1. **Write a program to implement the following methods using ArrayList : size(), isEmpty(), contains(), containsAll(), indexOf(), lastIndexOf(), get(), remove(), addAll(), subList(), hashCode().**

**Code:**

import java.util.\*;

class q1{

@SuppressWarnings("unchecked")

public static void main(String args[]){

ArrayList ar=new ArrayList();

System.out.println("Is list empty: "+ar.isEmpty());

System.out.println();

System.out.println("ArrayList1: ");

ar.add(new Integer(6));

ar.add(new Integer(56));

System.out.println(ar);

System.out.println();

System.out.println("Size of list: "+ar.size());

System.out.println();

System.out.println();

System.out.println("Is list empty: "+ar.isEmpty());

System.out.println();

ar.add(new Integer(4));

ar.add(new Integer(21));

ar.add(new Integer(90));

ar.add(new Integer(56));

System.out.println("ArrayList1: ");

System.out.println(ar);

System.out.println();

System.out.println("Size of list: "+ar.size());

System.out.println();

System.out.println();

System.out.println("Element at 5th postion: "+ar.get(5));

System.out.println("Index of element 4: "+ar.indexOf(4));

System.out.println("Last index of element 56: "+ar.lastIndexOf(56));

System.out.println("Hashcode for arraylist1: "+ar.hashCode());

System.out.println("Does the arraylist1 contain 5: "+ar.contains(5));

System.out.println();

System.out.println();

ArrayList l1=new ArrayList();

l1.add(new Integer(4));

l1.add(new Integer(21));

l1.add(new Integer(5));

l1.add(new Integer(6));

System.out.println("ArrayList2: ");

System.out.println(l1);

System.out.println();

System.out.println("Does the ArrayList1 contains all the elements of ArrayList2: "+ar.containsAll(l1));

System.out.println("Is ArrayList2 elements added to ArrayList1: "+ar.addAll(l1));

System.out.println();

System.out.println("ArrayList1: ");

System.out.println(ar);

System.out.println("Sublist: "+ar.subList(0,4));

System.out.println();

ar.remove(3);

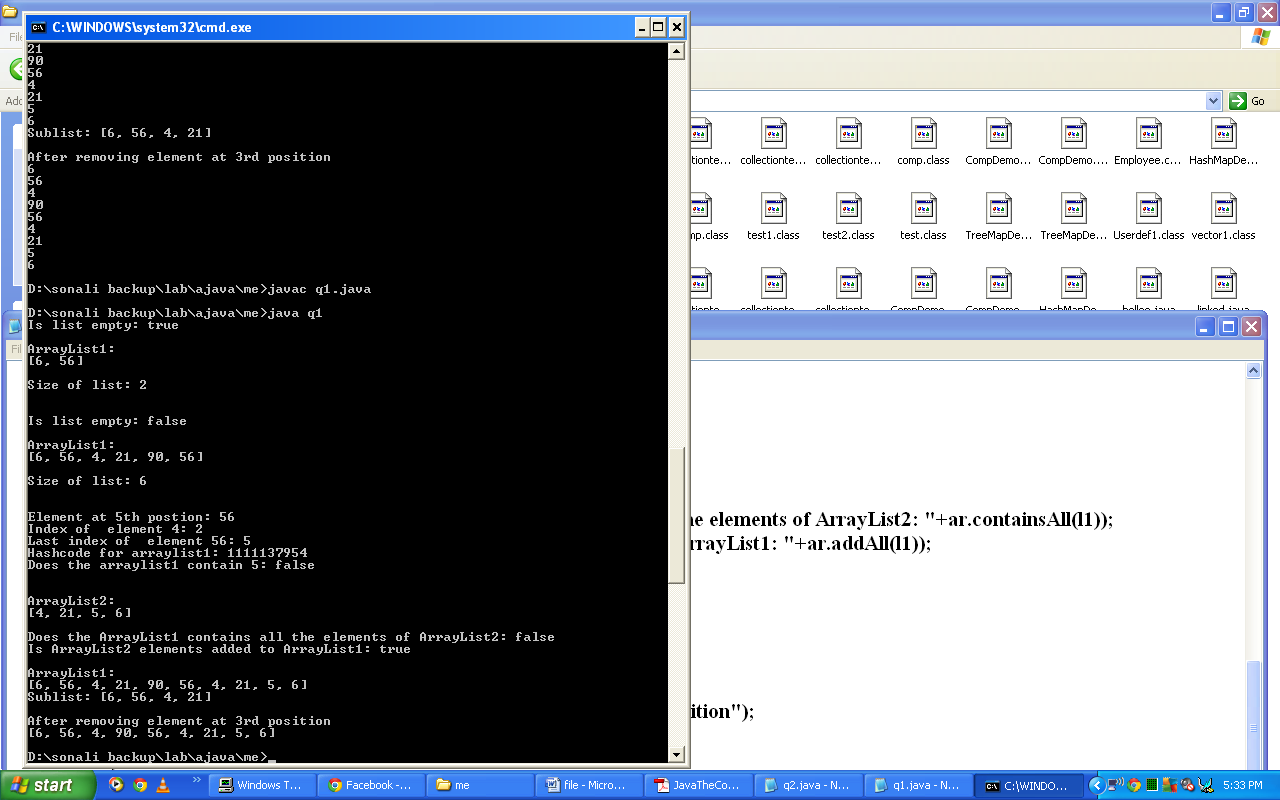
System.out.println("After removing element at 3rd position");

System.out.println(ar);

}

}

**Output:**

****

1. **Write a program to implement the following methods using ArrayList: clear(), clone(), toArray(), add(), iterator(), listIterator().**

**Code:**

import java.util.\*;

class q2{

@SuppressWarnings("unchecked")

public static void main(String args[]){

ArrayList ar=new ArrayList();

ar.add(new Integer(90));

ar.add(new Integer(35));

ar.add(new Integer(40));

ar.add(new Integer(35));

ar.add(new Integer(43));

System.out.println(ar);

System.out.println();

ArrayList ar1=(ArrayList)ar.clone();

System.out.println("Is clone empty: "+ar1.isEmpty());

System.out.println("Cloned ArrayList:");

System.out.println(ar1);

ar1.clear();

System.out.println("Is clone empty: "+ar1.isEmpty());

System.out.println();

ar.add(2,new Integer(10));

System.out.println("After adding 10 at 2nd position");

System.out.println(ar);

System.out.println();

System.out.println("Visiting elements using Iterator");

Iterator it1=ar.iterator();

while(it1.hasNext())

System.out.println(it1.next());

System.out.println();

System.out.println("Visiting elements using ListIterator");

ListIterator lit1=ar.listIterator();

while(lit1.hasNext())

System.out.println(lit1.next());

System.out.println();

System.out.println("Visiting elements in reverse order using ListIterator");

while(lit1.hasPrevious())

System.out.println(lit1.previous());

System.out.println();

System.out.println("Converting arraylist to array and printing it: ");

Integer arr[]=new Integer[ar.size()];

ar.toArray(arr);

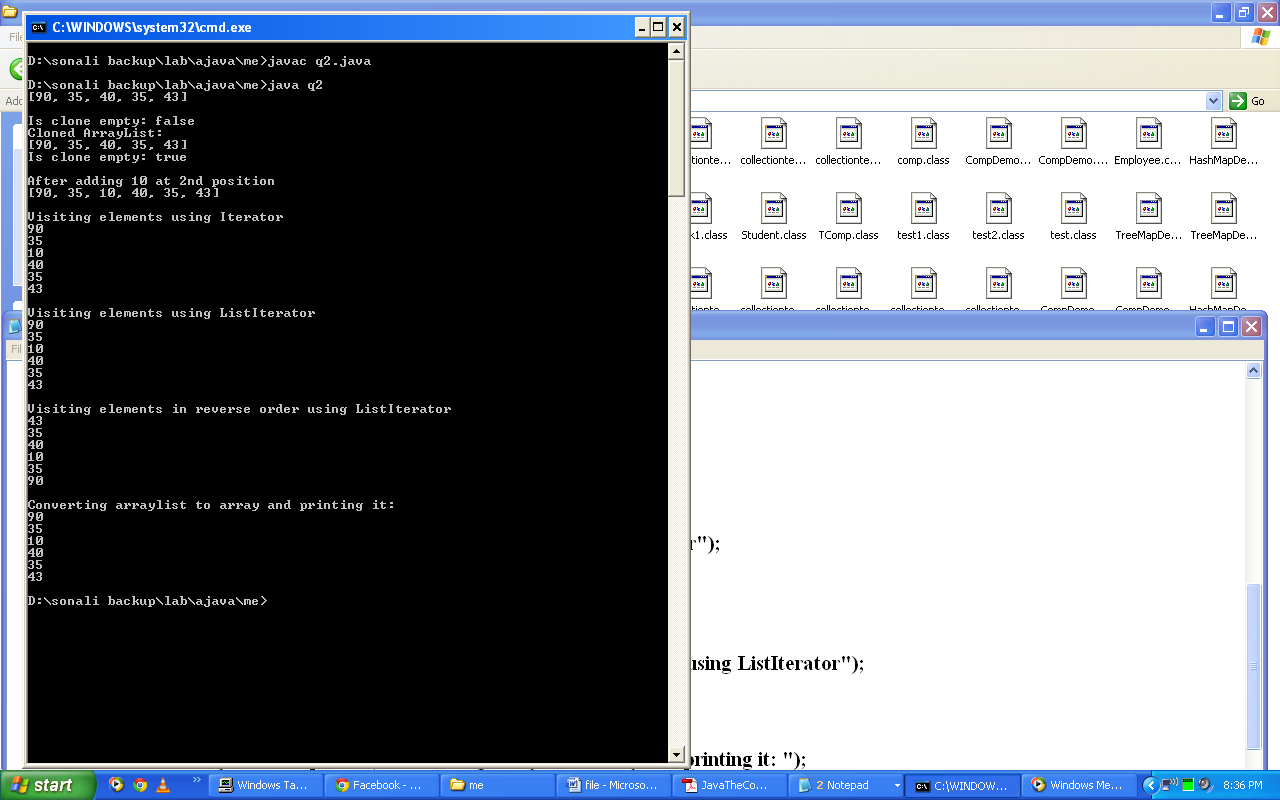
for(int el:arr)

System.out.println(el);

}

}

**Output:**

****

1. **Write a program to implement the followings methods using linked list: addFirst(), addLast(), removeFirst(), removeLast().**

**Code:**

import java.util.\*;

class q3{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedList l1=new LinkedList();

l1.add("Hello");

l1.add("Bye");

l1.add("Hi");

System.out.println("Linked List elements:");

System.out.println(l1);

System.out.println();

System.out.println("Adding 'Happy' as First element and 'Sad' as Last element.");

l1.addFirst("Happy");

l1.addLast("Sad");

System.out.println("Linked List after adding new elements:");

System.out.println(l1);

System.out.println();

System.out.println("Removing First Element and Last Element.");

l1.removeFirst();

l1.removeLast();

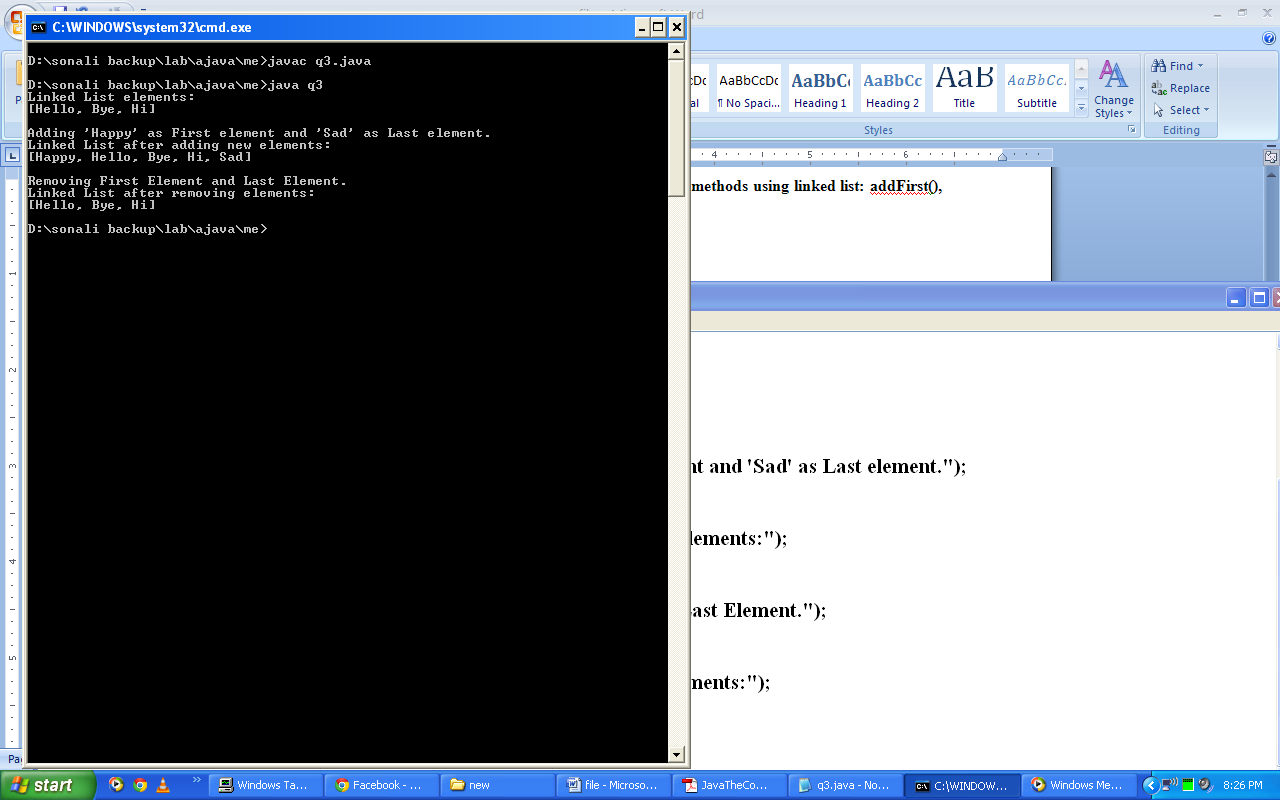
System.out.println("Linked List after removing elements:");

System.out.println(l1);

}

}

**Output:**

****

1. **Write a program to implement all the functions of Arrays class.**

**Code:**

import java.util.\*;

class q4{

public static void main(String args[]){

Object arr[]=new Object[10];

for(int i=0;i<10;i++)

arr[i]=i\*-3+2;

System.out.println("Array Elements: ");

for(int i=0;i<10;i++)

System.out.println(arr[i]);

System.out.println();

System.out.println("Searching element using binary search :");

int index= Arrays.binarySearch(arr,-10);

System.out.println("Index of -10 is " + index);

System.out.println();

System.out.println("Array Elements in Sorted Order : ");

Arrays.sort(arr);

for(int i=0;i<10;i++)

System.out.println(arr[i]);

System.out.println();

System.out.println("Creating copy of first three elements of array");

Object ar1[]=Arrays.copyOf(arr,3);

for(Object a:ar1)

System.out.println(a);

System.out.println();

System.out.println("Creating copy of elements of array from 3 to 5 position");

Object ar[]=Arrays.copyOfRange(arr,3,6);

for(Object a:ar)

System.out.println(a);

System.out.println();

System.out.println("Is two arrays equal: "+Arrays.equals(ar,ar1));

System.out.println("Is two arrays equal: "+Arrays.deepEquals(ar,ar));

System.out.println();

System.out.println("Filling first five array elements with 2");

Arrays.fill(arr,0,4,2);

for(int i=0;i<10;i++)

System.out.println(arr[i]);

System.out.println();

System.out.println("Fill all array elements with 2");

Arrays.fill(arr,2);

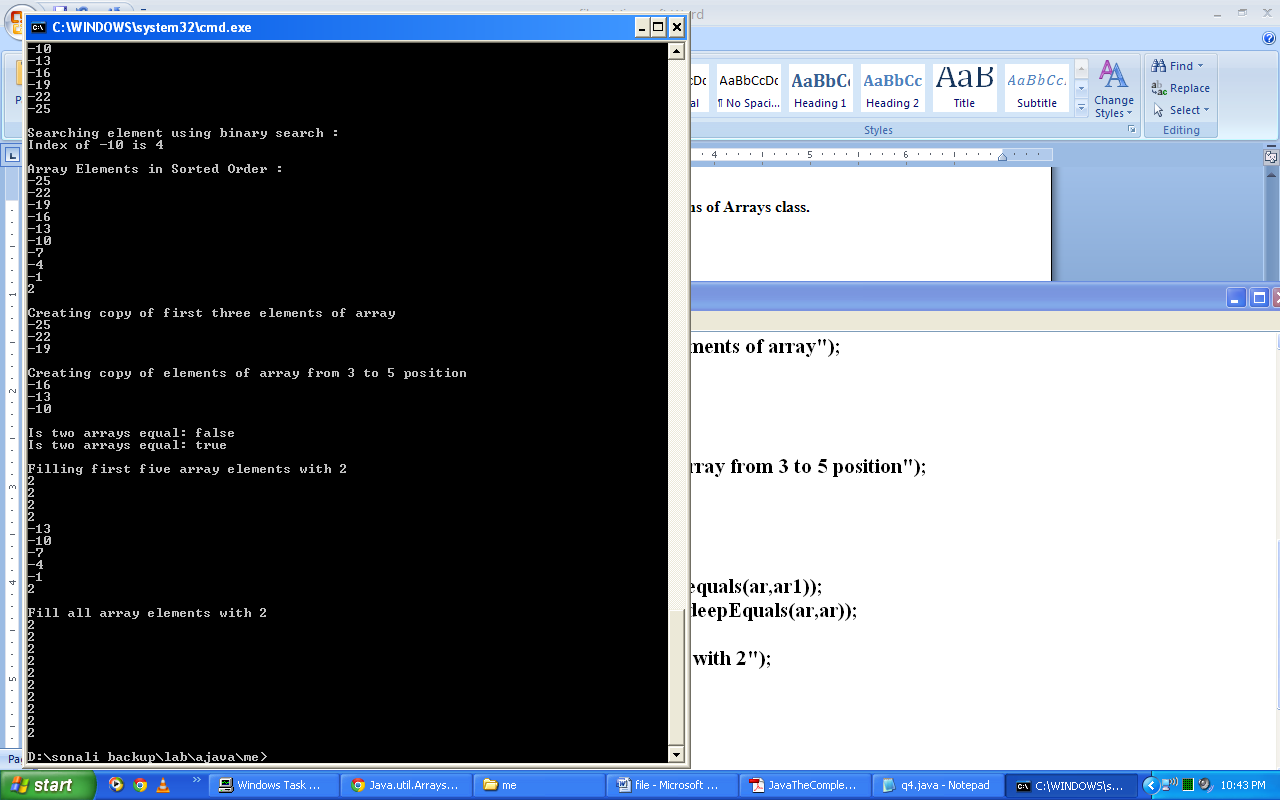
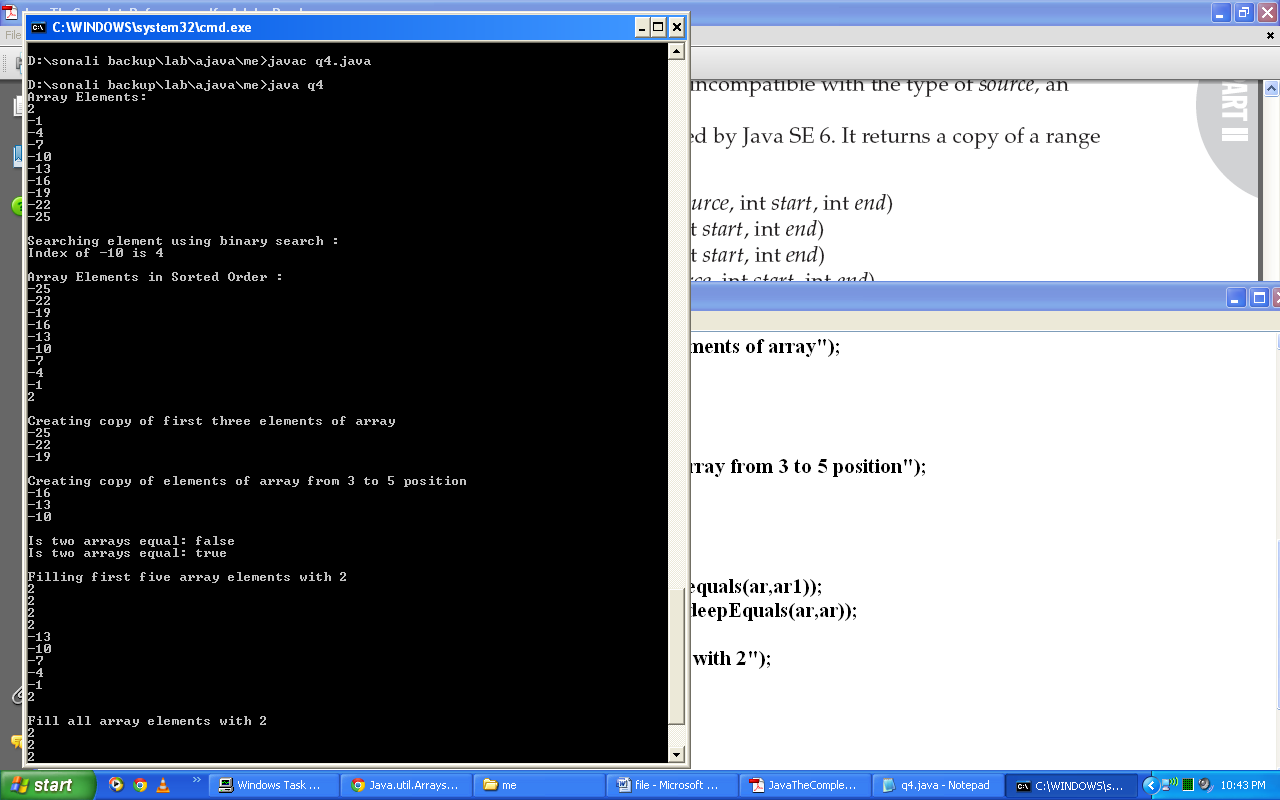
for(int i=0;i<10;i++)

