**HashMap:**

* (HashMap is similar to Map in c++)
* HashMap stores items in key/value pairs
* We can access values using index of another type (i.e String)
* Here Key is nothing but index
* We can use hashmap to store any types of key and values, like String keys and Integer values or String keys and String values

Syntax:

HashMap<Integer,Integer> map1 = new HashMap<Integer,Integer>();

HashMap<Integer,String> map2 = new HashMap<Integer,String>();

Important methods:

1. put(<key>,<value>):

//used to assign values for particular key

map1.put(1,100);

map1.put(120,120);

map2.put(10,”raheez”);

1. get(<key>):

//used to get value of specified key

map1.get(1);

//100

map1.get(10);

//null

map2.get(10);

//”raheez”

1. values();

//to get all the values in hashmap

for(int i: map1.values){

System.out.println(i);

}

//above code prints all the values in HashMap

1. keyset();

//used to get all the keys in HashMap

for(int i:map1.keySet()){

System.out.println(i);  
}

//above code prints all the keys in hashMap

//to print all keys as well as values we use below code

for(int i : map1.keySet()){

System.out.println(“key:”+i+”value”+map1.get(i));  
}

1. size();

//used to get size of HashMap

int length = map1.size();

1. remove(<key>):

//used to remove particular key element

map1.remove(1)

1. clear():

//used to remove all the elements

map1.remove();

**ArrayList:**

* It is similar to vector in c++
* It is nothing but resizable array
* Difference between built in array and arraylist is , size of array cannot be modified ,but in arraylist we can add any number of elements and remove whenever you want.

Syntax:

ArrayList<Integer> array1 = new ArrayList<Integer>();

ArrayList<String> arrray2 = new ArrayLisy<Integer>();

Important methods:

1. add(<element>):

//used to add elemets to the array

array1.add(1);

array1.add(100);

array2.add(“raheez”);

1. get(<index>):

//to get element at specified position(index)

array1.get(0);

//1

array1.get(2);

//arrayIndexOutOfBound

array2.get(0);

//”raheez”

1. set(<index>,<element>);

//used to modify the element in a specified location(index)

array1.get(0);

//1

array1.set(0,10);

array1.get(0);

//10

1. size();

//to get length of array

int length = array1.size();

1. remove(<index>);

//remove particular element

array1.remove(0);

1. clear();

//to delete all the elements in the array

1. contains(<item>);

//to check wheter particular element present in array

//it returns boolean value

System.out.println(array1.contain(1));

//true

**HashSet:**

* (Hashset is similar to set in c++)
* It is used to store unique items
* It is similar to arrayset but doesn’t contain any duplicate elements

Syntax:

HashSet<integer> set1 = new HashSet<Integer>();

HashSet<String> set2 = new HashSet<String>();

Important methods:

(methods are same as arraylist but here we don’t use get,set instead we use contains in place of get)

**LinkedList:**

* The LinkedList class is a collection which can contain many objects of the same type, just like the ArrayList.
* The LinkedList class has all of the same methods as the ArrayList class because they both implement the List interface. This means that you can add items, change items, remove items and clear the list in the same way.
* However, while the ArrayList class and the LinkedList class can be used in the same way, they are built very differently.

Syntax:

LinkedList <Integer> list1 = new LinkedList<Integer>();

LinkedList <String> list2 = new LinkedList<String>();

Important methods:

//methods are same as arraylist ,, and it has some extra methods like

1. addFirst();
2. addLast();
3. removeFirst();
4. removeLast();
5. getFirst();
6. getLast();

**List:**

List is same as ArrayList

It is not a class it is interface

Syntax:

List<Integer>list1 = new ArrayList<Integer>();

List<List<integer>> list2 = new ArrayList<List<Integer>>();