**Experiment -2.1**

**Student Name:Parikshit sharma UID:19BCS4520**

**Branch: CSE-IOT Section/GroupIOT/A**

**Semester:4 Date of Performance:24/3/2021**

**Subject Name:PBLJ Subject Code:CSP 296**

**1. Aim/Overview of the practical:**

In this experiment, examples of using Set in Java.

**2. Task to be done:**

(a) Create a program to set view of Keys from Java Hash table.

(b) Create a program to show the usage of Sets of Collection interface.

**3. Apparatus(For applied/experimental sciences/materials based labs):**

NetBeans IDE

**4. Algorithm/Flowchart (For programming based labs):**

Step1 create new, empty set

Step2 print element of the set

Step3 create the list add some elements

Step4 create the set by using constructor

Step5 then remove one element from myset2 and then again compare it

Step5 if set have all elements of the list use the iteration

Step6 use for the each set

Step7 clear all the elements

Step8 check number of the elements

Step9 create array with content of the set

Step10 exit

**5. Theme/Interests definition( For creative domains):**

* Hashset:It constructs a new, empty set.

HashSet(Collection col): It constructs a new set, which contains the elements of the given collection.

* HashSet(int intialCapacity):It constructs a new, empty set, with the specified initial capacity.
* HashSet(int intialCapacity, float loadFactor):It constructs a new, empty set, with the specified initial capacity and load factor.

**6. Steps for experiment/practical:**

(a) Create a program to set view of Keys from Java Hash table

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package hashtableconcept;

import java.util.Enumeration;

import java.util.Hashtable;

/\*\*

\*

\* @author dell

\*/

public class Hashtableconcept {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Hashtable h1=new HashTable();

h1.put(1,"Parikshit");

h1.put(2,"Avishu");

h1.put(3,"khesha");

h1.put(4,"Ram");

Enumeration e=h1.keys();

while(e.hasMoreElements()){

System.out.println(e.nextElement());

}

}

}

(b) Create a program to show the usage of Sets of Collection interface.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package hashsetex;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.HashSet;

import java.util.Iterator;

import java.util.Set;

/\*\*

\*

\* @author dell

\*/

public class Hashsetex {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Set<String> mySet1 = new HashSet<String>();

// We add a few elements

mySet1.add("1");

mySet1.add("3");

mySet1.add("1");

mySet1.add("2");

System.out.println("mySet1: " + mySet1);

List<String> list = new ArrayList<String>();

list.add("1");

list.add("3");

list.add("1");

list.add("1");

list.add("2");

list.add("3");

Set<String> mySet2 = new HashSet<String>(list);

System.out.println("list: " + list);

System.out.println("mySet2: " + mySet2);

System.out.println("MySet1 matches mySet2: " + mySet1.equals(mySet2));

mySet2.remove("1");

System.out.println("mySet2: " + mySet2);

System.out.println("MySet1 matches mySet2: " + mySet1.equals(mySet2));

System.out.println("MySet1 contains all the elements: " + mySet1.containsAll(list));

System.out.println("MySet2 contains all the elements: " + mySet2.containsAll(list));

Iterator<String> iterator = mySet1.iterator();

while (iterator.hasNext()) {

System.out.println("Iterator loop: " + iterator.next());

}

for (String str : mySet1) {

System.out.println("for-each loop: " + str);

}

mySet1.clear();

System.out.println("mySet1 is Empty: " + mySet1.isEmpty());

System.out.println("mySet1 has: " + mySet1.size() + " Elements");

System.out.println("mySet2 has: " + mySet2.size() + " Elements");

String[] array = mySet2.toArray(new String[mySet2.size()]);

System.out.println("The array:" + Arrays.toString(array));