System.out.println(arr[i]);

}

}

**Output:**

****

1. **Write a program to implement contains(), remove(), size(), add(), isEmpty() using HashSet.**

**Code:**

import java.util.\*;

class q5{

@SuppressWarnings("unchecked")

public static void main(String args[]){

HashSet hs1=new HashSet();

hs1.add(new Integer(56));

hs1.add(new Integer(64));

hs1.add(new Integer(21));

hs1.add(new Integer(9));

hs1.add(new Integer(10));

System.out.println("Set Elements:");

System.out.println(hs1);

System.out.println("Size of set: "+hs1.size());

System.out.println("Is Set empty: "+hs1.isEmpty());

System.out.println("Does hashset contains 10: "+hs1.contains(10));

System.out.println("After removing 2nd element:");

hs1.remove(10);

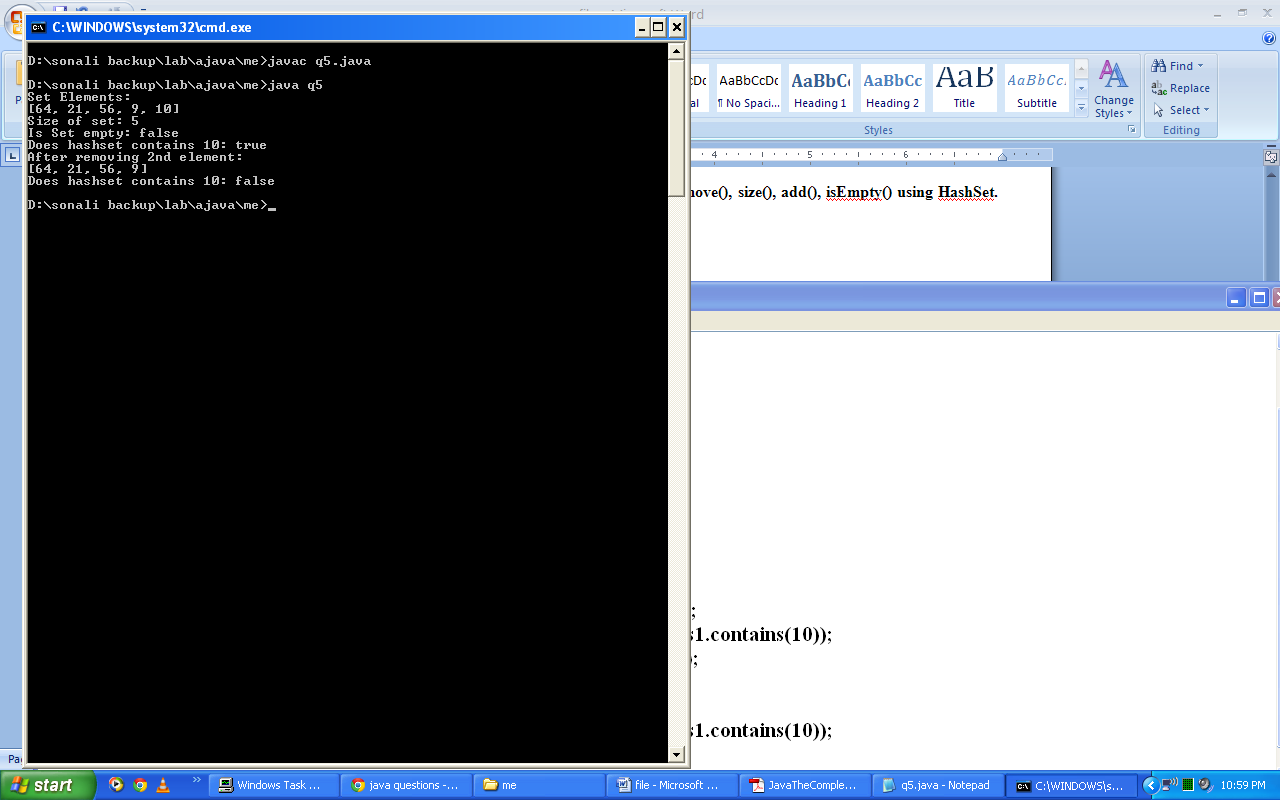
System.out.println(hs1);

System.out.println("Does hashset contains 10: "+hs1.contains(10));

}

}

**Output:**

****

1. **Write a program to implement all the functions of Collections class.**

**Code:**

import java.util.\*;

class q6{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedList l1=new LinkedList();

l1.add(new Integer(-8));

l1.add(new Integer(10));

l1.add(new Integer(90));

l1.add(new Integer(1));

l1.add(new Integer(90));

l1.add(new Integer(15));

System.out.println("List:");

System.out.println(l1);

System.out.println();

Comparator comp=Collections.reverseOrder();

Collections.sort(l1,comp);

System.out.println("List in sorted order:");

System.out.println(l1);

System.out.println();

Collections.shuffle(l1);

System.out.println("List after shuffling:");

System.out.println(l1);

System.out.println();

System.out.println("Maximum element in List: "+Collections.max(l1));

System.out.println("Minimum element in List: "+Collections.min(l1));

int index=Collections.binarySearch(l1,1);

System.out.println("Index of 1: "+index);

System.out.println();

LinkedList l2=new LinkedList();

l2.add(new Integer(56));

l2.add(new Integer(61));

l2.add(new Integer(72));

l2.add(new Integer(81));

l2.add(new Integer(45));

l2.add(new Integer(32));

l2.add(new Integer(21));

System.out.println("List 2:");

System.out.println(l2);

System.out.println("Coping list1 into list2:");

Collections.copy(l2,l1);

System.out.println(l2);

System.out.println();

System.out.println("List2 using Enumeration:");

Enumeration em=Collections.enumeration(l2);

while(em.hasMoreElements())

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

System.out.println();

System.out.println("Relacing 90 with 11 in List 2");

Collections.replaceAll(l2,90,11);

System.out.println(l2);

System.out.println();

System.out.println("Rotating List2 by factor of 2:");

Collections.rotate(l2,2);

System.out.println(l2);

System.out.println();

System.out.println("Swaping 2nd and 5th element in List 2:");

Collections.swap(l2,2,5);

System.out.println(l2);

System.out.println();

LinkedList l3=new LinkedList();

l3.add(new Integer(10));

l3.add(new Integer(21));

System.out.println("List3 : " + l3);

System.out.println();

System.out.println("index of sublist : " + Collections.indexOfSubList(l1,l2));

System.out.println("index of sublist : " + Collections.indexOfSubList(l2,l3));

System.out.println("last index of sublist : " + Collections.lastIndexOfSubList(l2,l1));

System.out.println();

System.out.println("Creating 5 copies of an element in list");

List a1 = Collections.nCopies(5, "bhawna");

Iterator it1=a1.iterator();

while(it1.hasNext())

System.out.println(it1.next());

System.out.println();

System.out.println("Filling all elements with 10");

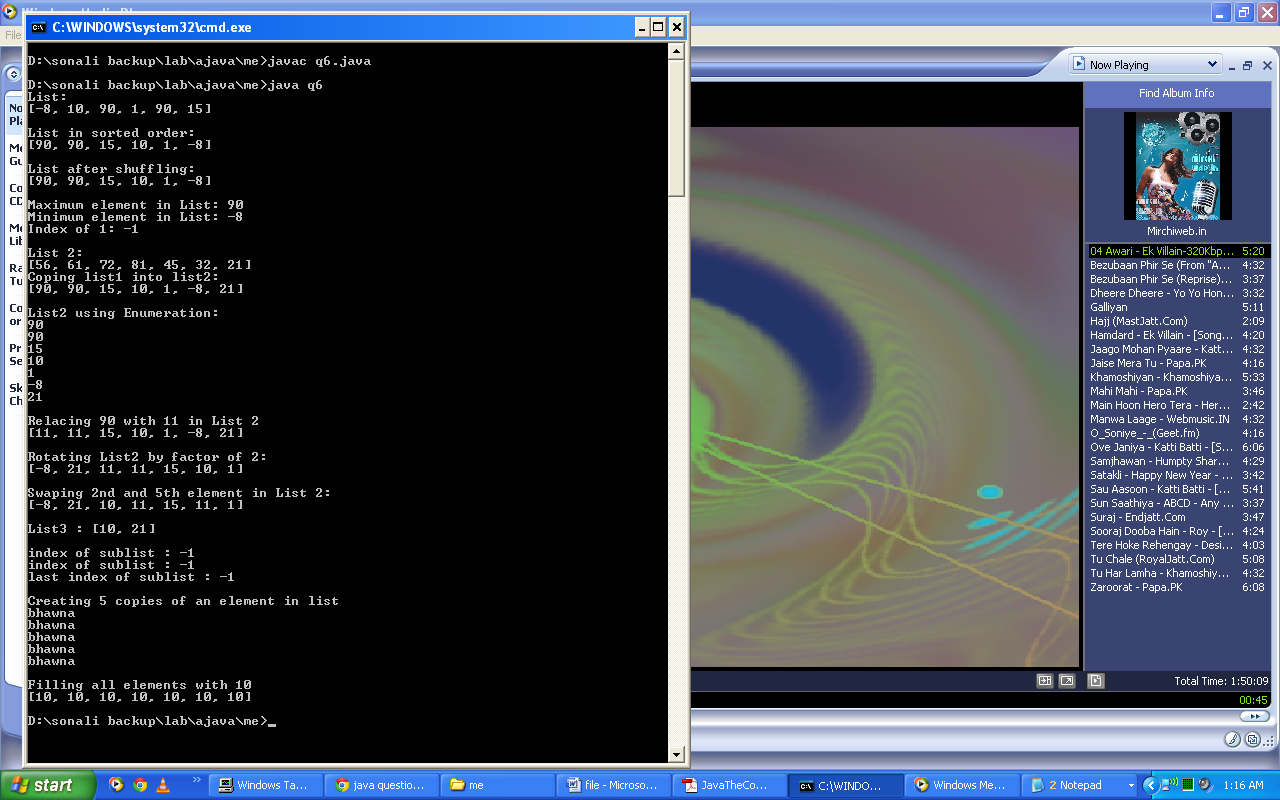
Collections.fill(l2,10);

System.out.println(l2);

}

}

**Output:**

****

1. **Write a program to create a user defined class to store students details (Roll no, Name, Marks). Print the details of every student using LinkedList.**

**Code:**

import java.util.\*;

class Student{

int roll,marks;

String name;

public Student(int r, String n, int m){

roll=r;

name=n;

marks=m;

}

public String toString(){

return "("+roll+" "+name+" "+marks+")";

}

}

class q7{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedList l1=new LinkedList();

l1.add(new Student(1,"sonali",90));

l1.add(new Student(2,"bhanu",80));

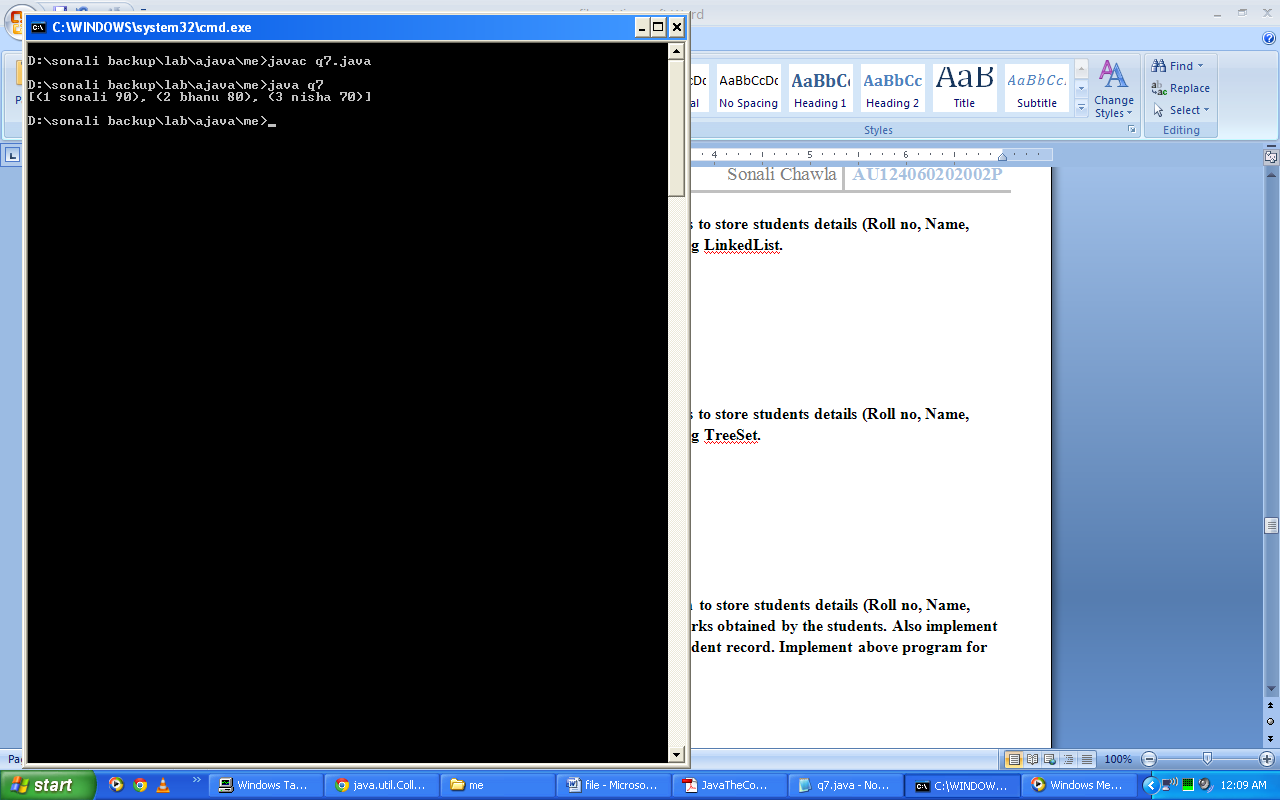
l1.add(new Student(3,"nisha",70));

System.out.println(l1);

}

}

**Output:**

****

1. **Write a program to create a user defined class to store students details (Roll no, Name, Marks). Print the details of every student using TreeSet.**

**Code:**

import java.util.\*;

class Student{

int roll,marks;

String name;

public Student(int r, String n, int m){

roll=r;

name=n;

marks=m;

}

public String toString(){return "("+roll+" "+name+" "+marks+")"; }

}

class MyComp implements Comparator{

public int compare(Object a, Object b){

Student aOb,bOb;

Integer aInt,bInt;

aOb=(Student)a; bOb=(Student)b;

aInt=aOb.roll; bInt=bOb.roll;

return aInt.compareTo(bInt);

}

}

class q8{

@SuppressWarnings("unchecked")

public static void main(String args[]){

TreeSet ts1=new TreeSet(new MyComp());

ts1.add(new Student(1,"sonali",90));

ts1.add(new Student(2,"bhanu",80));

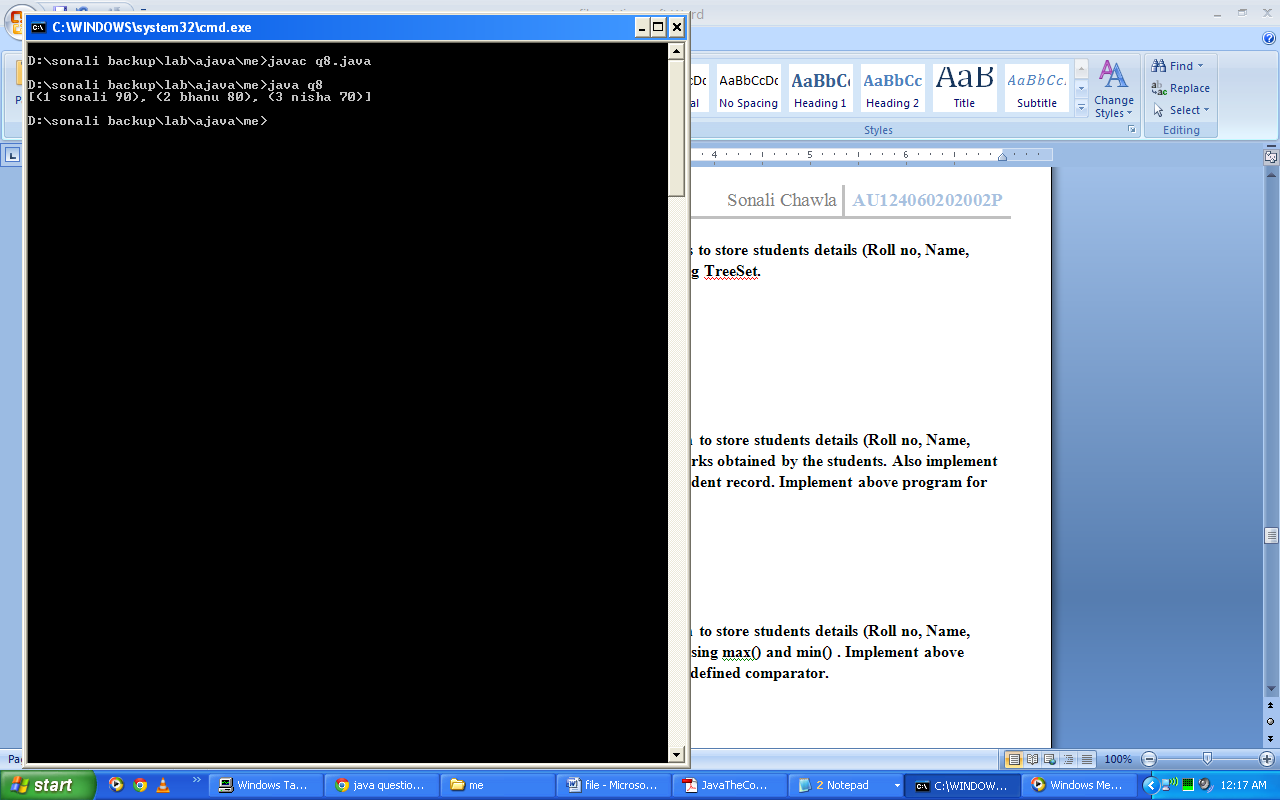
ts1.add(new Student(3,"nisha",70));

