**Assignment 1:**  
Write a program to remove the duplicate elements from the list and then iterate all the unique elements.

public class RemoveDuplicate {

public void duplicate(){

List<String> names=new List<String>();

Set<String> newnames=new Set<String>();

names.add('Ramesh');

names.add('Rakesh');

names.add('Mathew');

names.add('Max');

names.add('Mathew');

system.debug(+names);

newnames.addall(names);

system.debug(+newnames);

}

}

Graphical user interface, text, application

Description automatically generated  
**Assignment 2:**  
Write a program where you need to iterate Map's values using keySet and values method.

public class MapDemo {

public void demo(){

Map<string,string> names=new Map<string,string>();

names.put('as1', 'Mathew');

names.put('as2', 'John');

names.put('as3', 'Max');

names.put('as4', 'priya');

names.put('as5', 'Kenny');

System.debug(names);

System.debug(names.get('as1'));

Set<string>key=names.KeySet();

for(string newkey:key){

system.debug('Key is = '+newkey + 'value is = ' +names.get(newkey));

}

System.debug('Values of all the keys '+names.values());

Iterator<String> itr= key.iterator();

while(itr.hasNext()){

System.debug( names.get( itr.next()));

}

}

}

A screenshot of a computer

Description automatically generated

**Assignment 3:**  
Write a program to handle nullpointerexception and listexception, in any case if either of the above exception is not occured into your program but it might be the chance of throwing any other exception so in this case you have to write your code in a way so that your program can handle, it won't break.

public class ExceptionHandling {

public void getnewdemo(){

try{

Addition ad;

ad.dosum();

List<String> newdemo = new List<String>();

newdemo.add('max');

newdemo.add('Kenny');

newdemo.add('will');

system.debug(newdemo[3]);

}catch(ListException e){

System.debug('List exception fired');

}catch(NullPointerException ne){

System.debug('Please donot call the method with the object that poiting to null');

}catch(Exception me){

System.debug('Server is busy');

}

finally{

System.debug('Finally will execute with or without exception');

}

}

}

Graphical user interface, text, application, Word

Description automatically generated