Collections () R. Hariba Kyle 192311087 List infort jova util list; public class Main { public thatic void noin ( String [ I wags) { A geray List < String > obj = new Array list < >(); obj. add ("one"); obj - add ("T,00"); obj. odd ("Three"); System. out. firstla ("A rorry List: "+ obj 1; outfut: - [ " one", " Two ", " Thra "]; infort jova , stil . Land Array List; class Mais ? public static said main ( String [] args) { List < integer > numbers = new Axanglist < >(); numbers, add (1); numbers. add (2); numbers, add (3); system . out . println (" List: " + numbers); int get number = numbers get(e); System out fristly ("Accessed Floment: " + get musber). ist genove number = numbers. remove (1); System. out. printle ("Renove element: " Asses remove Number);

```
Linked list
import java. util. List;
import jova - itil. Luled list;
   fublic static soid main (S Tring [] args) {
Class Main &
    List cinteger > numbers = new tister Linked list < >0);
    munlers. add (1).
    numbers add (2)
    numbers . add (3);
     System. out - frintln ("Accessed Element: " + number);
     int index = numbers index of (2)
     System. out . prientle ("Position of 2 "rindex);
     int removed Number = number. remove(1)
     system. out . pointles ( "Position of 2 "+ index);
     int removed Number = Number. remove (1);
     System. out. printles ("Removed about" + gramowed number).
   Vector
   inhert java util victor.
   class Main &
   Aublic static void main (String [ ] args) }
       vector String > fruits = new vector < >(1;
       fruits . add ("Apple");
       fruits add (" Oxange");
       fruits . add ("Mango");
       System. out. println ("Vector: "+ fruits);
       String element = fruits. get (2);
```

```
int size = fruits . size ();
Sorten. out . frintln ("Sige: "+ iese);
iterators < string > iterate = fruits : iterator ();
System. out . print ("Voctor");
Evhile ("therate has Noset()) {
      System. out. print (iterate next());
Collection revouse
infort . jave . util . * ;
infort. java. itil. Iterator;
inport . java . util . I inked List;
emport java util collections;
Closed Main {
     public static Void Main (String [] args) {
     List < string > fruits = new Linked list < >();
     fruity. add ("Apple").
    fruits . add (" Orange");
    fruits . odd ("Manga").
    System. out . printles ("I inked list: "+ fruits);
    fruits. add (2, "Banana");
    System. out. println ("Linked list: " + fruits );
    collections . noverse (fruits);
     System out frintly ("Fourts in oreverse order: " + fruits ),
     rollection. at sort (fruits);
    System out println ("Fruits in Ascending order; " fruits);
```

```
System. out. printle ("Finity in the basket: ");
For (int i = fruits. size () -1; i >=0; 1--) {
     System . ait . println (faints . get(i));
Stack ( push, Poh, peak, enfity)
inhort . java . long *;
inhort . java . util . Stack ;
Class Main }
   public Static void mais (String [] args) }
        Stack ( thring 7 fruits = new stack ( <> ();
        fruits . hut ("Apple");
       fruits . bush (" orange ");
       fruits - furt ("Mango");
       System. out. println ("Stack:" + fruits);
        5 tring removed = fruits hoft );
       System. out . printly (" Stack: "+ semover);
       fruits. Just ("Pinkapple");
       System. out - printles ( "Stack: "+ friets );
       5 bring dishlay = fruits . peek();
       System . out - println ("Stack: "+ display);
boolean e = frits . enty ();
System . outfast . println ("It the stack
```

```
Queul
import java . long . * ;
infort java util . Linked List;
infort jova . Itil . Queue;
Class Main {
 fablic static word main (String [] args) {
    Queul < string > faits = new linked list < 7 ();
    fruits add ("Affle");
    fruits add ("Oxonge");
     frits add ("Mango");
    System at fint ("Osere: "+91);
    system. out . perintle ("Ocene:" + fruits);
     String dirlay fruits feek ();
     sorten at fristle ("Is the Queue is ently: "+e);
     fruits. clear ();
 infort jova . long . ;
 intert jour . itil. Linked list;
 infort java. Itil. Array Dewse;
 Clars Main 9
   public static void rain (String[] args)?
```

Agray Deque ( 5 tring [ ] args ) { Affility . add ("Apple"); fruits add ( Bonona'); fuits . add (" Oxonge"); fruits . add first ("orange"); fruits. add Last ("Mango"); System out printles ("Dequeue: " + fruits); String or = fruits . remove (); System out frietly ("Dequeue: "+ 91): Storing display = fruits feek (); System. out println ("Dequeve: " + diffey) boolean e - fruits is Emply (); family clear (V) System. out. println("Ently Dequere: "+ fruit ). booleaned = fruits is Empty (); System. act: println ("Is the dequeue Emply: "+ 21). Hash mah: import java. will. Map; import java . atil . HushMah; Class Main & Jullie Static Void main (Storing [] args) & Map ( I religion String > fruits = new Hash Map < >(); failts - pet (1, "Apple"); fruits . fut (2," Orange"); fruits . fut (3, "Margo"); System. out println ("Map", Fruits); System. out. println ("Keys: "+ whent)). fruits. key set () System. out . println (" wales & fruits walres)); System. out. feintlet Entries: " + frints Entry set ()); boolean valre = fruits. remove (2, "orange"). System out . println (" Record removed value: "+ walke); System. out. println (" New Map: " + Fruits );

HILLING HILLIAM INC.