System.out.println(ts1);

}

}

**Output:**

****

1. **Write a program to create your own collection to store students details (Roll no, Name, Marks). Arrange the collection in order of marks obtained by the students. Also implement binary search function to find a particular student record. Implement above program for the sorted and unsorted collection.**

**Code:**

import java.util.\*;

class Student{

int roll,marks;

String name;

public Student(int r, String n, int m){

roll=r;

name=n;

marks=m;

}

public String toString(){

return "("+roll+" "+name+" "+marks+")";

}

}

class MyComp implements Comparator{

public int compare(Object a, Object b){

Student aOb,bOb;

Integer aInt,bInt;

aOb=(Student)a;

bOb=(Student)b;

aInt=aOb.marks;

bInt=bOb.marks;

return bInt.compareTo(aInt);

}

}

class q9{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedList l1=new LinkedList();

l1.add(new Student(1,"sonali",90));

l1.add(new Student(2,"bhanu",70));

l1.add(new Student(3,"nisha",80));

System.out.println(l1);

System.out.println("Finding record for nisha:");

Student s1=new Student(3,"nisha",70);

int index=Collections.binarySearch(l1,s1,new MyComp());

System.out.println("Nisha Found at index: "+index);

System.out.println("Sorted List");

Collections.sort(l1,new MyComp());

System.out.println(l1);

System.out.println("Finding record for nisha:");

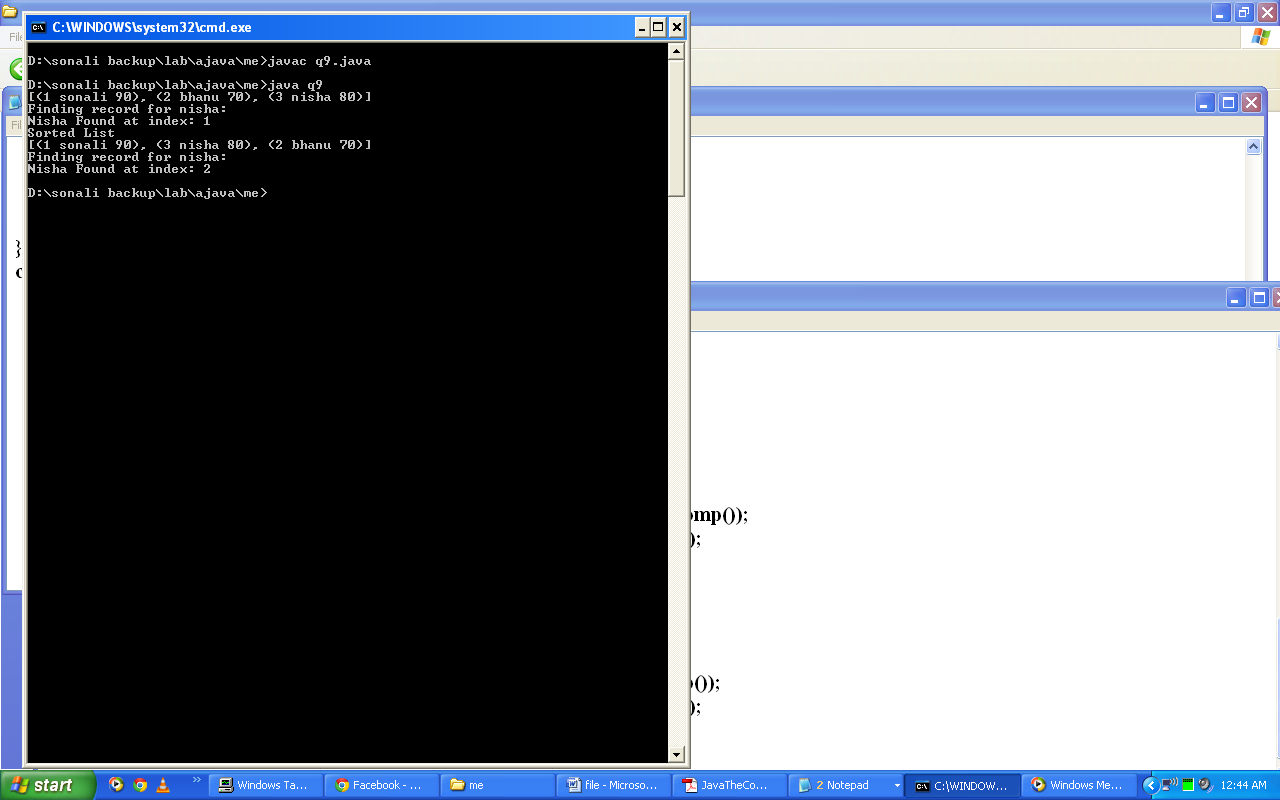
index=Collections.binarySearch(l1,s1,new MyComp());

System.out.println("Nisha Found at index: "+index);

}

}

**Output:**

****

1. **Write a program to create your own collection to store students details (Roll no, Name, Marks). Find student scoring highest and lowest marks using max() and min() . Implement above program on basis of natural ordering or user-defined comparator.**

**Code:**

import java.util.\*;

class Student{

int roll,marks;

String name;

public Student(int r, String n, int m){

roll=r; name=n; marks=m; }

public String toString(){return "("+roll+" "+name+" "+marks+")"; }

}

class MyComp implements Comparator{

public int compare(Object a, Object b){

Student aOb,bOb;

Integer aInt,bInt;

aOb=(Student)a; bOb=(Student)b;

aInt=aOb.marks; bInt=bOb.marks;

return bInt.compareTo(aInt); }

}

class q10{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedList l1=new LinkedList();

l1.add(new Student(1,"sonali",90)); l1.add(new Student(2,"bhanu",70));

l1.add(new Student(3,"nisha",80)); System.out.println(l1);

System.out.println("Maximum Marks: "+Collections.max(l1,new MyComp()));

System.out.println("Minimum Marks: "+Collections.min(l1,new MyComp()));

System.out.println("Sorted List"); Collections.sort(l1,new MyComp());

System.out.println(l1);

System.out.println("Maximum Marks: "+Collections.max(l1,new MyComp()));

System.out.println("Minimum Marks: "+Collections.min(l1,new MyComp()));

}

}

**Output:**

****

1. **Write a program to create a collection to store students details with student roll no, name and marks. Implement the reverse() , shuffle() and rotate() function on the above collection.**

**Code:**

import java.util.\*;

class Student{

int roll,marks;

String name;

public Student(int r,String n,int m){

roll=r;

name=n;

marks=m;

}

public String toString(){return "("+roll+", "+name+", "+marks+")";}

}

class q11{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedList l1=new LinkedList();

l1.add(new Student(1,"bhanu",80));

l1.add(new Student(2,"sonali",90));

l1.add(new Student(3,"nisha",70));

System.out.println("List: "+l1);

Collections.reverse(l1);

System.out.println("Reverse: "+l1);

Collections.rotate(l1,2);

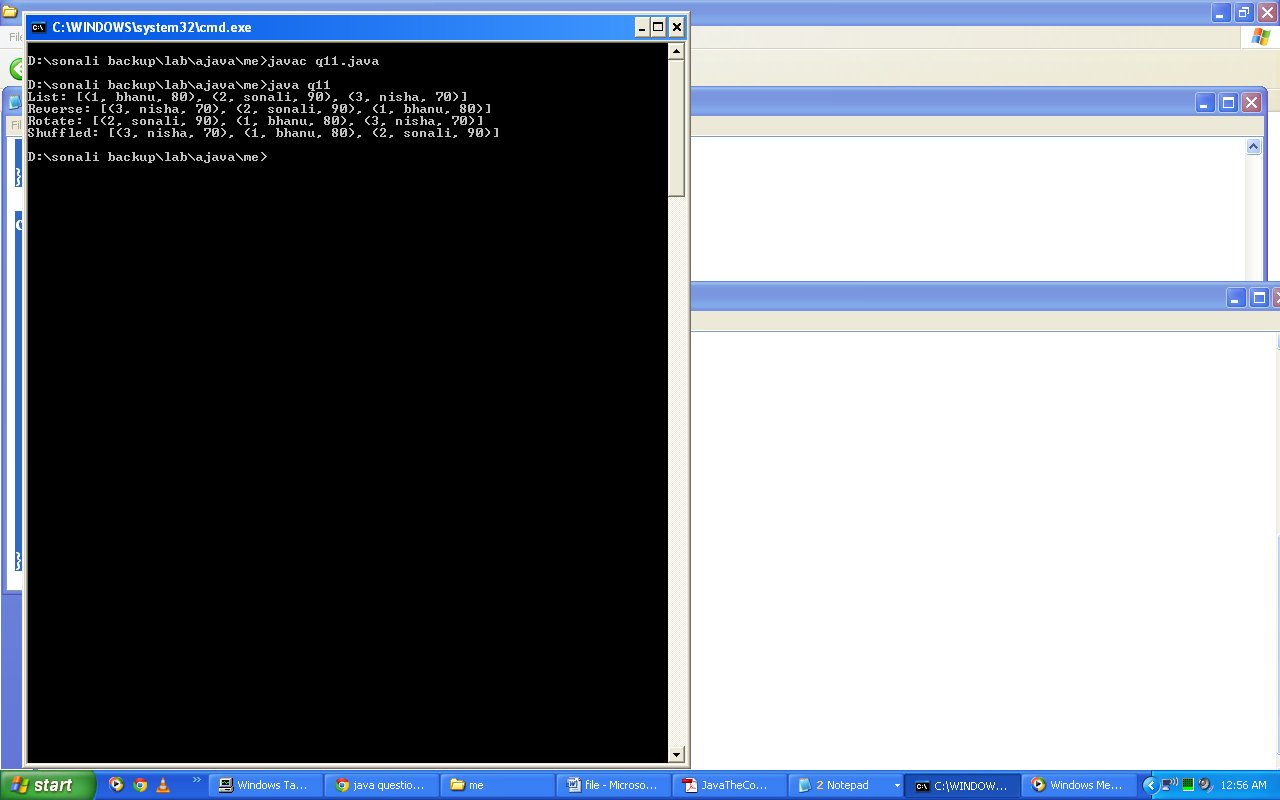
System.out.println("Rotate: "+l1);

Collections.shuffle(l1);

System.out.println("Shuffled: "+l1); }

}

**Output:**

****

1. **Write a program to copy one vector to another vector using Vector class. Create a clone of the vector. Also add six more elements to the clone vector. Check size and capacity before and after adding elements. Create an arraylist with 4 elements and check for them in Vector formed.**

**Code:**

import java.util.\*;

class q12{

@SuppressWarnings("unchecked")

public static void main(String args[]){

Vector v1=new Vector();

v1.add("orange"); v1.add("apple");

v1.add("banana"); v1.add("grapes");

System.out.println("Vector: "+v1);

String[] s1= new String[v1.size()]; v1.copyInto(s1);

System.out.println("Copied array : " );

for (String s2:s1)

System.out.println(s2);

Vector v2=(Vector)v1.clone();

System.out.println("Clone of Vector: "+v2);

System.out.println("Size of Clone Vector: "+v2.size());

System.out.println("Capacity of CloneVector: "+v2.capacity());

v2.add("guava"); v2.add("pineapple");

v2.add("leechi"); v2.add("gooseberry");

v2.add("raspberry"); v2.add("cheery");

System.out.println("Size of Clone Vector: "+v2.size());

System.out.println("Capacity of CloneVector: "+v2.capacity());

ArrayList ar=new ArrayList(); ar.add("guava");

ar.add("apple"); System.out.println("List: "+ar);

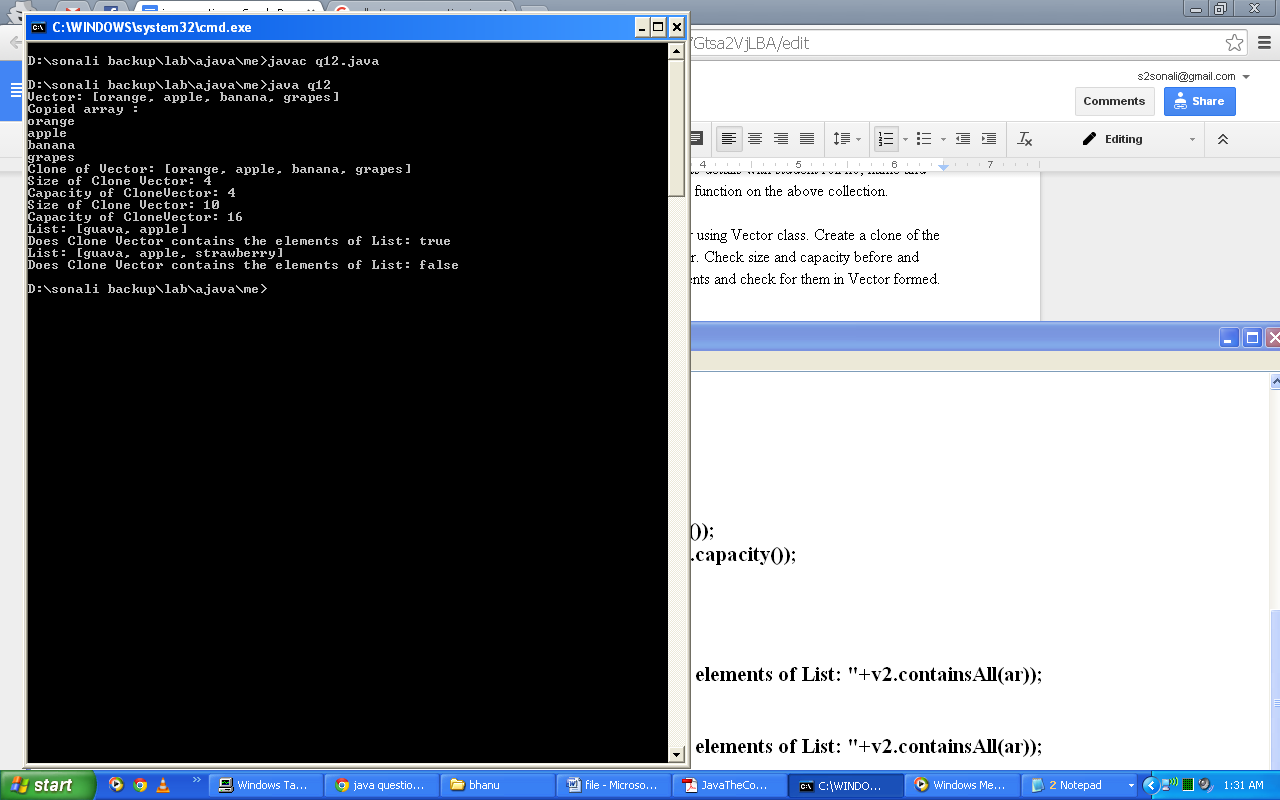
System.out.println("Does Clone Vector contains the elements of List: "+v2.containsAll(ar));

ar.add("strawberry"); System.out.println("List: "+ar);

System.out.println("Does Clone Vector contains the elements of List: "+v2.containsAll(ar)); }

}

**Output:**

****

1. **Write a program to implement is.Empty(), firstElement(), lastElement(), elementAt(), indexOf(), lastIndexOf(), insertElementAt(), removeElementAt(),removeAllElements(), setSize() and ensureCapacity() using Vector class.**

**Code:**

import java.util.\*;

class q13{

public static void main(String args[]){

Vector v1=new Vector();

System.out.println("Is vector empty: "+v1.isEmpty());

v1.add(new Integer(67)); v1.add(new Integer(23));

v1.add(new Integer(12));v1.add(new Integer(90));v1.add(new Integer(12));

System.out.println("Is vector empty: "+v1.isEmpty());

System.out.println("First Element of Vector: "+v1.firstElement());

System.out.println("Last Element of Vector: "+v1.lastElement());

System.out.println("Element at 2nd position: "+v1.elementAt(2));

System.out.println("Index of 23: "+v1.indexOf(23));

System.out.println("Last Index of 12: "+v1.lastIndexOf(12));

v1.insertElementAt(25,2);

System.out.println("Vector after adding new element at 2nd position: "+v1);

v1.removeElementAt(3);

System.out.println("Vector after removing element at 3rd position: "+v1);

System.out.println("Size of Vector: "+v1.size());

System.out.println("Capacity of Vector: "+v1.capacity());

v1.setSize(6); v1.ensureCapacity(6);

System.out.println("After setting size and capacity of vector:");

System.out.println("Size of Vector: "+v1.size());

System.out.println("Capacity of Vector: "+v1.capacity());

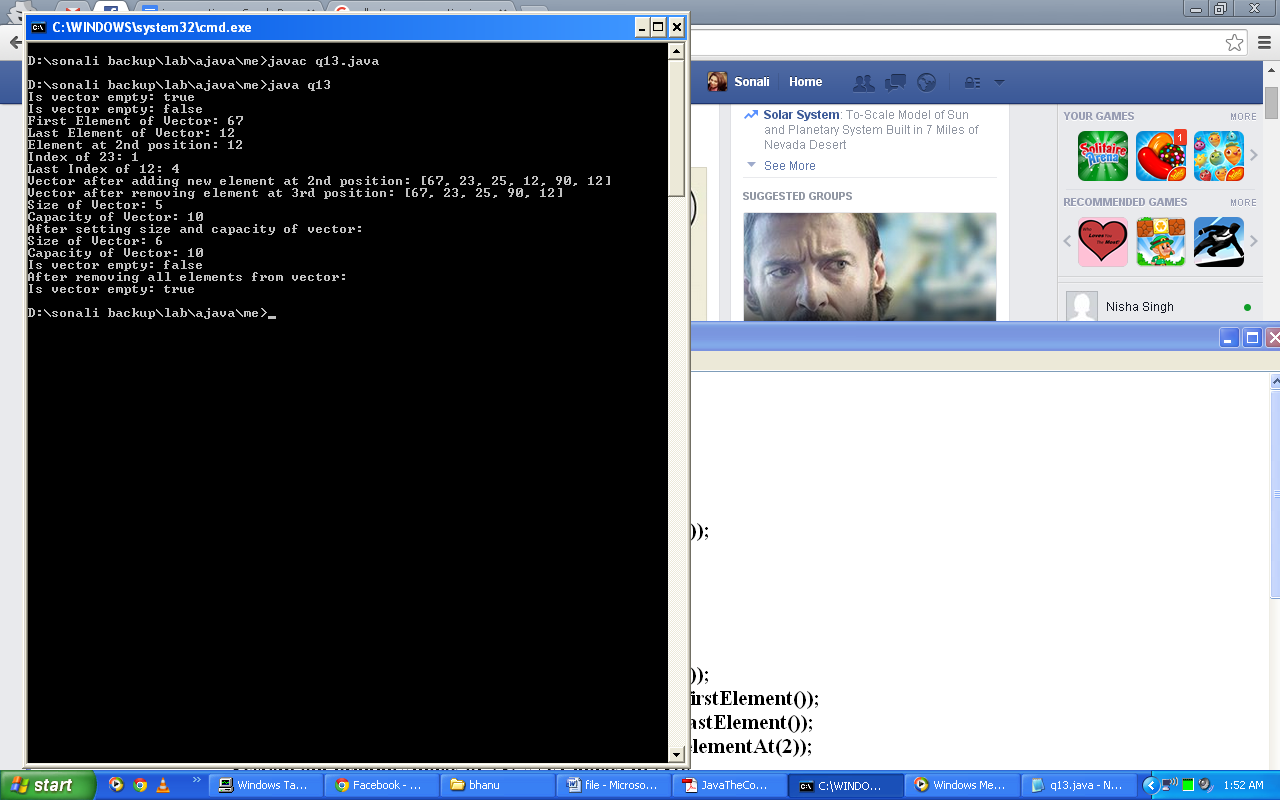
System.out.println("Is vector empty: "+v1.isEmpty());v1.removeAllElements();

