

1. ARRAY LIST OPERATIONS:

```
import java.util.ArrayList;
import java.util.Scanner;
public class ArrayListOperations {
    public static void main (String[] args) {
        ArrayList<String> list = new ArrayList<>();
        Scanner s = new Scanner (System.in);
        System.out.println ("enter names to add");
        while (true) {
            String input = scanner.nextLine();
            if (input.equalsIgnoreCase ("exit")) break;
            list.add (input);
        }
        System.out.print ("enter name to search");
        String searchname = scanner.next();
        int position = list.indexOf (searchname);
        if (position != -1) {
            System.out.println ("found");
        }
        else {
            System.out.println ("not found");
        }
        System.out.println ("current list elements");
        for (String name : list) {
            System.out.println (name);
        }
        Scanner.close ();
    }
}
```


2. HASHSET STORE COLLECTION:

```
import java.util HashSet;
import java.util Scanner;

public class Hashoperations {
    public static void main (String [] args) {
        HashSet<String> names = new HashSet<>();
        System.out.println ("enter name to remove");
        String removeName = scanner.nextLine ();
        names.remove (removeName);
        System.out.print ("enter name to check");
        String checkName = scanner.nextLine ();
        if (names.contains (checkName)) {
            System.out.println (checkName);
        } else {
            System.out.println ("current name");
            for (String name: names) {
                System.out.println (name);
            }
        }
        scanner.close ();
    }
}
```

PRIORITY QUEUE:

```
import java.util. priorityqueue;
import java.util. scanner;

public class priority queue operations {
    public static void main (String [] args) {
        priority queue<employee> queue;
        scanner S = new scanner (system.in);
        System.out.println ("enter employee names");
        while (true) {
            String input = scanner.nextLine ();
            if (input.equals ("ignore case (exit)")) {
                break;
            }
        }
        static class employee implements comparable {
            String name;
        }
    }
}
```



```

    int priority;
    employee (String name, int priority) {
        this.name = name;
        this.priority = priority;
    }
}

```

```

    public int compare to (employee other) {
        return Integer.compare (this.priority,
            other.priority);
    }
}
}
}

```

4 HASHMAP:

```

import java.util.HashMap;
import java.util.Scanner;
public class HashmapOperations {
    public static void main (String[] args) {
        HashMap<Integer, String> student map
            = new HashMap<>();
    }
}

```

```

        Scanner s = new Scanner (System.in);
        System.out.print ("enter student ID");
        int search id = scanner.nextInt ();
        if (student name != null) {
            System.out.println ("found student");
        }
    }
}

```

```

        else {
            System.out.println ("no student found");
        }
    }
}

```

```

        System.out.println ("enter student ID to remove");
        int remove ID = scanner.nextInt ();
        System.out.println ("Student current entries");
        for (Integer id : student map.keySet ()) {
            System.out.println ("ID");
        }
    }
}

```

```

        scanner.close ();
    }
}
}

```