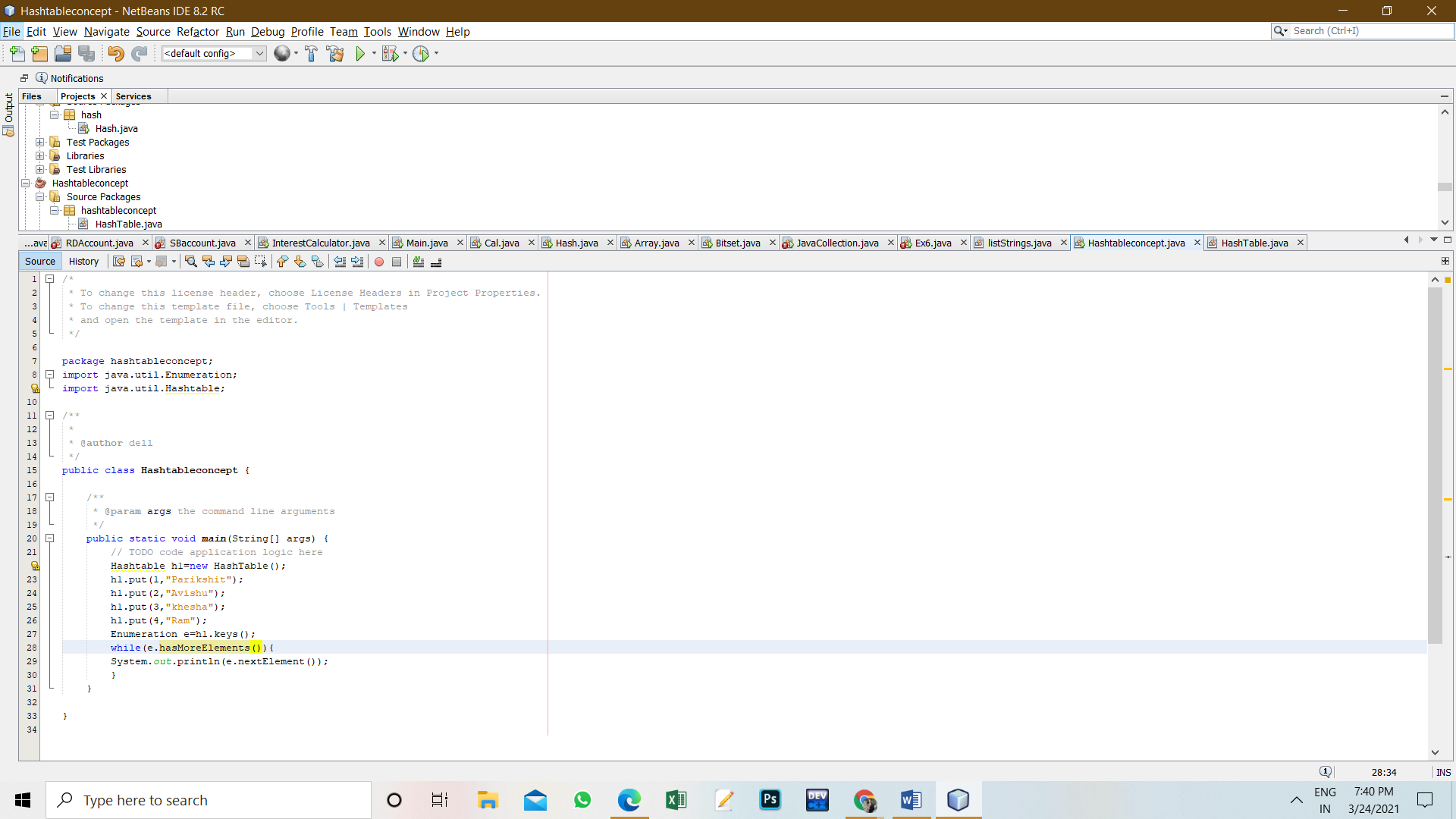
}

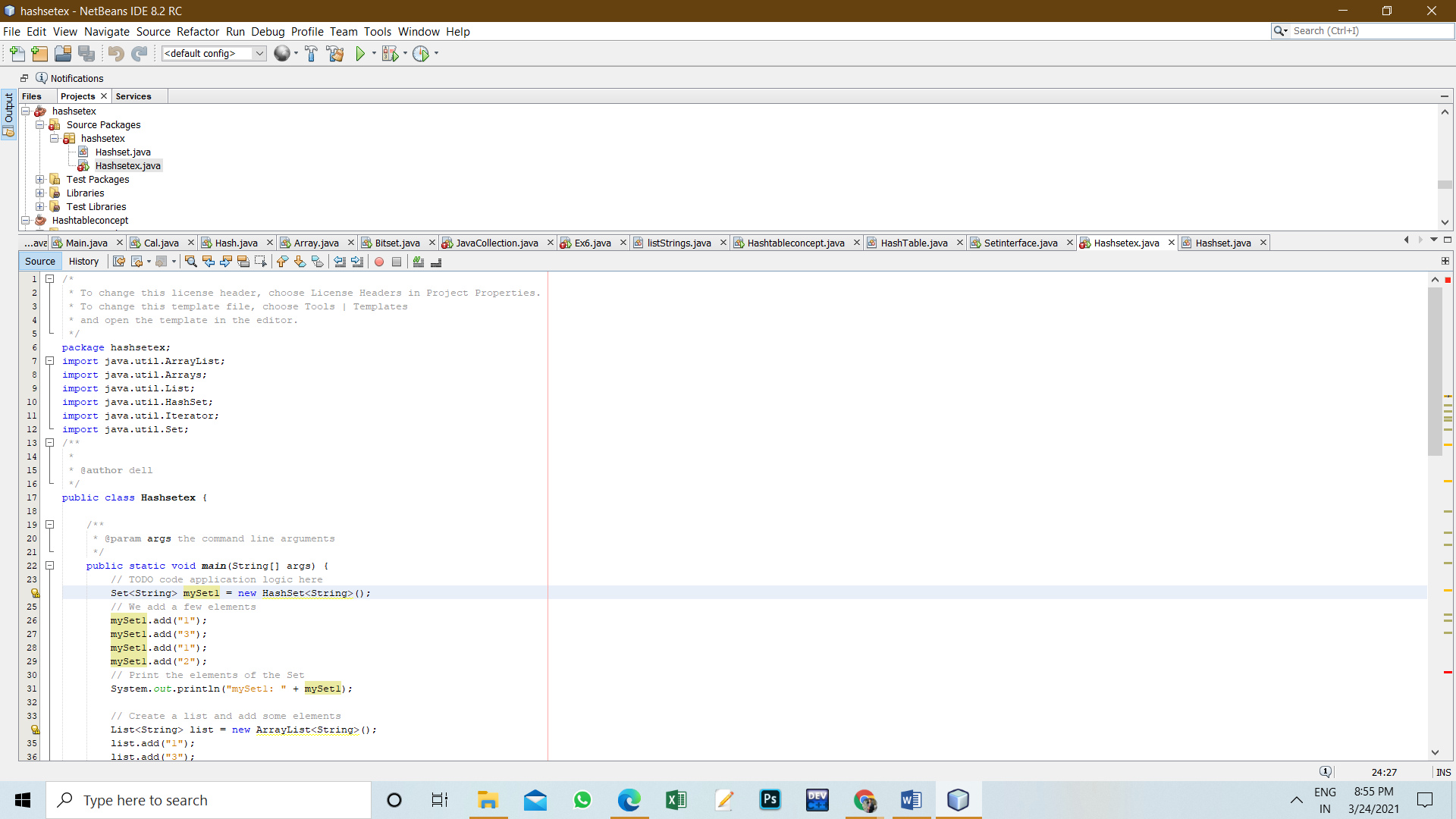
}

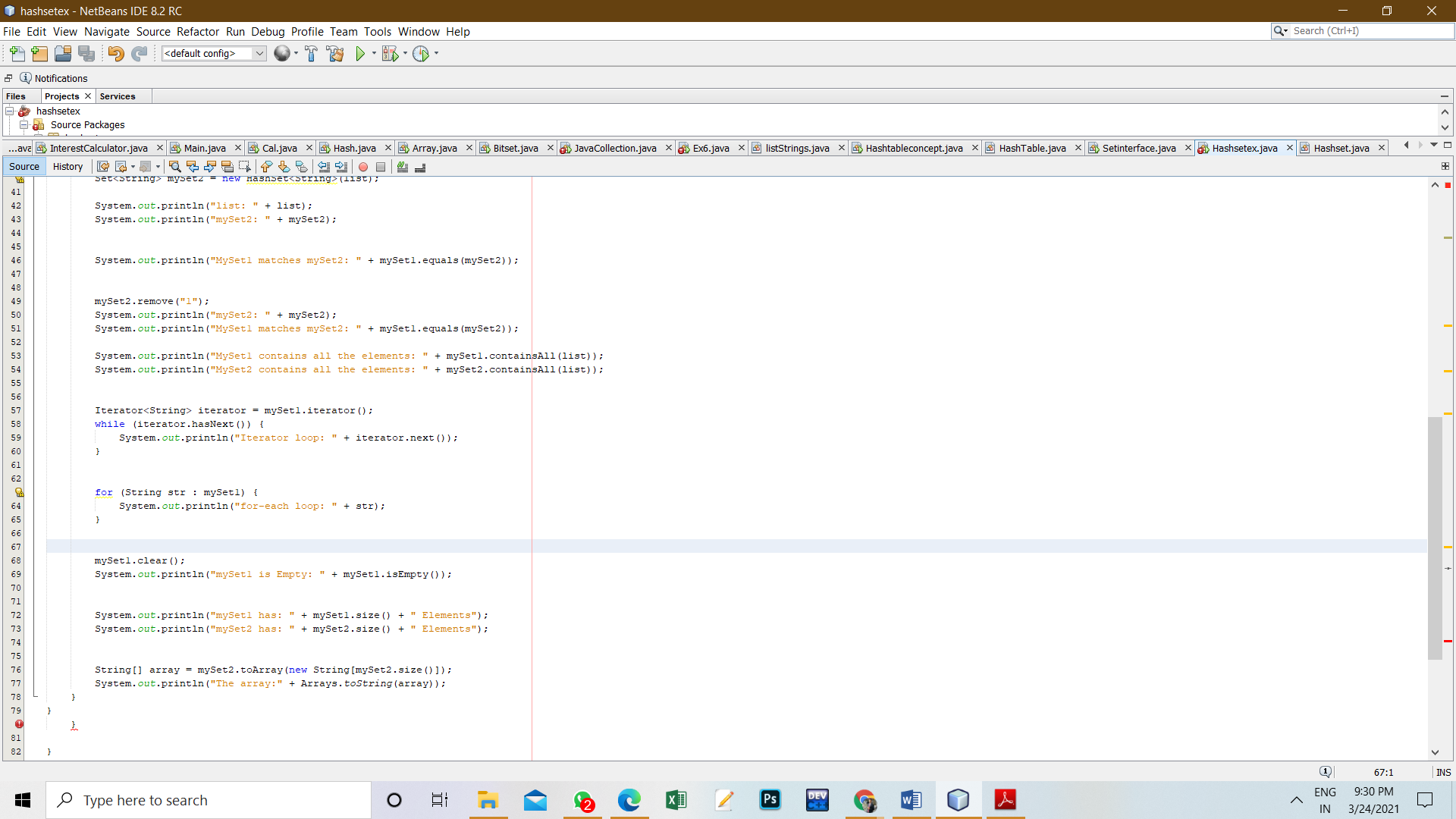
}

}

**7. Observations/Discussions(For applied/experimental sciences/materials based labs):**