System.out.println("After removing all elements from vector: ");

System.out.println("Is vector empty: "+v1.isEmpty()); }

}

**Output:**

****

1. **Write a program to copy one stack to another using Stack class. Create a clone of the stack. Add three elements to the stack and implement search operation on it. Remove two elements from stack.  Also check for size and capacity of the stack before and after removing elements. Print all the elements with the help of iterator() and print the topmost element in stack.**

**Code:**

import java.util.\*;

class q14{

@SuppressWarnings("unchecked")

public static void main(String args[]){

Stack s1=new Stack();

s1.push(new Integer(34));

s1.push(new Integer(50));

s1.push(new Integer(12));

s1.push(new Integer(5));

System.out.println("Stack1: "+s1);

Integer[] ar=new Integer[s1.size()];

s1.copyInto(ar);

System.out.println("Copy of Stack: ");

for(int p: ar)

System.out.println(p);

Stack s2=(Stack)s1.clone();

System.out.println("Clone of Stack: "+s2);

s2.push(new Integer(55));

s2.push(new Integer(60));

s2.push(new Integer(77));

System.out.println("Clone of Stack after adding new elements: "+s2);

System.out.println("Searching element 12: "+s2.search(12));

System.out.println("Size of Clone: "+s2.size());

System.out.println("Capacity of Clone: "+s2.capacity());

s2.pop();

s2.pop();

System.out.println("Clone of Stack after removing 2 elements: "+s2);

System.out.println("Size of Clone: "+s2.size());

System.out.println("Capacity of Clone: "+s2.capacity());

System.out.println("Visiting Elements using Iterator: ");

Iterator it1=s2.iterator();

while(it1.hasNext())

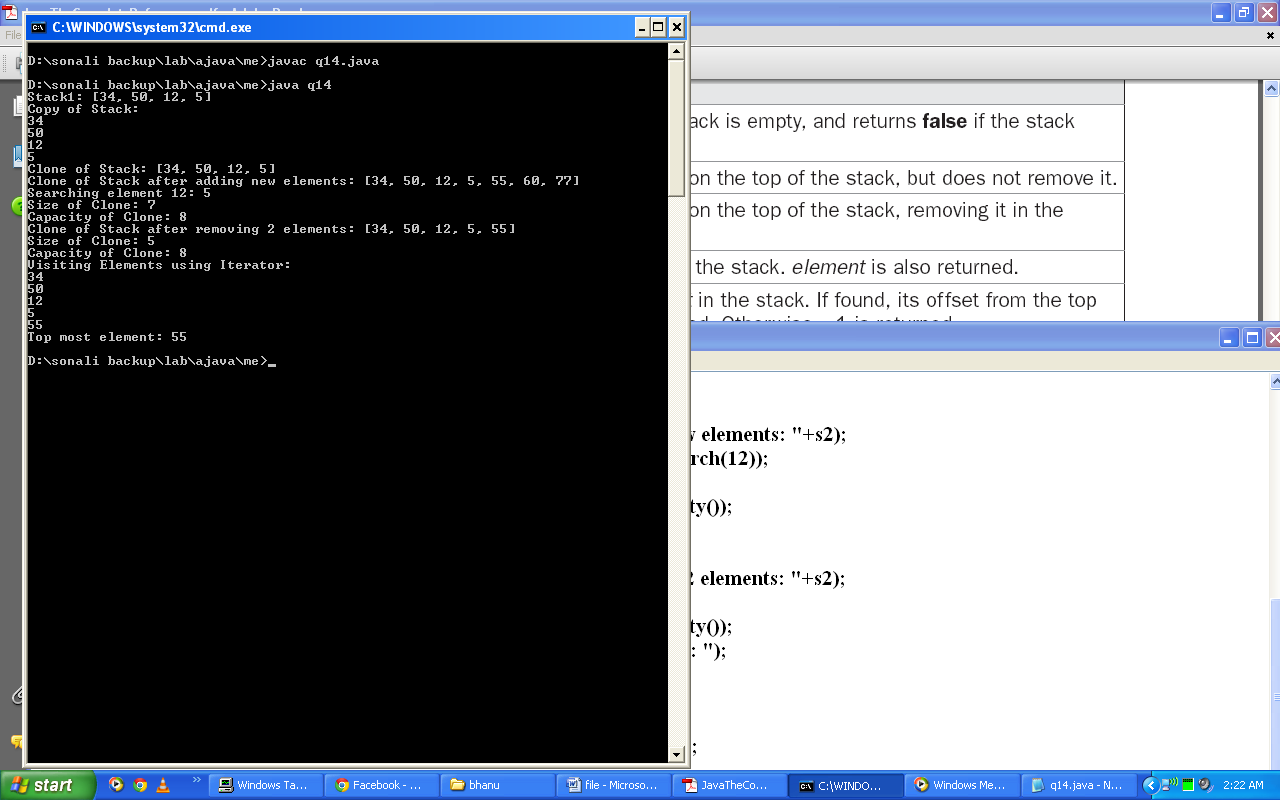
System.out.println(it1.next());

System.out.println("Top most element: "+s2.peek());

}

}

**Output:**

****

1. **Write a program to implement isEmpty(),firstElement(),lastElement(),elementAt(), indexOf(), removeElement(),setElementAt(),setSize(), insertElementAt() and trimToSize() using Stack class.**

**Code:**

import java.util.\*;

class q15{

@SuppressWarnings("unchecked")

public static void main(String args[]){

Stack s1=new Stack();

System.out.println("Is Stack empty: "+s1.isEmpty());

s1.push(new Integer(34)); s1.push(new Integer(50));

s1.push(new Integer(12)); s1.push(new Integer(5));

System.out.println("Stack1: "+s1);

System.out.println("Is Stack empty: "+s1.isEmpty());

System.out.println("First Element of Stack1: "+s1.firstElement());

System.out.println("Last Element of Stack1: "+s1.lastElement());

System.out.println("Element at 1st position: "+s1.elementAt(1));

System.out.println("Index of 5: "+s1.indexOf(5));

s1.setElementAt(10,0);

System.out.println("Stack after setting 10 at 0: "+s1);

s1.insertElementAt(56,3);

System.out.println("Stack after adding 56 at 3rd position: "+s1);

System.out.println("Size of Stack: "+s1.size());

System.out.println("Capacity of Stack: "+s1.capacity());

s1.setSize(4); s1.trimToSize();

System.out.println("After setting size and capacity:");

System.out.println("Size of Stack: "+s1.size());

System.out.println("Capacity of Stack: "+s1.capacity());

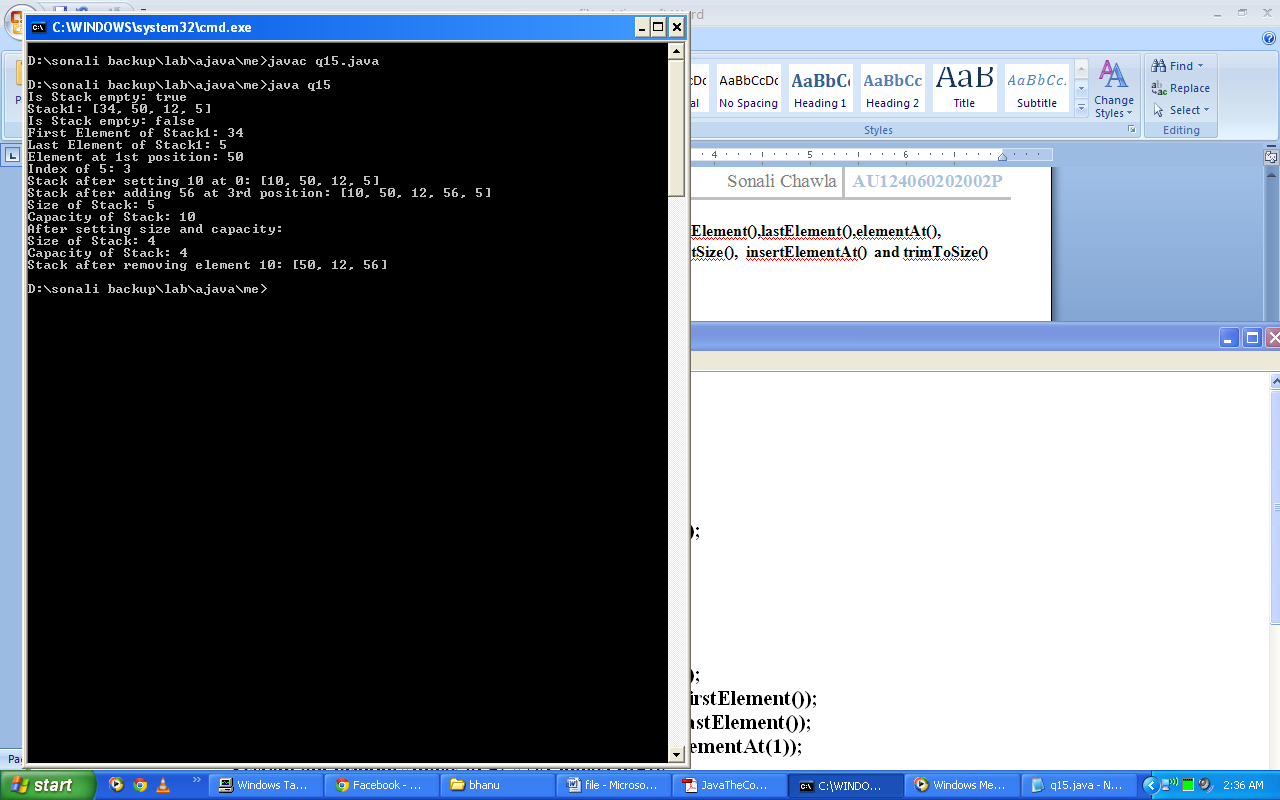
s1.removeElement(10);

System.out.println("Stack after removing element 10: "+s1);

}

}

**Output:**

****

1. **Write a program to implement put(), putAll(), getKey(), getValue(), size(), isEmpty(), Map.Entry(), clear(), containsKey(), containsValue(). Take two hashmaps and prove whether they are equal or not.**

**Code:**

import java.util.\*;

class q16{

public static void main(String args[]){

HashMap hm=new HashMap(); HashMap hm2=new HashMap();

System.out.println("Is Hashmap empty: "+hm.isEmpty());

hm.put(1,"hello");hm.put(2,"hi");hm.put(3,"Bye"); hm.put(4,"Good");

hm.put(5,"Bad"); System.out.println("Size of Hashmap:"+hm.size());

System.out.println("Is Hashmap empty: "+hm.isEmpty());

Set<Map.Entry> set=hm.entrySet(); System.out.println("Hashmap:");

for(Map.Entry me:set)

System.out.println(me.getKey()+": "+me.getValue());

System.out.println();

System.out.println("Putting all elements of Hashmap to another hashmap:");

hm2.putAll(hm); System.out.println(); System.out.println("Hashmap2:");

set=hm2.entrySet();

for(Map.Entry me:set)

System.out.println(me.getKey()+": "+me.getValue());

System.out.println();

System.out.println("Does hashmap contain 2 as key: "+hm.containsKey(2));

System.out.println("Does hashmap contain 'helo' as Value: "+hm.containsValue("hello"));

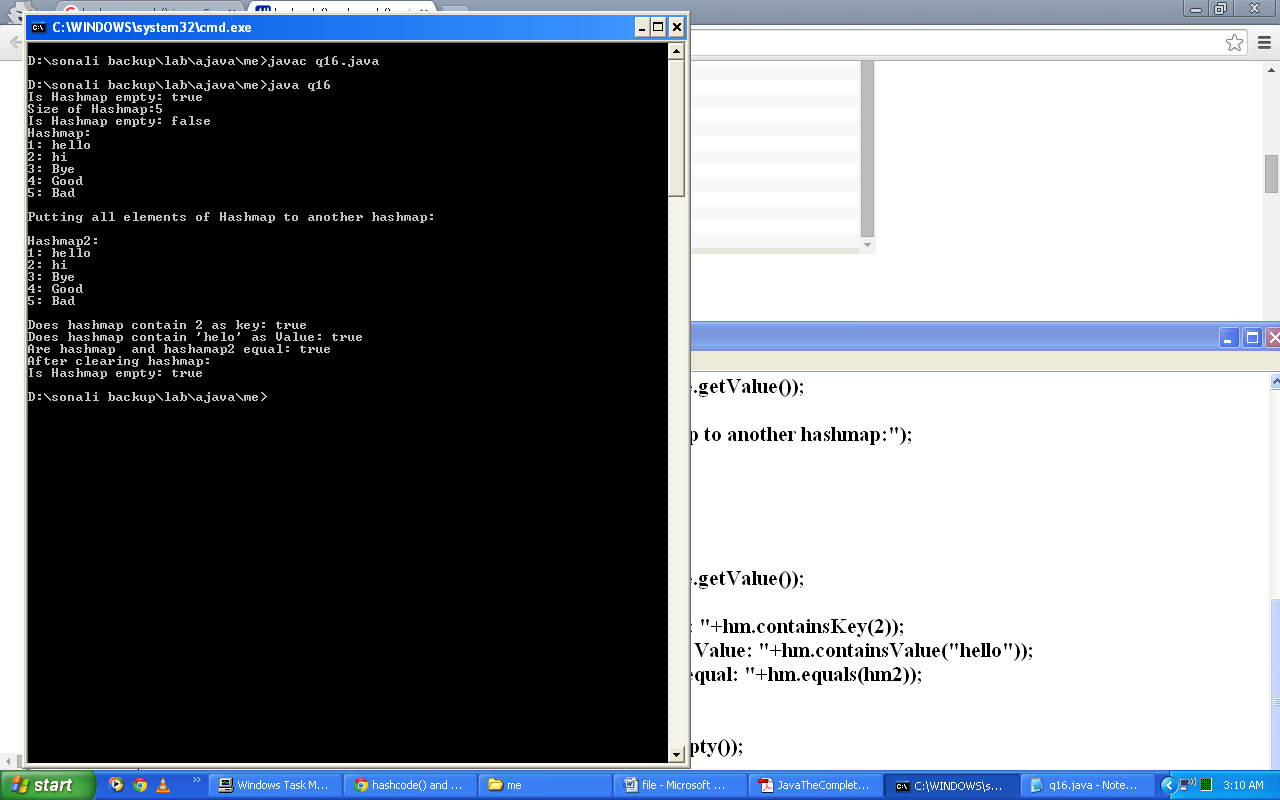
System.out.println("Are hashmap and hashamap2 equal: "+hm.equals(hm2));

hm.clear(); System.out.println("After clearing hashmap:");

System.out.println("Is Hashmap empty: "+hm.isEmpty()); }

}

**Output:**

****

1. **Write a program to create a hashmap and implement entrySet(). Also try to modify the value of a key.**

**Code:**

import java.util.\*;

class q17{

@SuppressWarnings("unchecked")

public static void main(String args[]){

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

hm.put("John",new Integer(333));

hm.put("Tom",new Integer(111));

hm.put("Jane",new Integer(222));

hm.put("Tod",new Integer(555));

hm.put("Ralph",new Integer(-1000));

Set<Map.Entry<String,Integer>> set=hm.entrySet();

for(Map.Entry<String,Integer> me:set)

System.out.println(me.getKey()+": "+me.getValue());

System.out.println();

int balance=hm.get("John");

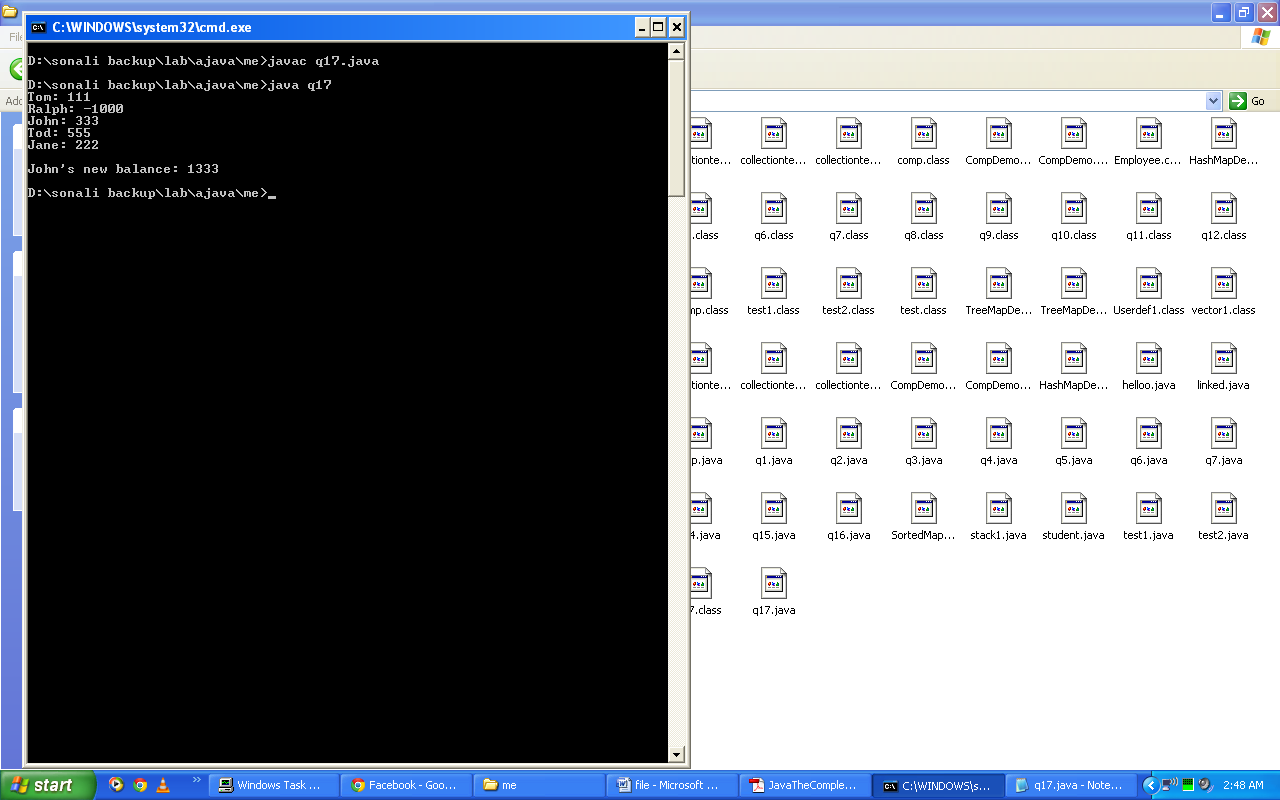
hm.put("John",balance+1000);

System.out.println("John's new balance: "+ hm.get("John"));

}

}

**Output:**

****

1. **Write a program to implement headMap(), lastKey(), subMap(), tailMap() with object as argument using sorted maps.**

**Code:**

import java.util.\*;

class q18{

@SuppressWarnings("unchecked")

public static void main(String args[]){

SortedMap sm1=new TreeMap();

sm1.put("A",new Integer(34));

sm1.put("B",new Integer(23));

sm1.put("C",new Integer(15));

sm1.put("D",new Integer(30));

sm1.put("E",new Integer(89));

