CSA0914 - programming in Java for Raspberry Pi   T [illegible] Kumar
192202210

1) **Pseudocode:**

```
PROGRAM Arraylist operations
  Declare list As Arraylist of string
  ADD "apple", "Banana", "cherry", "dates" to list
  Point list
  SET remove index to 2
  Remove element to remove index from list
  Print remove element, list
  SET Search element 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
```

Program:-

```
import java.util.Arraylist;
Public Class Arraylist operations{
  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 remove index = 2;
    String removelement=list remove(removindex);
    System.out.println("remove element:"+ removed element);
    System.out.Println("list after removal:"+list);
    String search element = "data"
    If(search index ≠ -1) {
```

```
        System.out.println("element found at :" + searchIndex);
    } else {
        System.out.println("element 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 Hashset operations
    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 remove name + nameset
        SET Search name to "Alice"
        If nameset Contains search name 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 HashSetExample {
    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(newName);
        System.out.println("After adding: " + nameSet);
        String removeName = "Bob";
        nameSet.remove(removeName);
        System.out.println("After removing: " + nameSet);
        String searchName = "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:

```
Initial Set: [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 Priority queue 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 Priority Queue of Employee.
    set Pq to new priority queue (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. Priority queues
Class employee{
    String name;
    int Priority;
    Public employee (string name, int Priority) {
        this. name = name;
        this. Priority = Priority;
    }
}
Public class Priority queue Example{
    Public static void main (String [] args) {
        Priority queue < Employee > pq = new priority queue >
        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 highestPriorityEmployee = pq.poll();
System.out.println("Removed employee" + highestPriority);
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 Hashmap example
  Declare StudentMap AS hashmap of integers 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 + student map
    Else
      Print "not found"
      Print Student map
      For each id in Studentmap.key set
        Print id * Studentmap.get(id)
      END FOR
      END PROGRAM
```

Program

```
import java.util.HashMap;
public class HashMapExample4
    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(searchId)) {
            System.out.println("name is present");
        } else {
            System.out.println("name is not present");
        }
        System.out.println("HashMap 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