① Array operations :-

Aim :-

To write program on array operations.

Pseudocode :

Initialize an empty array list

string.

Add elements ("Apple", "Banana", Cherry;)

Arraylist

Specify an index to remove ("index")

Use index Method

print position

Use a Loop.

Print element.

Code :-

```
import java.util.Arraylist
Public class Arraylist operations {
public static void main (String [] args) {
list.add ("Apple");
list.add ("Banana");
list.add ("cherry);
list.add ("Date");
System.out.print in ("Initial list :" + list);
int indexToRemove : ()
```

```
int (Index.ToRemove >= 0 && indexToRemove < list.size()){
    list.remove(indexToRemove);
    System.out.println("List after removing:")
        for (String element, list){
            System.out.println(element)
    }
```

Initial List: [Apple, Banana, Cherry, Date]

List after removing element at index 2:

[Apple, Banana, Date]

The position of Date "is": 2

All elements in the list":

Apple, Banana, Date

(2)

Hash set operation:-

Aim:-

To write program on Hashset operations.

Pseudocode:-

Declare new hashset with names

Add names in names [John, Alice, Bob]

Create boolean is present

Declare & function names.contains("Bob")

```
for (String employee: employees) {
        System.out.println (employees);
    }
  }
}
```

(4) Hashmap:-

Aim:-

To write Hashmap program in java.

Pseudocode:-

Initialize new Hashmap (string)

Insert John
Alice
Bob

Index [0, 1, 2, 3]

Use for each loop

print "ID" : + entry.get (1 + ", entry
.get.value(1);

Code:-

```
import java.util.Hashmap;
    public class Hashmap oDemo {
        public static void main (string (args) {
          Hash Map (Integer. stig < students - new)
```

```
HashMap<> ();
student.put (101, "John");
student.put (102, "Alice");
student.put (103, "Bob");
String name = student.get (102);
System.out.println ("student with ID 102" + name);
   student.remove (1=3);
   for ( Hashmap.entry < Integers, String, entry, student.
                                        entry()){
   System.out.println ("+"0" entry.get())
          : name ( :+ entry.getvalue());
      }
   }
```

output:-

student with 102 : Alice

101 : John

102 : Alice.

Print "Present"

Initialize for each loop with String name : names

PRINT "name"

PRINT "name"

end loop.

Code :-

```
import java.util.HashSet;
Public class Hashoperations {
 Public static void (String() args){
        Hashset < string> names = new. Hashset<> ();
            names.add ("John");
            names.add ("Alice");
            names.add ("Bob");
            names.add ("Alice");
boolean is present = names.contains ("Bob");
 System.out.println ("is Bob present:" + is
                                    present");
    for (String name; names)
    System.out.println (names);
    }
}
}.
```

o/p :-

John, Alice

Bob, Alice.

(3) Priority Queue:

Aim :-

To write program on Priority Queue

Pseudocode :-

→ Declare new Queue (string)
Add elements (Alice, John)
Employee,
Initialize for each loop;
string employee; employee,
PRINT 'Employee');
}
}
}

Code :-

import .java.util.priority Queue;
public (class priority){
public (static Queue (string) > Employee;
New priority Queue;
Employees.add ("Alice");
Employees.add ("John");
Employees.add ("Eve");
System.out.println ("Removed highest priority)
employee.poll();