(3)

13/08/24

ASSIGNMENT-2

JAVA PROGRAMMING

B. Abhilaash
192365076
B.E-CSE.

Assignment - 3

```
Import java.util.Array list
    public class main {
        public static void main
                (String [ ]. args) {
    Array list <String > obj = new
    Array list<> ();
        obj.add ("one");
        obj.add ("two");
        obj. add ("three");
    System.out.println ("Array list". + obj);
    }
    }
```

Arraylist in collection

```
Import java.util.list;
Import java.util.Array.list.
class main {
    public static void main
                (String[] args){
list <integer> numbers = new array
                list();
```

```
number.add(1);
number.add(2);
number.add(3);
System.out.println("List: " + numbers);
int getnumber = numbers.get(2)
System.out.println("Acuired element:".
                   +getnumber);

int removeNumber = numbers.
                   remove(1);

System.out.println("Removed Element:"
                   + remove
                   number);
}
}
```

Linked list :-

```
Import java.util.list;
Import java.util.linked list;
Class main{
    public static void main
                (String[] args){
List<integer> number = new linked
                       <>();
number.add(1);
number.add(2);
number.add(3);
```

```
System.out.println("List "+number);
int number= number.get(2);
System.out.println("Accessed element:"
                    +number);
    int Index= number.Index of(2);
System.out.println("pointer of 2 is"
                    +index);
    int removed number
                =number.
                remove(1);
System.out.println("Removed element"
                    +removed number);
}
```

Vector:

```
Import java util.Iteration
Import java util.Vector.
Import java util.Collections
class main{
    public static void main(String[] args){
        Vector<String> fruits = new linked_
                                list<>();
fruits.add("Apple");
fruits.add("Mango");
fruits.add("orange");
```

```
System.out.println("Linked List = " fruits);
fruit.add(2, "Banana");
System.out.println("Linked list " + fruits);
Collections.reverse(fruits);
System.out.println("fruits in reverse order
                    " + fruits);

Collection.sort(fruits);
System.out.println("Fruits in Ascending
                    order " + fruits);

Collections.sort(fruits, Collection.reverseOrder());
System.out.println("Fruits in
                    descending order " +
                    + frist);

System.out.println("fruits in the
                    basket");
for(int i=0; i<fruits.size(); i++){
System.out.println("fruits in the
                    bracket");
for(int i = fruits.size()-1; i>=0;
                    i--);
    System.out.println(fruits.get(i));
}
}
```

Stack (Push, pop, peek, empty)

```
Import.java.lang;
Import.java.util.Stack
class main{
    public static void main (String[] args){
    Stack<String> fruits.newStack<>();
    fruits.polish("Apple");
    fruits.push("orangke");
    fruits.push("Mango");
    System.out.println("Stack:"+fruits);
String remove=fruits.pop();
System.out.println("Stack"+remove);
fruits.push("pineapple");
    System.out.println("Stack+fruits);
String display=fruits.pack();
System.out.println("Stack"+display);
int position=fruits.Search.
System.out.println("position of fruit"
                    +position);
boolean e=fruits.empty();
System.out.println("Is the stack is empty"
                    +e);
fruits.clear();
```

```
boolean = fruits.empty();
System.out.println("Is the stack is
                    empty:"+r1);
}
}
```

Queue

```
Import java.lang.*;
Import java.util.LinkedList;
Import java.util.queue
Class main{
    public static void main(String[] args){
    Queue<String> fruits = new LinkedList
                           <>();
    fruits.add("Apple");
    fruits.add("orange");
    fruits.add("Mango");
System.out.println("Queue = " + fruits);
String ry= fruits.remove();
System.out.println("Queue "+r);
System.out.println("Queue "+r'fruit);
String display = fruits.peek();
System.out.println("Stack"+fruits);
boolean e = fruits.isempty();
```

```
System.out.println ("Is the queue is empty "+e);
fruits.clear();
boolean1 = fruits.isempty();
System.out.println (" is the queue is empty"+e);
}
```

Hash map

```
Import java.util.map;
Import java.util.Hashmap;
class main {
    public static void main (String[] args) {
        map<Integer,String>fruits = new hashmap
                                    <>();
        fruits.put (1,"Apple");
        fruits.put (2,"orange");
        fruits.put(3,"mango");
    System.out.println ("Map:"+fruits);
    System.out.println ("values"+fruits
                                values());
    System.out.println ("Entries" ~~entryset~~()
                                +fruits.entryset());
    boolean value = fruits.remove.
                    (2,"orange");
```

```
System.out.println("Available In the
                    bucket :" + value.I);
fruits.replace(3, "Mango", "Banana");
System.out.println("After replacing
                    the item " + fruits);
}
}
```