System.out.println("SortedMap:");

System.out.println(sm1);

System.out.println("Head Map: "+sm1.headMap("D"));

System.out.println("Last Key of SortedMap: "+sm1.lastKey());

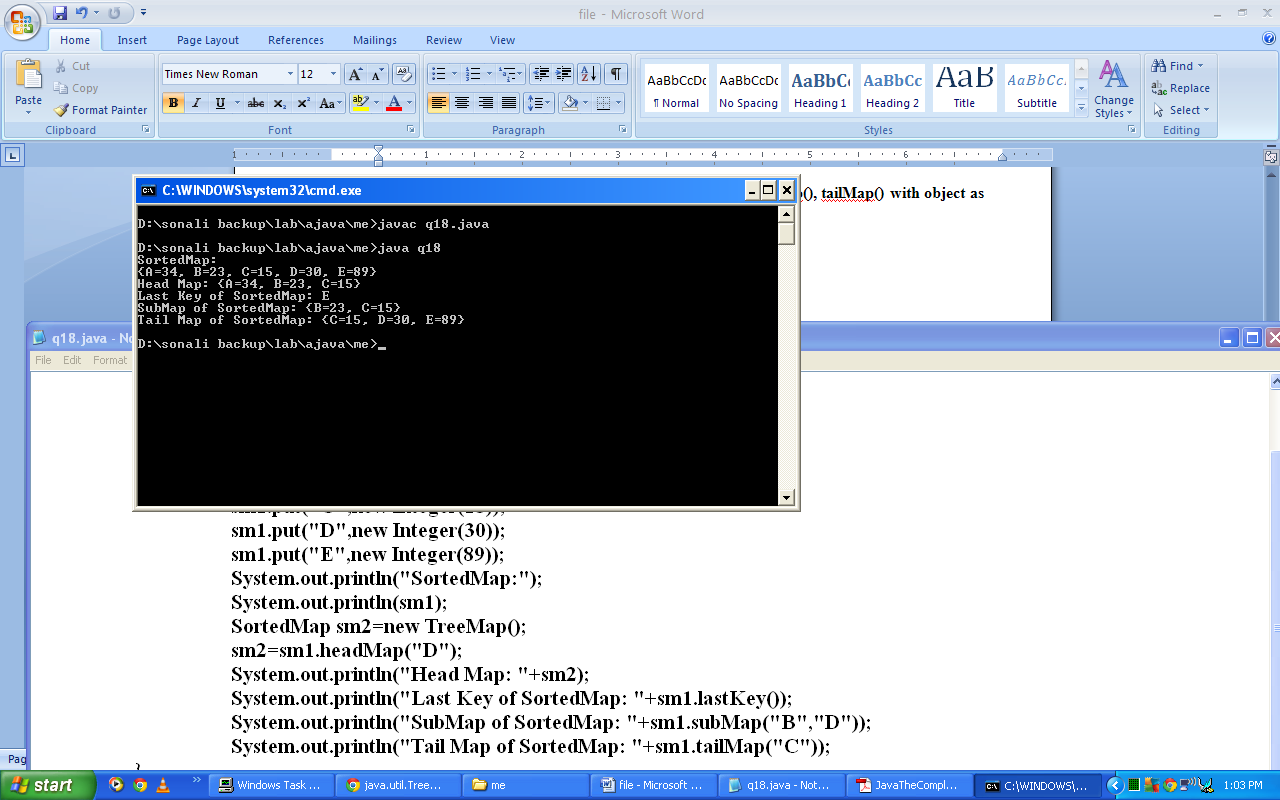
System.out.println("SubMap of SortedMap: "+sm1.subMap("B","D"));

System.out.println("Tail Map of SortedMap: "+sm1.tailMap("C"));

}

}

**Output:**

****

1. **Write a program to implement comparator in TreeMap.**

**Code:**

import java.util.\*;

class q19{

@SuppressWarnings("unchecked")

public static void main(String args[]){

TreeMap sm1=new TreeMap();

sm1.put("A",new Integer(34));

sm1.put("B",new Integer(23));

sm1.put("C",new Integer(15));

sm1.put("D",new Integer(30));

sm1.put("E",new Integer(89));

System.out.println("TreeMap: "+sm1);

System.out.println("Ceiling entry for 'D': "+sm1.ceilingEntry("D"));

System.out.println("Ceiling Key for 'E': "+sm1.ceilingKey("F"));

System.out.println("Size of TreeMap: "+sm1.size());

TreeMap sm2=(TreeMap)sm1.clone();

System.out.println("Clone of TreeMap: ");

Set<Map.Entry> set=sm2.entrySet();

for(Map.Entry me:set)

System.out.println(me.getKey()+": "+me.getValue());

System.out.println("Does Map contains B as key: "+sm2.containsKey("B"));

System.out.println("Does Map contains 51 as value: "+sm2.containsValue(51));

System.out.println("First Entry: "+sm2.firstEntry());

System.out.println("Last Entry: "+sm2.lastEntry());

System.out.println("Value of key C: "+sm2.get("C"));

sm2.remove("A");

System.out.println("Clone Map after removing A: "+sm2);

System.out.println("Is Map empty: "+sm1.isEmpty());

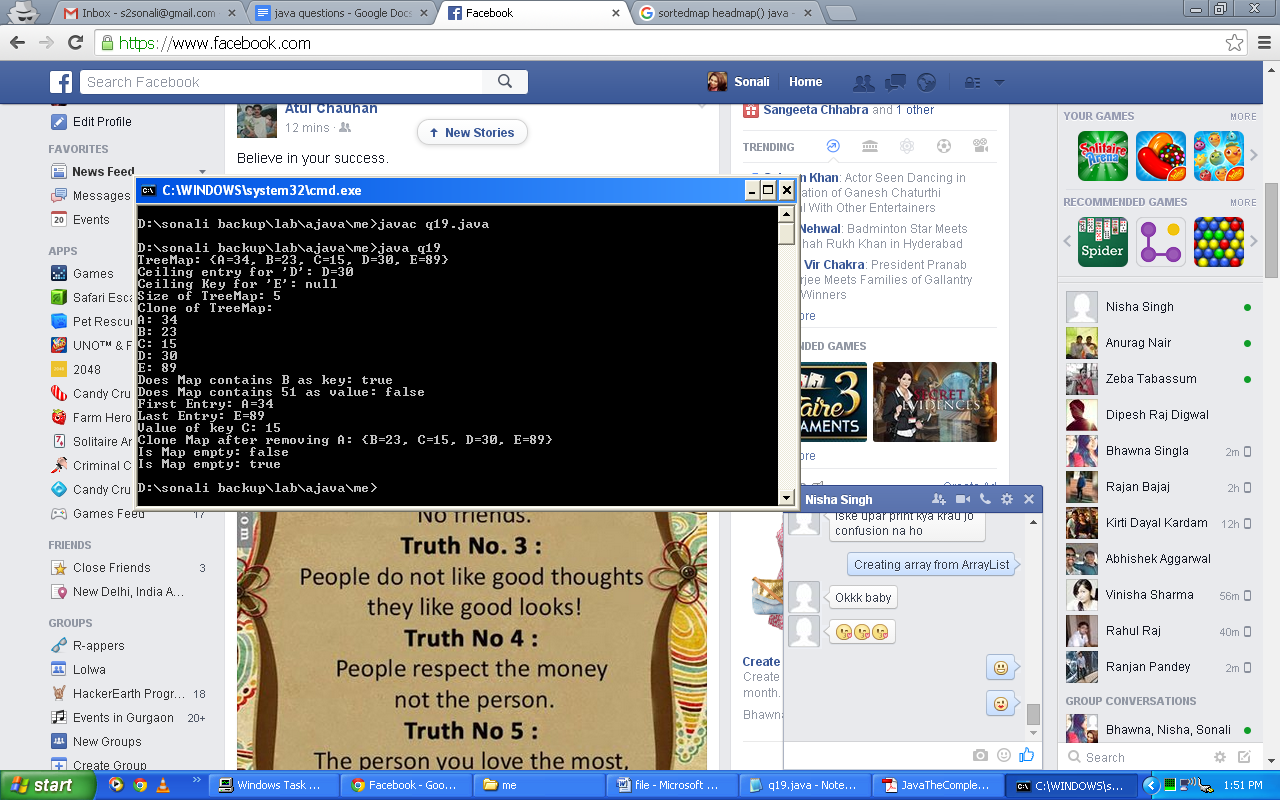
sm1.clear();

System.out.println("Is Map empty: "+sm1.isEmpty());

}

}

**Output:**

****

1. **Write a program to implement comparator in TreeSet.**

**Code:**

import java.util.\*;

class q20{

@SuppressWarnings("unchecked")

public static void main(String args[]){

TreeSet ts1=new TreeSet();

ts1.add("A"); ts1.add("B");

ts1.add("C"); ts1.add("D");

ts1.add("E");

System.out.println("TreeSet: "+ts1);

System.out.println("Size of TreeSet: "+ts1.size());

System.out.println("First Element of TreeSet: "+ts1.first());

System.out.println("Last Element of TreeSet: "+ts1.last());

TreeSet ts2=(TreeSet)ts1.clone();

System.out.println("TreeSet2: "+ts2);

System.out.println("Visiting TreeSet2 using Iterator: ");

Iterator it1=ts2.iterator();

while(it1.hasNext())

System.out.println(it1.next());

ts2.remove("E");

System.out.println("TreeSet2 after remoing 'E': "+ts2);

System.out.println("SubSet of TreeSet2: "+ts2.subSet("B","D"));

System.out.println("TailSet of TreeSet2: "+ts2.tailSet("C"));

System.out.println("Is TreeSet empty: "+ts1.isEmpty());

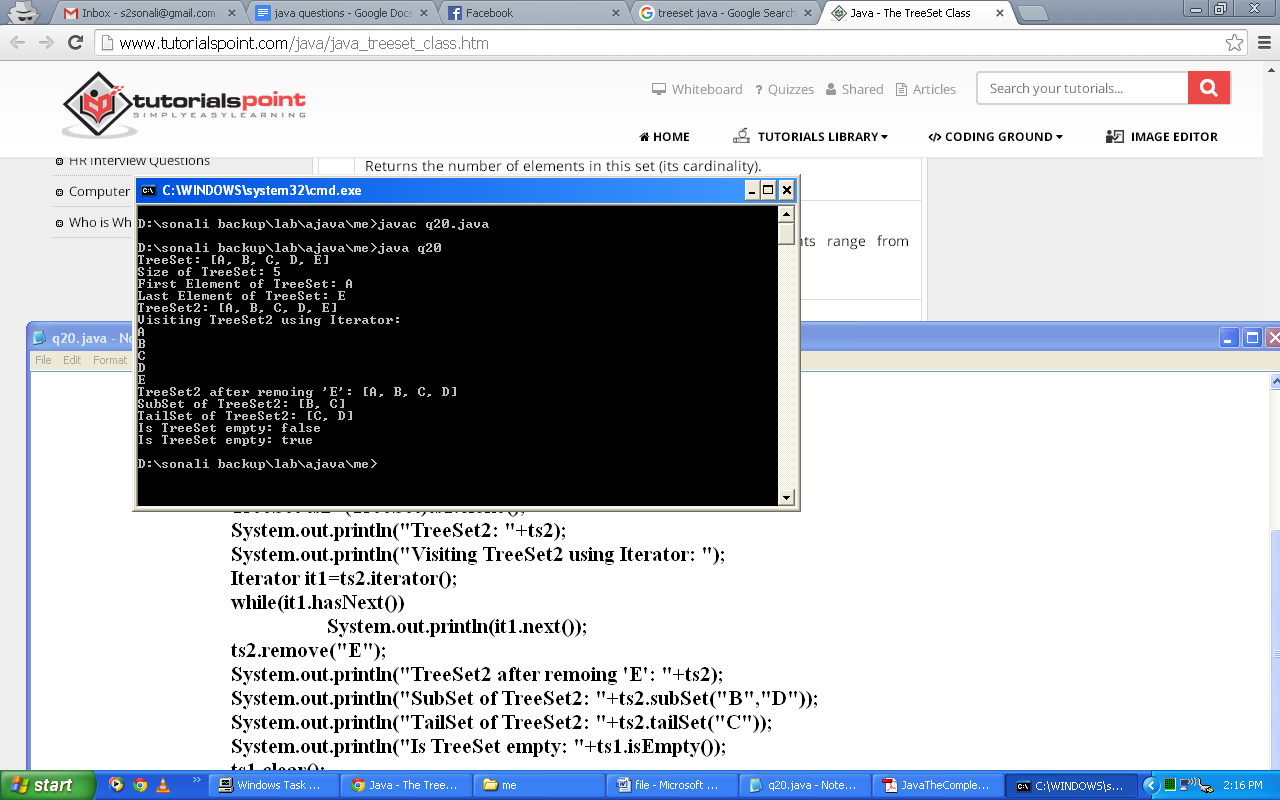
ts1.clear();

System.out.println("Is TreeSet empty: "+ts1.isEmpty());

}

}

**Output:**

****

1. **Write a program to implement the methods of LinkedHashMap.**

**Code:**

import java.util.\*;

class q21{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedHashMap lhm1=new LinkedHashMap();

lhm1.put("A",new Double(343.34));

lhm1.put("B",new Double(121.12));

lhm1.put("C",new Double(565.56));

lhm1.put("D",new Double(898.89));

lhm1.put("E",new Double(474.47));

System.out.println("LinkedHashMap: "+lhm1);

System.out.println("Does LinkedHashMap contains B as key: "+lhm1.containsKey("B"));

System.out.println("Value of D: "+lhm1.get("D"));

System.out.println("Is TreeSet empty: "+lhm1.isEmpty());

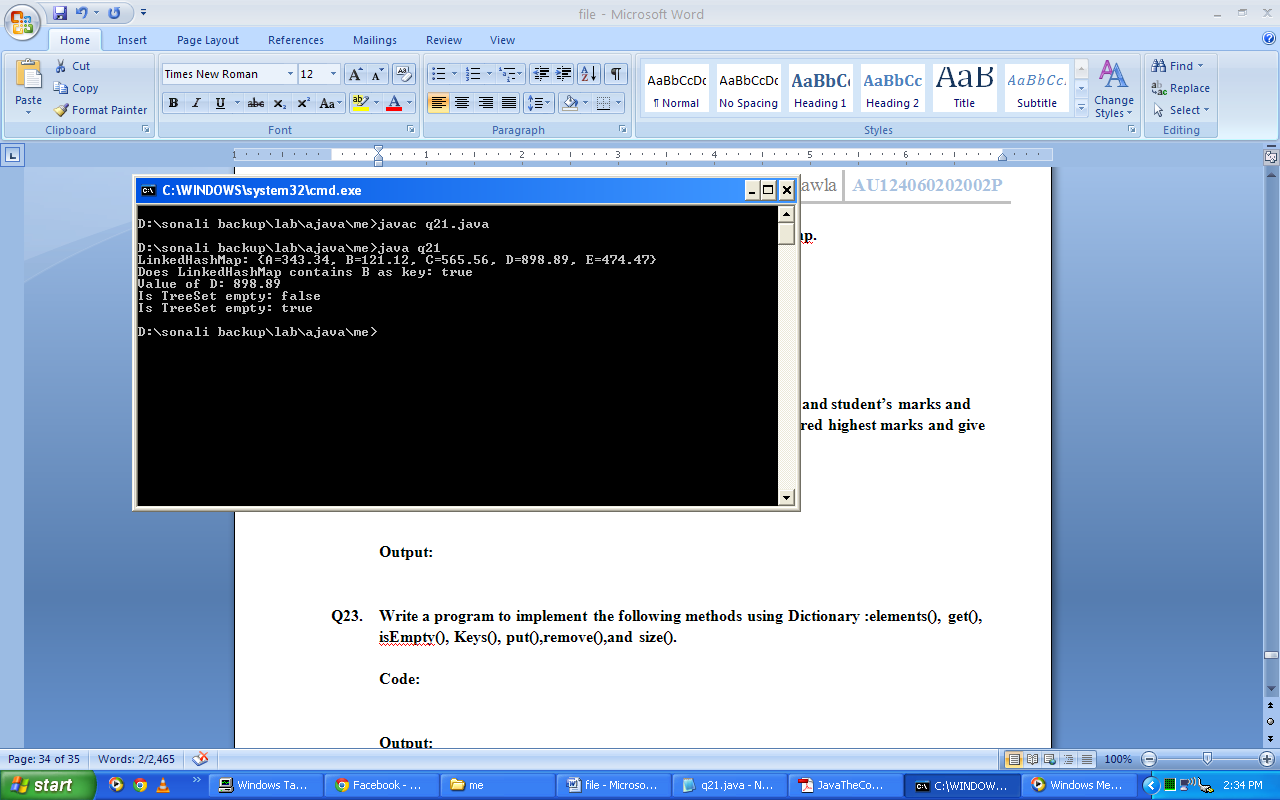
lhm1.clear();

System.out.println("Is TreeSet empty: "+lhm1.isEmpty());

}

}

**Output:**

****

1. **Write a program to create hashtable to store student’s roll no and student’s marks and revisit each element of hashtable. Find student details that scored highest marks and give him bonus of 10 marks additional and update in hashtable.**

**Code:**

import java.util.\*;

class q22

{

@SuppressWarnings("unchecked")

public static void main(String args[]){

Hashtable h1=new Hashtable();

h1.put(1,60); h1.put(2,70);

h1.put(3,80); int max=0;

Object oMax=-1;

System.out.println("Score:");

Enumeration e=h1.elements();

while(e.hasMoreElements())

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

Enumeration e1=h1.keys();

while(e1.hasMoreElements()) {

Object ob=e1.nextElement();

Object ob1=h1.get(ob);

int obInt=(((Integer)ob1).intValue());

if(obInt>max){

max=obInt;

oMax=ob; }

}

h1.put(oMax,max+10);

System.out.println();

System.out.println("Updated Score:");

Enumeration e2=h1.elements();

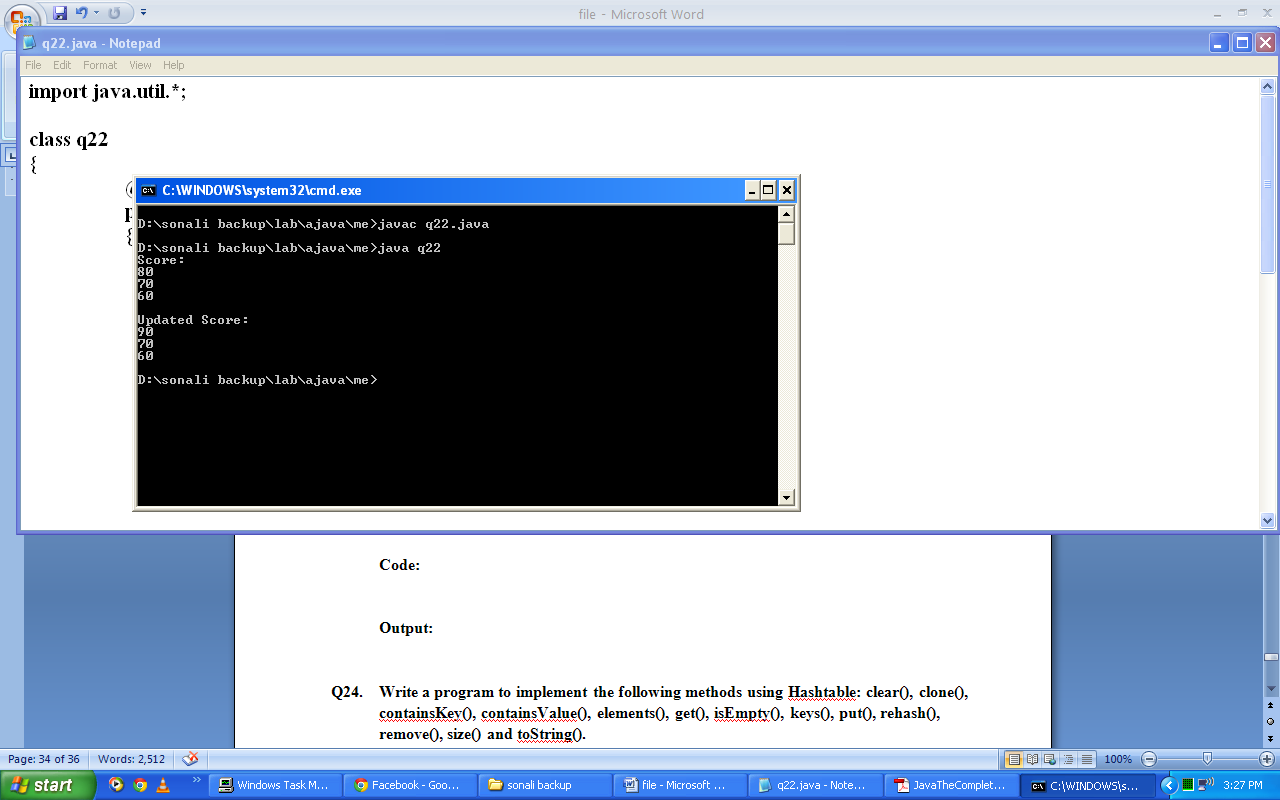
while(e2.hasMoreElements())