****

****

****

**8. Percentage error (if any or applicable):**

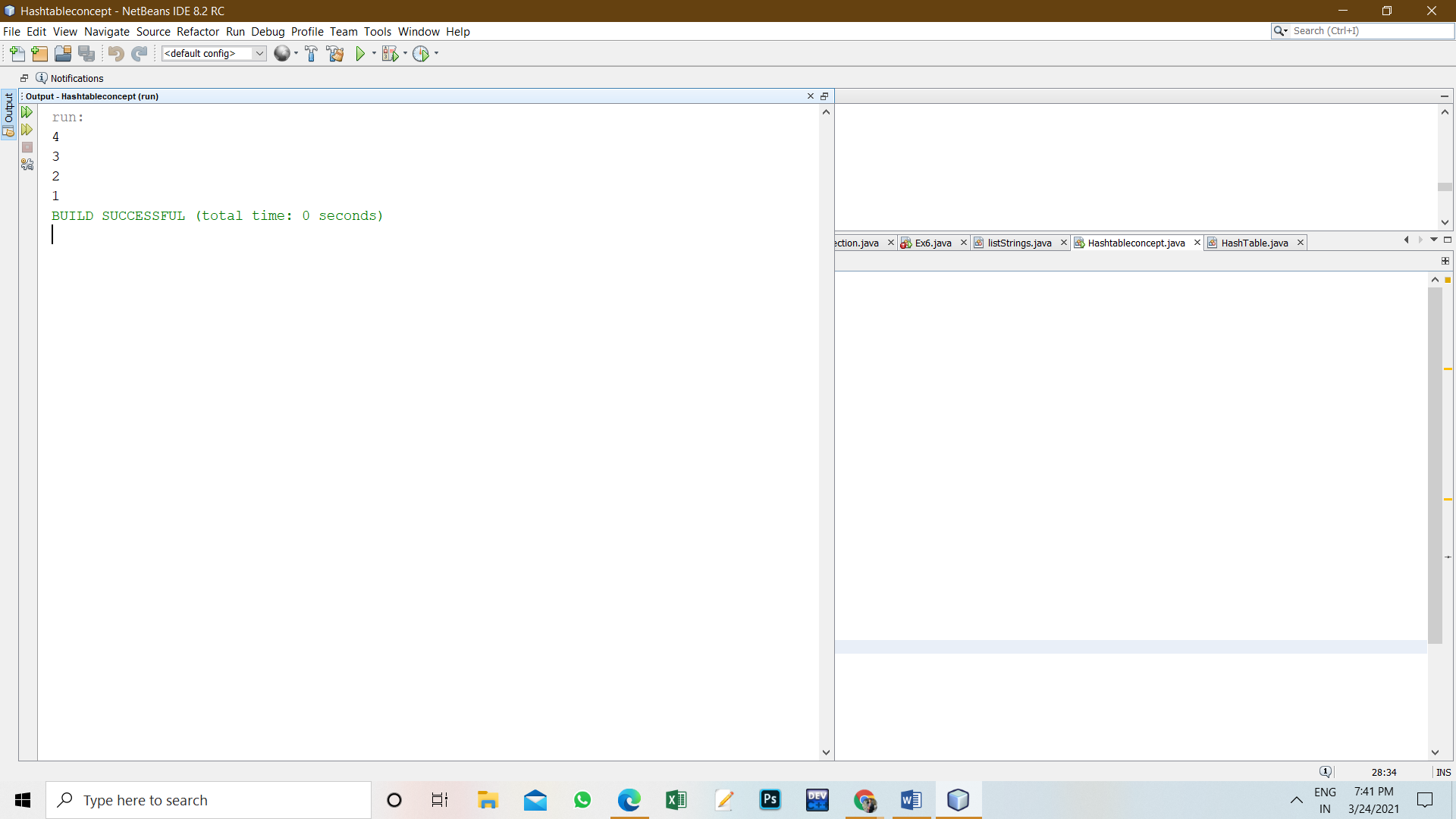
No

**9. Calculations/ Chemical Reactions / Theorems /Formulas used etc :**

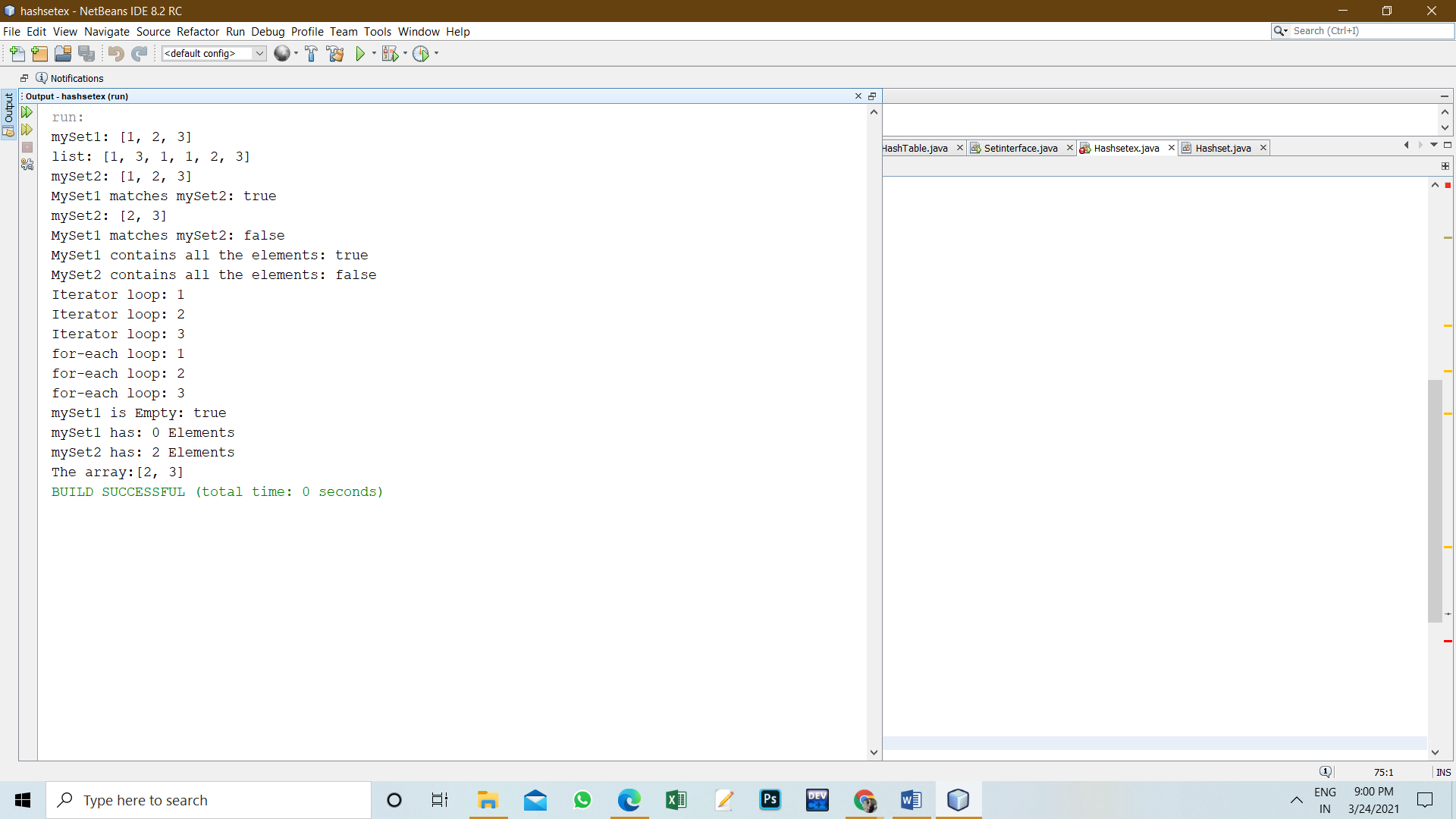
NO

**10. Result/Output/Writing Summary:**

(a) Create a program to set view of Keys from Java Hash table

****

(b) Create a program to show the usage of Sets of Collection interface.

****

**11. Graphs (If Any): Image /Soft copy of graph paper to be attached here**

no

**Learning outcomes (What I have learnt):**

* 1. 1 HashSet stores the elements by using a mechanism called **hashing.**
  2. 2 HashSet contains unique elements only.
  3. 3 HashSet allows null value.
  4. 4 HashSet class is non-synchronized.
  5. 5 HashSet doesn't maintain the insertion order. Here, elements are inserted on the basis of their Hashcode.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |