
    Set Searchid to 103
    If studentmap contain key searchid then
      print Searchid + studentmap
    else
      print "not found"
   print studentmap
   For each id in studentmap.keyset
     print id + studentmap.get(id)
   END FOR
END PROGRAM.
```

Program:-

```
import java.util.Hashmap;
public class Hashmapexample{
    public static void main(string[] args){
      Hashmap<Integer,string> studentmap = new Hashmap<>();
        studentmap.put (101, "John");
        Studentmap.put (102, "Alice");
        Studentmap.put (103, "Bob");
        Studentmap.put (104, "Eve");
      System.out.println("Initial Hashmap "+
                              Studentmap);
int searchid= 103;
if (studentmap.containskey (search id)) {
    system.out.println ("name is present:");
}
else{
 System.out.println ("name is not present:");
}
System.out.println("Hash map after removing" +
                          studentmap);
}
}
```

Output:-

```
Initial Hashmap: {101=John, 102=Alice, 103=Bob,
                   104= Eve}
Student ID 103 corresponds to Bob
displaying all names:
  ID: 101, Name: John
  ID:103 Name: Bob.
```