System.out.println(e2.nextElement());

}

}

**Output:**

****

1. **Write a program to implement the following methods using Dictionary: elements(), get(), isEmpty(), keys(), put(),remove(),and size().**

**Code:**

import java.util.\*;

class q23

{

@SuppressWarnings("unchecked")

public static void main(String args[]) {

Dictionary d1=new Hashtable();

d1.put("A", new Double(3333.333));

d1.put("B", new Double(6666.666));

d1.put("C", new Double(5555.555));

d1.put("D", new Double(7777.777));

System.out.println("Dictionary list: " + d1);

System.out.println("Is dictionary Empty: " + d1.isEmpty());

System.out.println("Size of dictionary: " + d1.size());

System.out.println("Value of 'C': " + d1.get("C"));

d1.remove("B");

System.out.println("Dictionary after removing element: " +d1);

System.out.println("Visiting elements using enumertion: ");

Enumeration e1=d1.elements();

while(e1.hasMoreElements())

System.out.println(e1.nextElement());

System.out.println("Visiting keys using enumertion: ");

Enumeration e2=d1.keys();

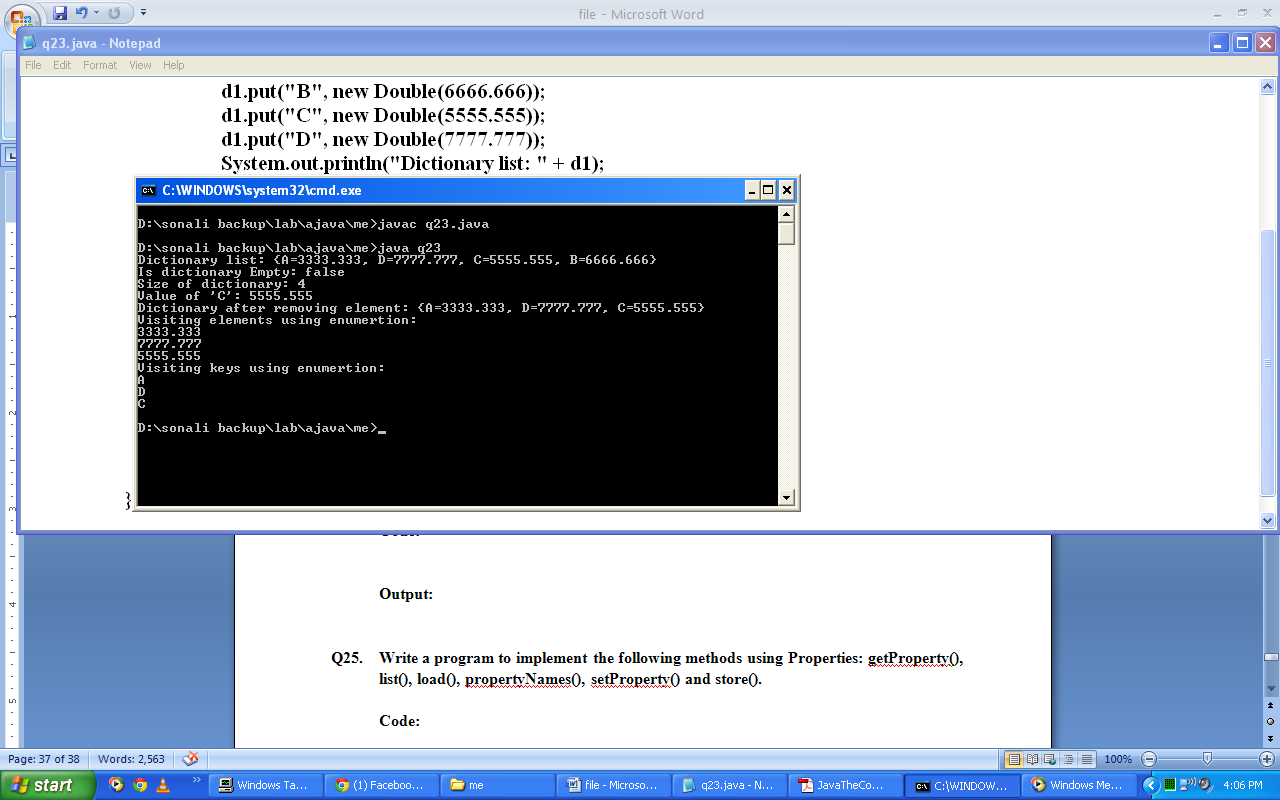
while(e2.hasMoreElements())

System.out.println(e2.nextElement());

}

}

**Output:**

****

1. **Write a program to implement the following methods using Hashtable: clear(), clone(), containsKey(), containsValue(), elements(), get(), isEmpty(), keys(), put(), rehash(), remove(), size() and toString().**

**Code:**

import java.util.\*;

class q24

{

@SuppressWarnings("unchecked")

public static void main(String args[]) {

Hashtable ht1=new Hashtable();

ht1.put("A", new Double(3333.333));ht1.put("B", new Double(6666.666));

ht1.put("C", new Double(5555.555));ht1.put("D", new Double(7777.777));

System.out.println("HashTable: " + ht1.toString());

System.out.println("Is hashtable Empty: " + ht1.isEmpty());

System.out.println("Size of hashtable: " + ht1.size());

System.out.println("Value of 'C': " + ht1.get("C"));

ht1.remove("B");System.out.println("Hashtable after removing element: " +ht1);

System.out.println("Visiting elements using enumertion: ");

Enumeration e1=ht1.elements();

while(e1.hasMoreElements())

System.out.println(e1.nextElement());

System.out.println("Visiting keys using enumertion: ");

Enumeration e2=ht1.keys();

while(e2.hasMoreElements())

System.out.println(e2.nextElement());

System.out.println("Does hashtable contains A as key: " + ht1.containsKey("A"));

System.out.println("Does hashtable contains 15 as value: " + ht1.containsValue(15));

System.out.println("Is hashtable Empty: " + ht1.isEmpty());

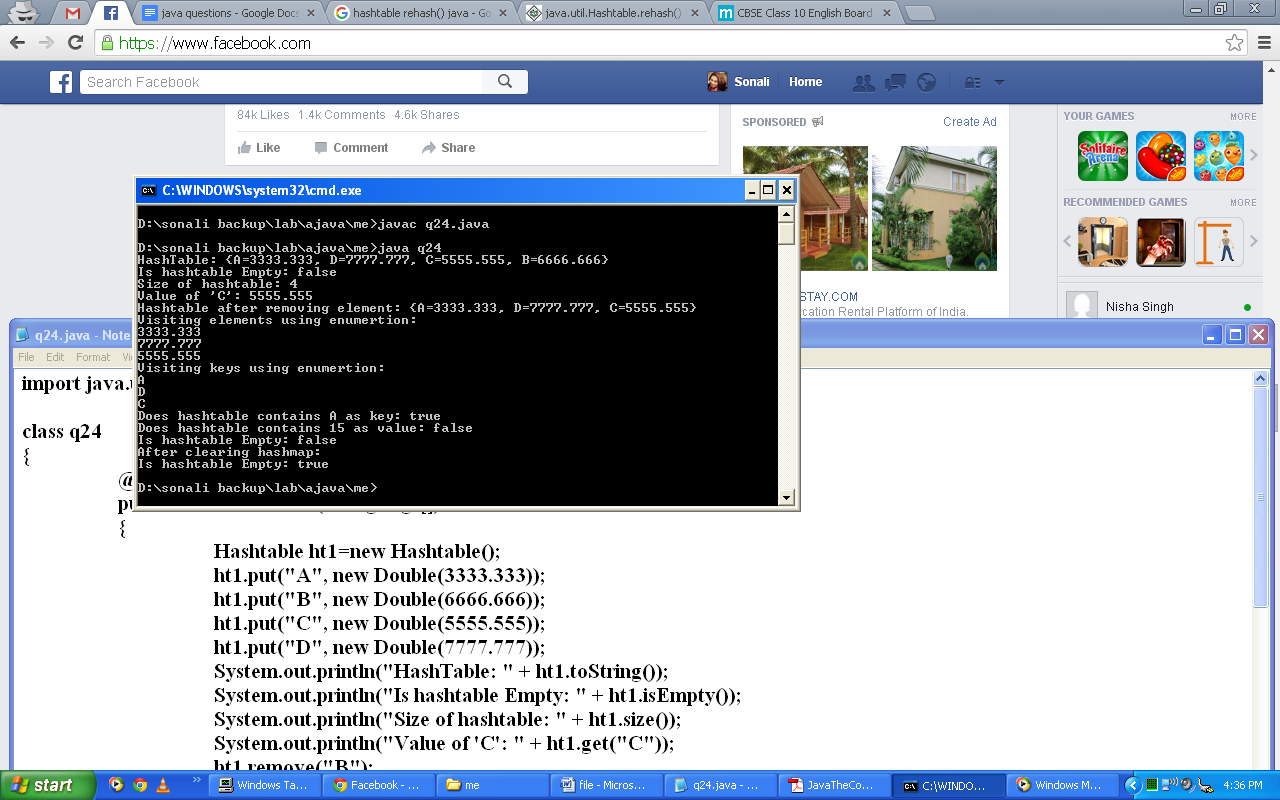
ht1.clear(); System.out.println("After clearing hashmap: ");

System.out.println("Is hashtable Empty: " + ht1.isEmpty());

}

}

**Output:**

****

1. **Write a program to implement the following methods using Properties: getProperty(), list(), load(), propertyNames(), setProperty() and store().**

**Code:**

import java.util.\*;

import java.io.\*;

class q25{

public static void main(String args[]){

Properties ln = new Properties();

Set fn;

String str;

ln.put("Sonali", "Chawla");

ln.put("Bhawna", "Singla");

ln.put("Rajan", "Bajaj");

ln.put("Vinisha", "Sharma");

ln.put("Pooja", "Dudeja");

ln.list(System.out);

System.out.println();

fn= ln.keySet();

System.out.println("Visiting elements using enumeration");

Enumeration e1 = ln.propertyNames();

while(e1.hasMoreElements()) {

str = (String) e1.nextElement();

System.out.println("The last name of " +str + " is " + ln.getProperty(str) + ".");

}

System.out.println();

System.out.println("Searching for Last name of Vinisha:");

str = ln.getProperty("Vinisha");

System.out.println("The last name of Vinisha is "+ str + ".");

FileOutputStream fout=null;

try{

fout=new FileOutputStream("pp.txt");

ln.store(fout,"First and Last Name");

fout.close();

}catch(IOException ex){}

System.out.println();

Properties p2=new Properties();

FileInputStream fin = null;

try { fin = new FileInputStream("pp.txt");

if(fin != null) {

p2.load(fin);

fin.close();

}

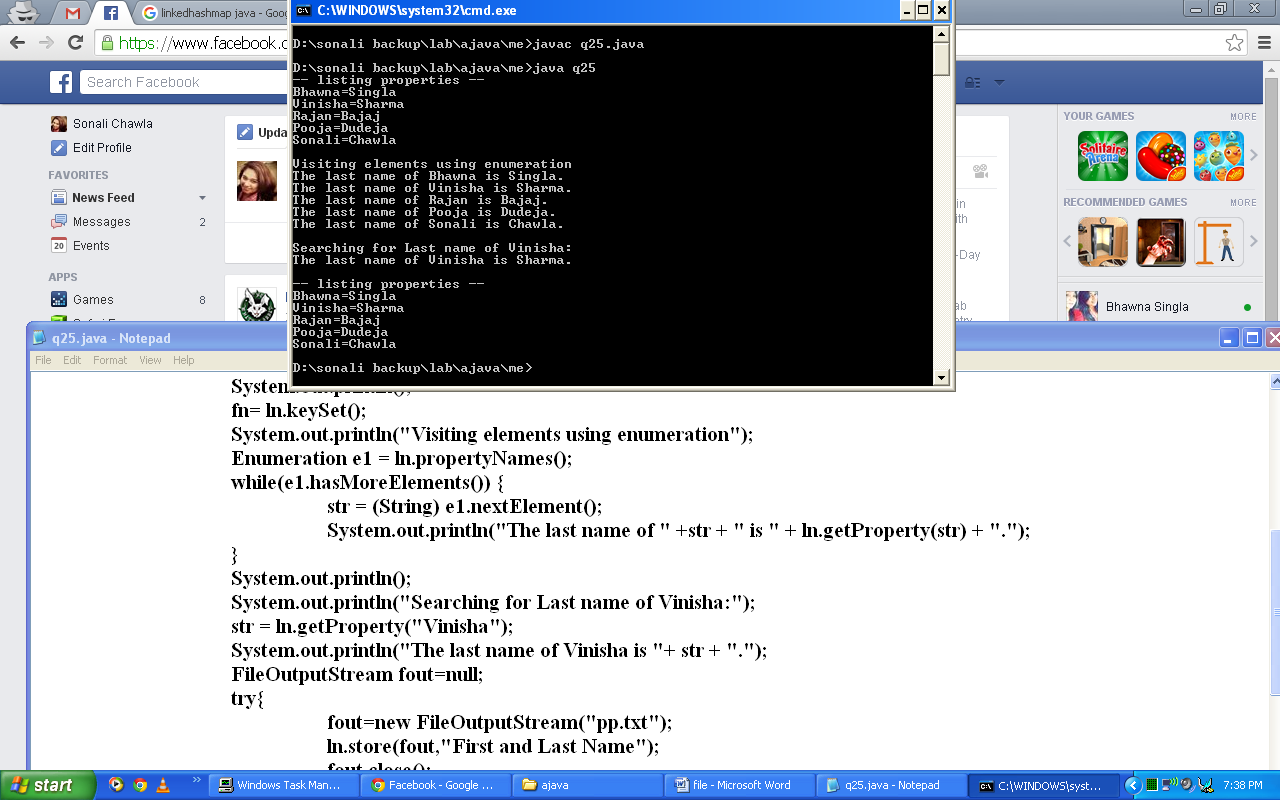
} catch(IOException e) {}

p2.list(System.out);

}

}

**Output:**

****

1. **Write a program to create a simple telephone number database that uses a property list.**

**Code:**

import java.util.\*;

import java.io.\*;

class q26{

public static void main(String args[]) throws IOException{

Properties p1 = new Properties();

BufferedReader br =new BufferedReader(new InputStreamReader(System.in));

String name, number; FileInputStream fin = null; boolean changed = false;

try { fin = new FileInputStream("phonebook.dat");

} catch(FileNotFoundException e) {}

try { if(fin != null) {p1.load(fin); fin.close(); }

} catch(IOException e) {

System.out.println("Error reading file."); }

do {

System.out.println("Enter new name" +" ('quit' to stop): ");

name = br.readLine();

if(name.equals("quit")) continue;

System.out.println("Enter number: ");

number = br.readLine();p1.put(name, number);

changed = true;} while(!name.equals("quit"));

if(changed) {

FileOutputStream fout = new FileOutputStream("phonebook.dat");

p1.store(fout, "Telephone Book");

fout.close(); }

do {

System.out.println("Enter name to find" +" ('quit' to quit): ");

name = br.readLine();

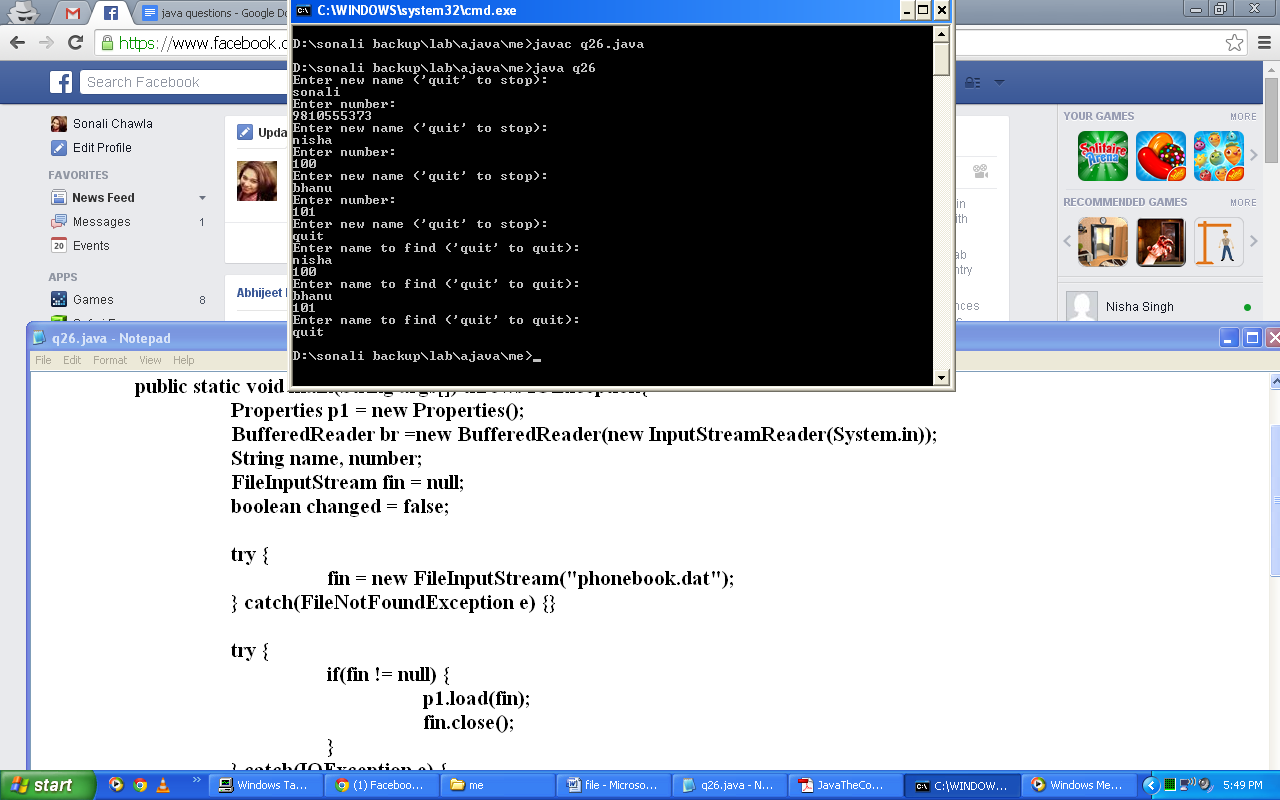
if(name.equals("quit")) continue;

number = (String) p1.get(name);System.out.println(number);

} while(!name.equals("quit")); }

}

**Output:**

****

1. **Write a program to create collection to store student records (Roll number and marks). Arrange this data to be written in the file in sorted order. Students detail must be entered through the keyboard.**

**Code:**

import java.util.\*;

import java.io.\*;

class q27{

@SuppressWarnings("unchecked")

public static void main(String args[]) throws IOException{

TreeMap ts1=new TreeMap();

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String roll,marks;

do{

System.out.println("Enter 'quit' to quit");

System.out.println("Enter roll no.");

roll=br.readLine();

if(roll.equals("quit")) continue;

System.out.println("Enter marks");

marks=br.readLine();

ts1.put(roll,marks);

}while(!roll.equals("quit"));

System.out.println("Record: "+ts1);

Properties p1=new Properties();

p1.putAll(ts1);

try{

FileOutputStream fout=new FileOutputStream("ts.txt");

p1.store(fout,"Student Details");

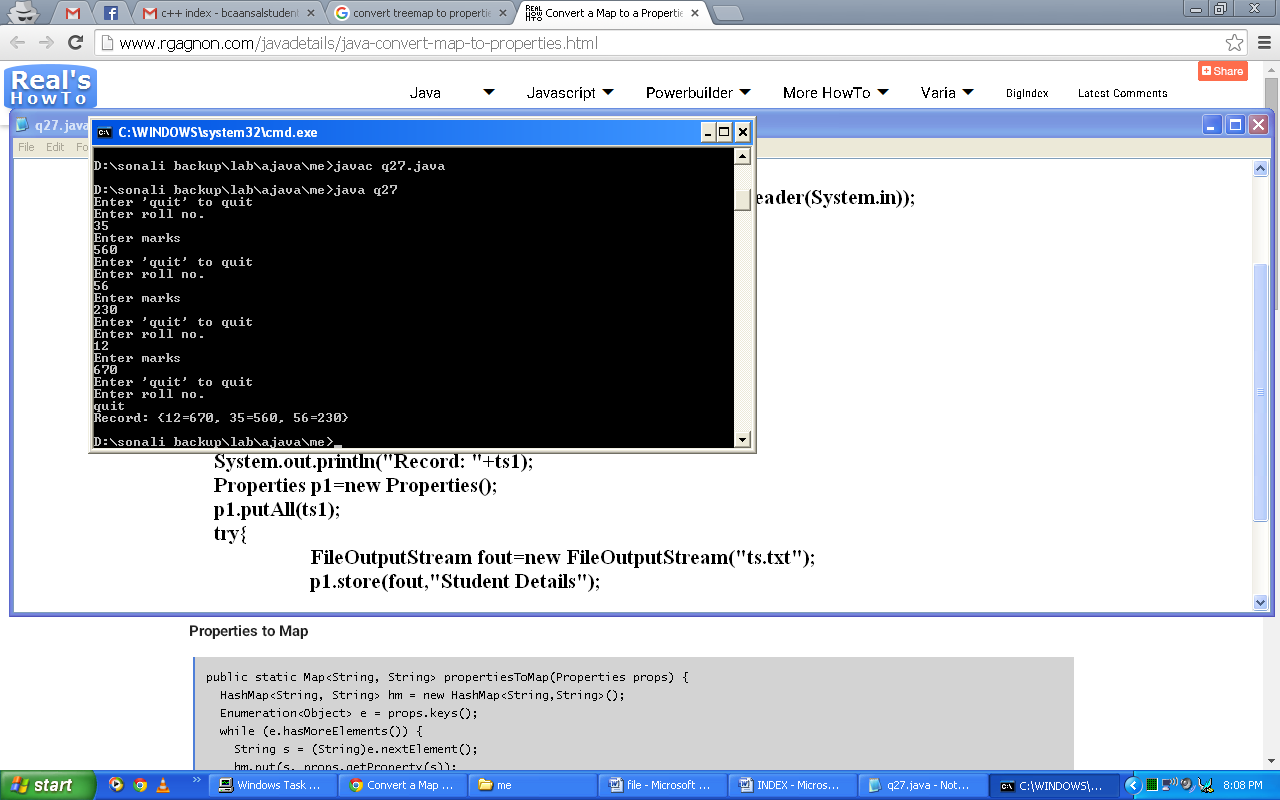
fout.close();

}catch(IOException ex){}

}

}

**Output:**



1. **Create your own collection of an employee number, name, basic salary, DA, HRA and PF. Calculate net salary of the employee. Arrange employee’s record in order of net salary. For comparison create your comparator and also create the code to search for a particular employee on the basis of employee name.**

**Code:**

import java.util.\*;

class Employee{

int empno,basic\_sal,da,hra,pf,net\_sal;

String empname;

public Employee(int n,String na,int bs,int d,int h, int p){

empno=n;

empname=na;

basic\_sal=bs;

da=d;

hra=h;

pf=p;

net\_sal=bs-d+h+p;

}

public String toString(){

return "("+empno+","+empname+","+net\_sal+")";

}

}

class MyComp implements Comparator{

public int compare(Object a,Object b){

Employee aEmp, bEmp;

Integer aInt, bInt;

aEmp=(Employee)a;

bEmp=(Employee)b;

aInt=aEmp.net\_sal;

bInt=bEmp.net\_sal;

return aInt.compareTo(bInt);

}

}

class q28{

@SuppressWarnings("unchecked")

public static void main(String args[]){

LinkedList l1=new LinkedList();

l1.add(new Employee(11,"bhawna",100000,300,100,50));

l1.add(new Employee(2,"nisha",30000,100,100,50));

l1.add(new Employee(322,"sonali",20000,500,50,50));

System.out.println(l1);

System.out.println();

System.out.println("After Sorting:");

Collections.sort(l1,new MyComp());

System.out.println(l1);

int f=0;

Iterator it1=l1.iterator();

while(it1.hasNext()){

Object o1=it1.next();

Employee e1=(Employee)o1;

if((e1.empname).equals("nisha")){

System.out.println("Employee Found at index: "+l1.indexOf(e1));

f=1;

}

}

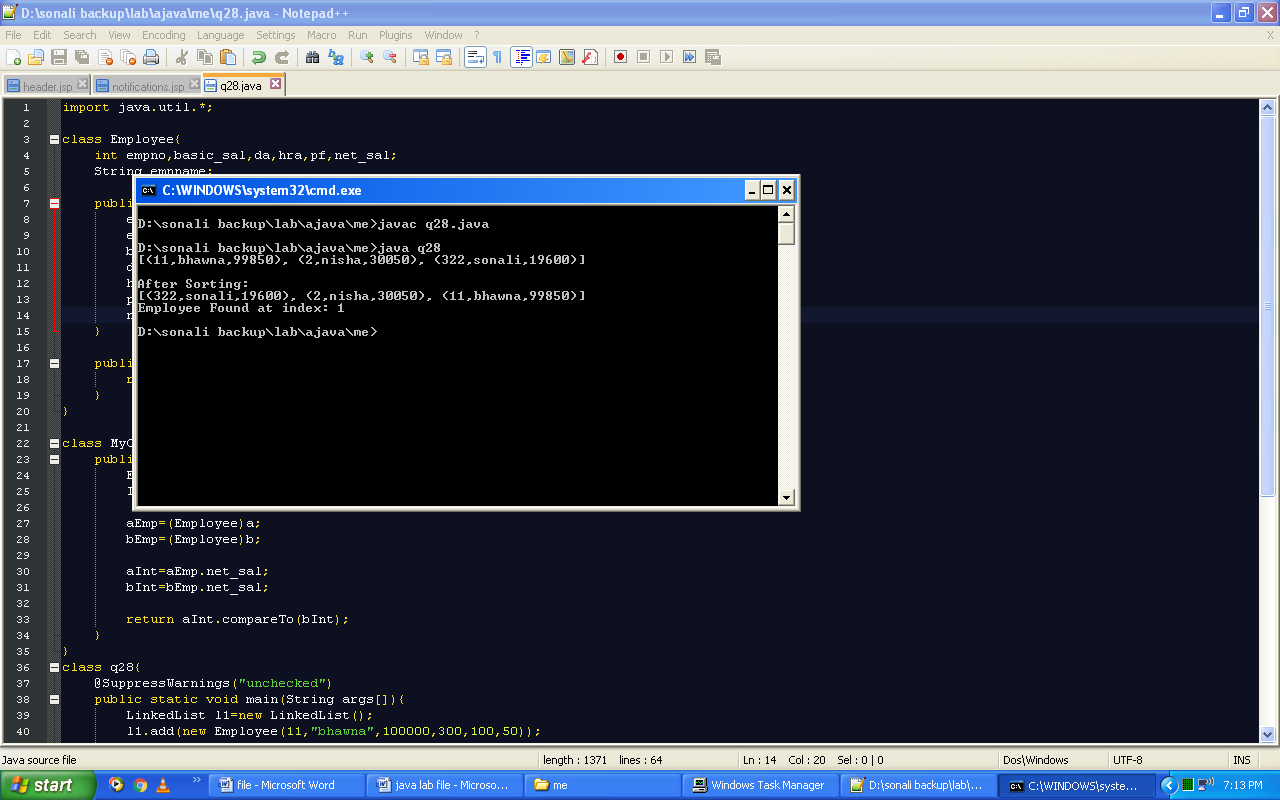
if(f==0)

System.out.println("Employee Not Found");

}

}

**Output:**

****

1. **Create a session to store date as well as time when the particular page was last accessed. If there is no existing session, then create a new session.**

**Code:**

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class q29 extends HttpServlet{

public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException{

HttpSession session=request.getSession();

response.setContentType("text/html");

PrintWriter out=response.getWriter();

Date d1=(Date)session.getAttribute("date1");

if(d1!=null)

out.println("Last Access: "+d1+"<br>");

d1=new Date();

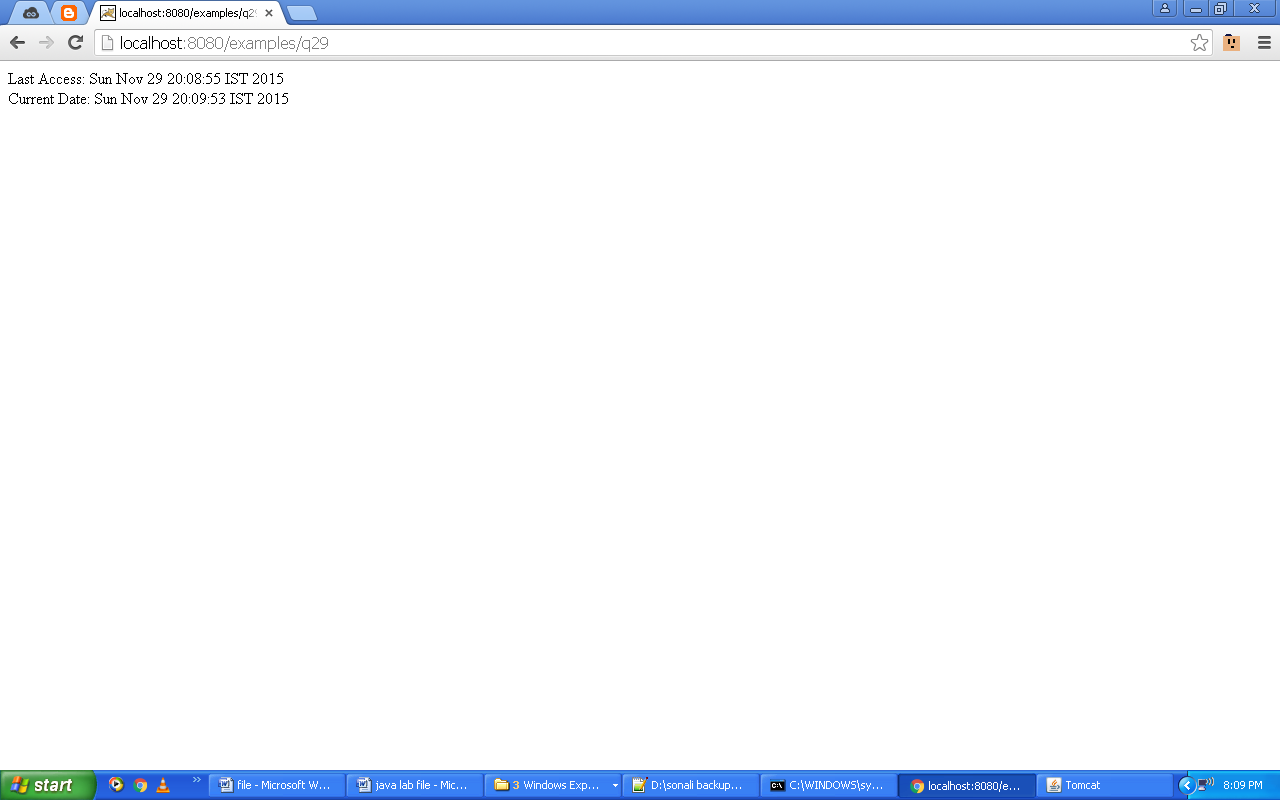
out.println("Current Date: "+d1);

session.setAttribute("date1",d1);

}

}

**Output:**

****

1. **Write a program to implement 3-level chat.**

**Code:**

**(chat.jsp)**

<html>

<head><title>Chat</title></head>

<body><center>

<h1>3-Level Chat</h1>

<table border=0><form method="post" action="q30">

<tr><td align="center">Enter your name:</td></tr>

<tr><td><input type="text" name="name" /><input type="hidden" name="count" value="1"/></td></tr>

<tr><td align="center"><input type="submit" value="Submit"/></td></tr>

</form></table></center>

</body>

</html>

**(q30.java)**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class q30 extends HttpServlet{

public void doPost(HttpServletRequest request,HttpServletResponse response) throws ServletException,IOException{

response.setContentType("text/html");

PrintWriter out=response.getWriter();

String name=request.getParameter("name");

String count=request.getParameter("count");

int c1=Integer.parseInt(count);

out.println("<html><body><center><table border=0><form method='post' action='q30'><tr><td align='center'>");

if(c1==1)

out.println("Hello "+name+". How are you? ");

else if(c1==2)

out.println("OK! What is your favourite color?");

else

out.println("Nice! ;)");

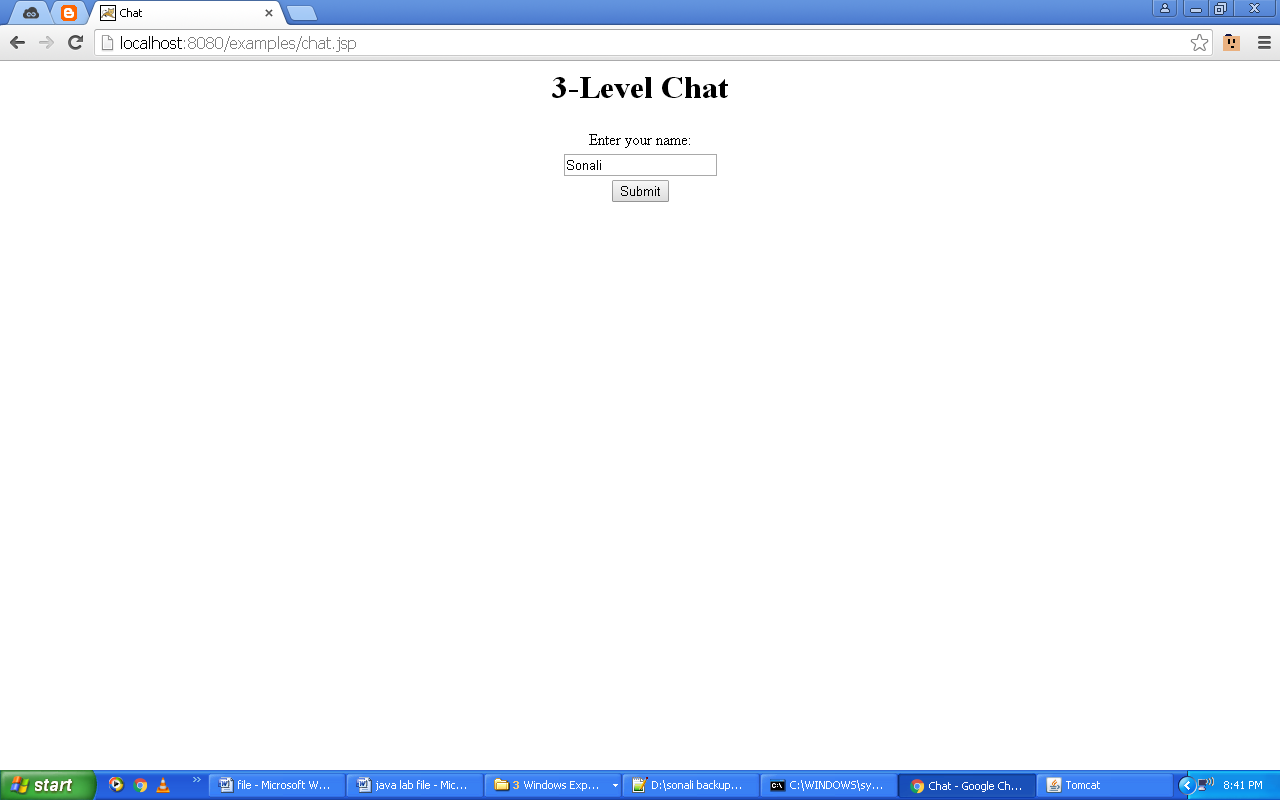
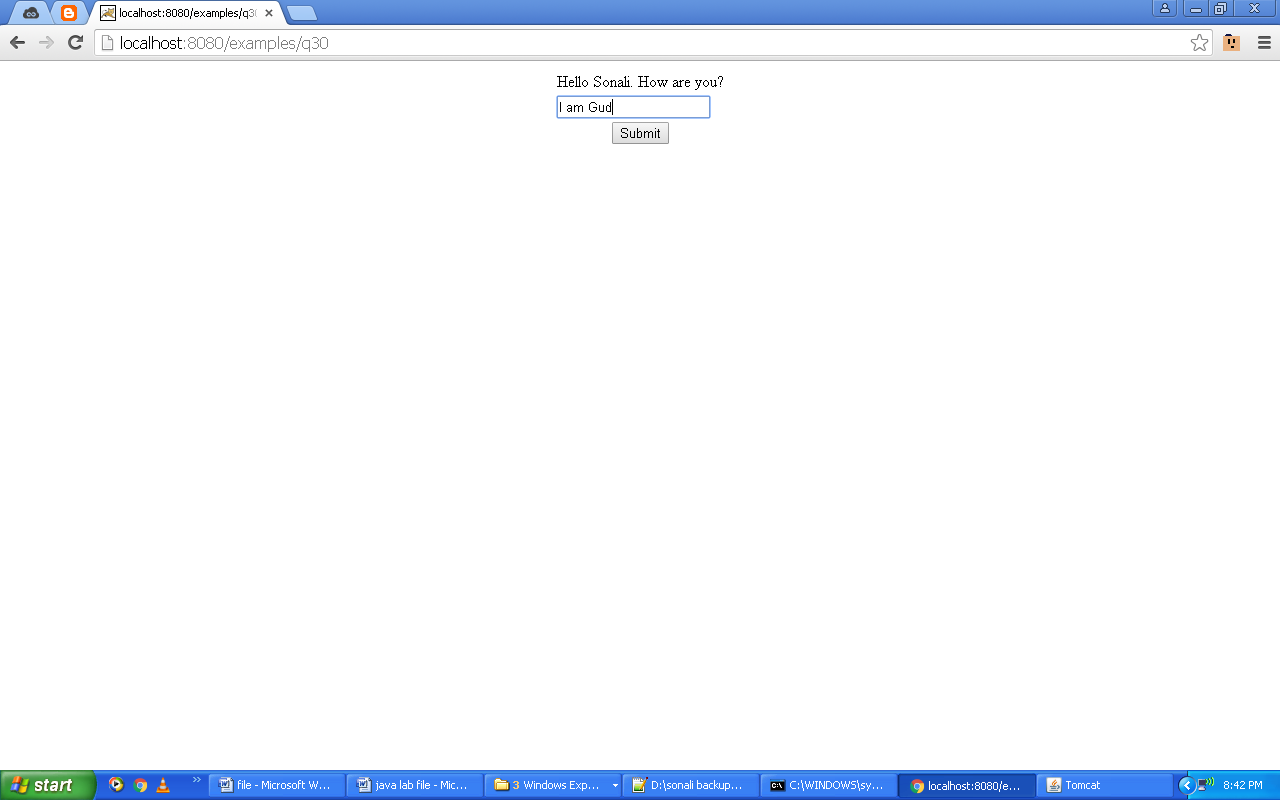
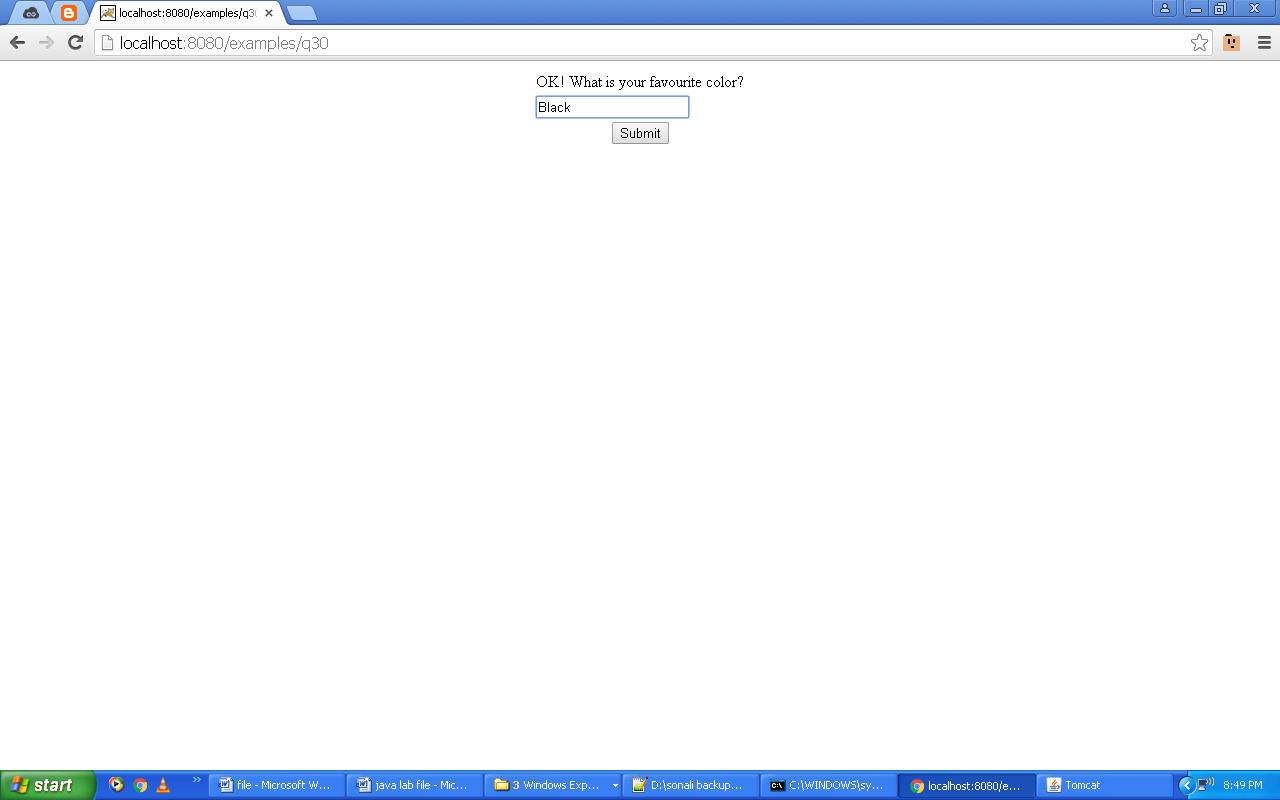
c1=c1+1;

out.println("</td></tr><tr><td><input type='text' name='name'/><input type='hidden' name='count' value='"+c1+"'/></td></tr><tr><td align='center'><input type='submit' value='Submit'/></td></tr></form></table></center></body></html>");

}

}

**Output:**

**  **

1. **Write a program to create a login page and implement database connectivity.**

**Code:**

**(login.jsp)**

<html>

<head><title>Login</title></head>

<body>

<center>

<h2>LOGIN</h2>

<form name="f1" method="post" action="login">

<table border=0>

<tr><th colspan="2" align="center">Enter credentials</th></tr>

<tr><td>Username:</td><td><input type="text" name="uname"/></td></tr>

<tr><td>Password:</td><td><input type="password" name="pass"/></td></tr>

<tr><td colspan="2" align="center"><input type="submit" value="Submit"/></td></tr>

<tr><td colspan="2" align="center"><b>Haven't registered yet? Create <a href="register.jsp">new</a> account.</b></td></tr>

</table>

</form>

</center>

</body>

</html>

(login.java)

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.sql.\*;

public class login extends HttpServlet{

Connection con;

Statement stmt;

ResultSet rs;

public login(){

try{

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost/test","root","");

stmt=con.createStatement();

}catch(Exception e){System.out.println(e);}

}

public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException{

response.setContentType("text/html");

PrintWriter out=response.getWriter();

String uname=request.getParameter("uname");

String pass=request.getParameter("pass");

try{

rs=stmt.executeQuery("select \* from user where username='"+uname+"' and password='"+pass+"'");

if(rs.next()){

int uid=rs.getInt("userid");

RequestDispatcher view=request.getRequestDispatcher("welcome.jsp?i="+uid+"&n="+uname);

view.forward(request,response);

}

else{

out.println("<center><font color='red'>Invalid Username or Password.</font></center>");

RequestDispatcher view=request.getRequestDispatcher("login.jsp");

view.include(request,response);

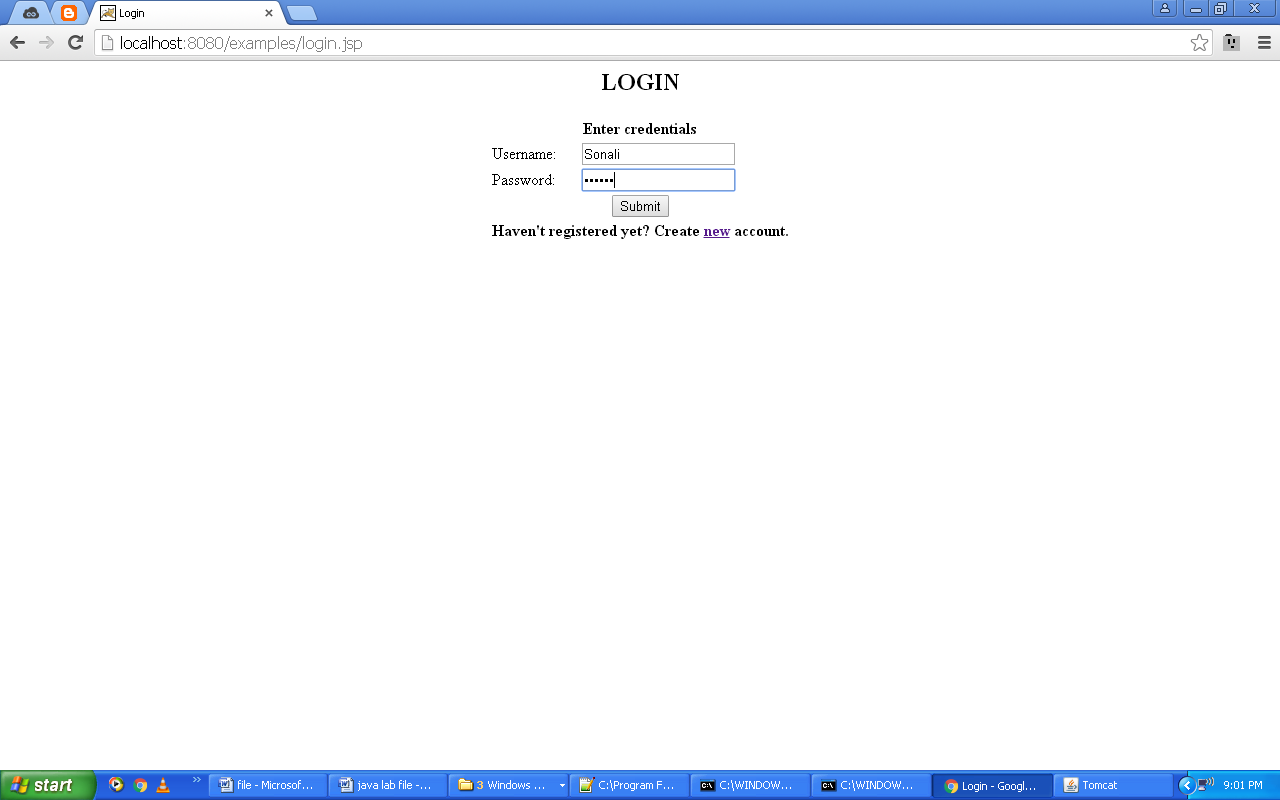
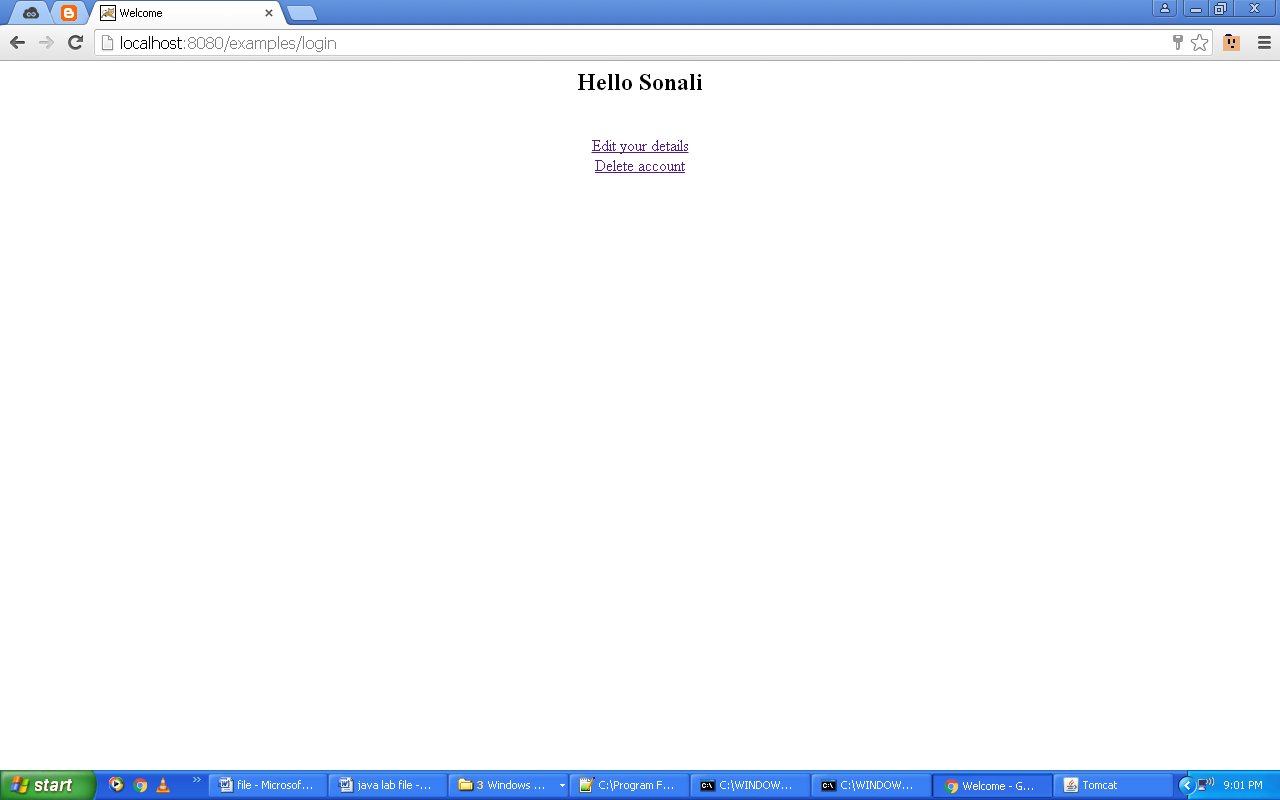
}

}catch(Exception e){System.out.println(e);}

}

}

**Output:**

** **

1. **Create one JSP page that helps you to select a region or a locale and displays date according to selected locale.**

**Code:**

**(File.jsp)**

<html><body><center><table><form action="file1.jsp" method="post">

<tr><td><select name="locale">

<option value="Asia/Kolkata">Kolkata (Asia)</option>

<option value="America/Los\_Angeles">Los Angeles (America)</option>

<option value="Europe/Madrid">Madrid (Europe)</option>

</select></td></tr><tr><td colspan="2" align="center"><input type=submit value="Find Date"></td></tr></form></table></body></html>

**(File1.jsp)**

<html><body><%@ page import="java.util.Date,java.text.\*,java.util.\*"%>

<%String locale=request.getParameter("locale");

Date today=new Date();

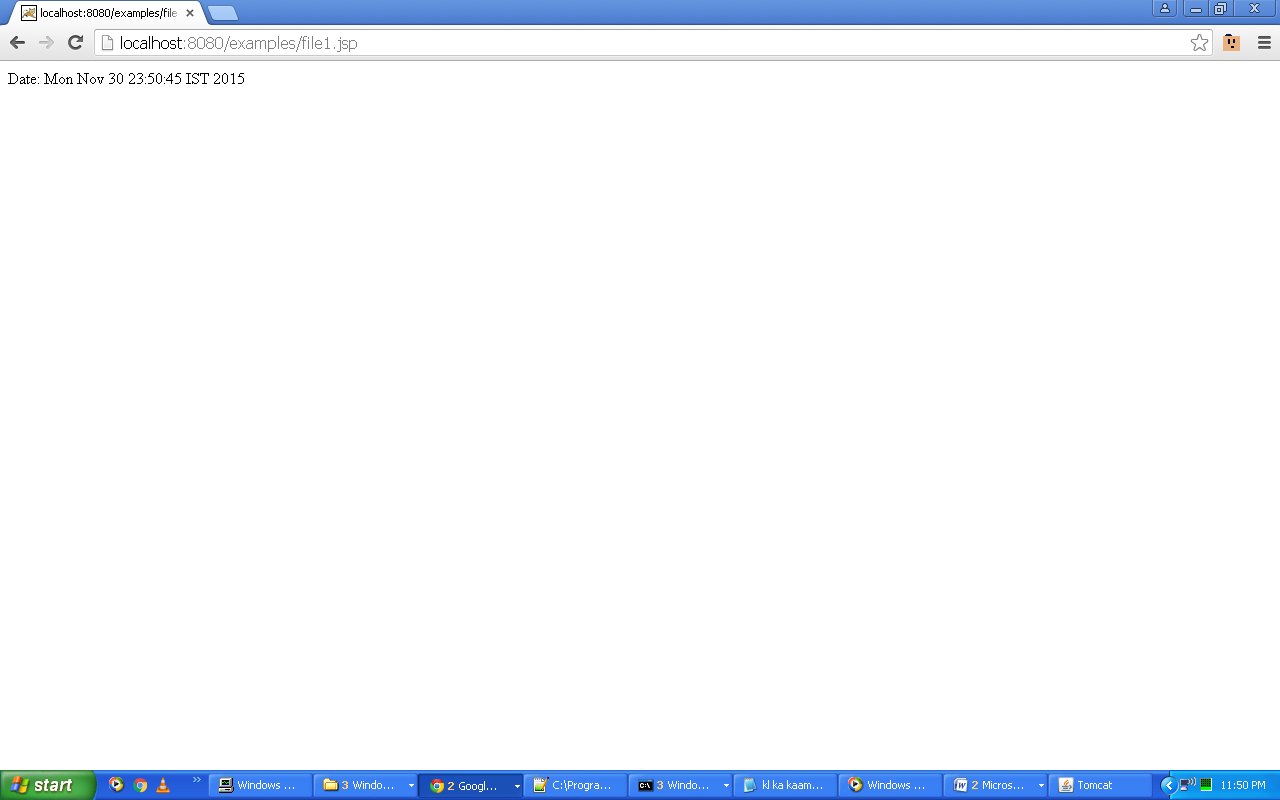
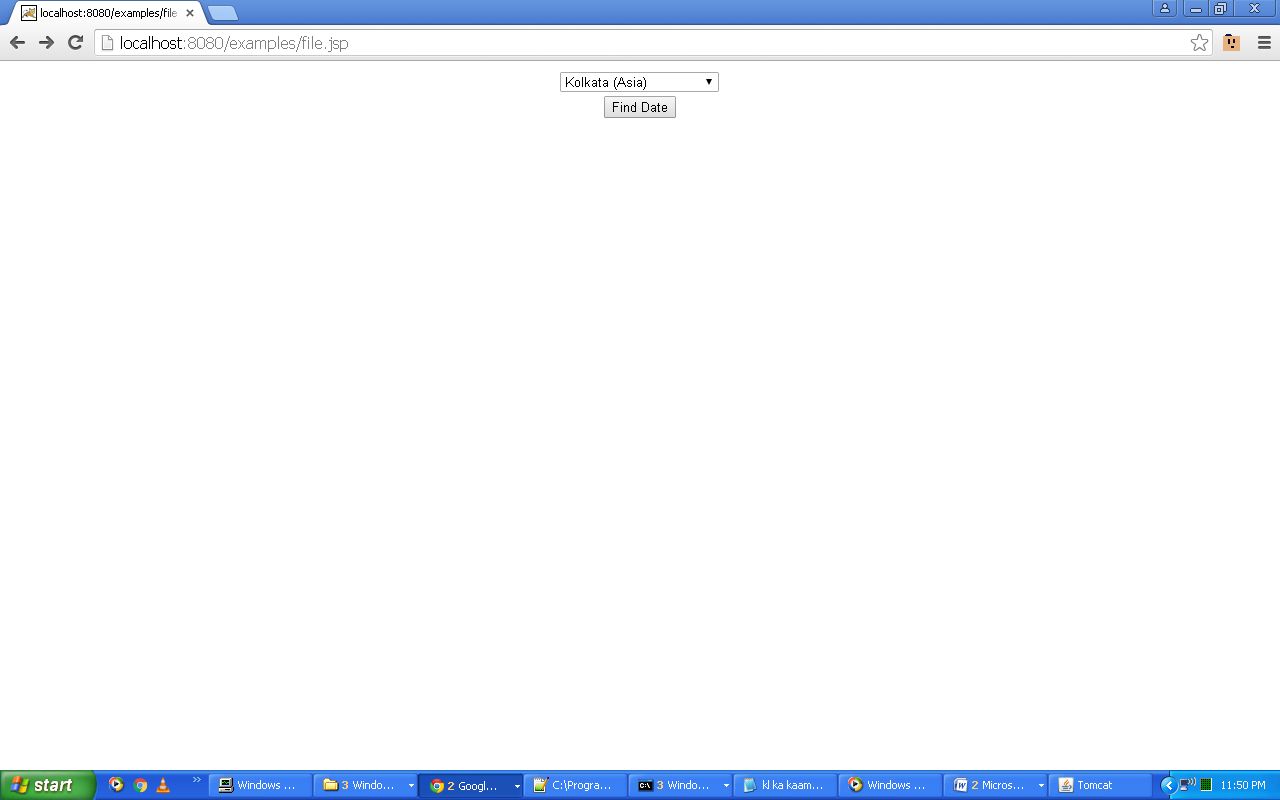
DateFormat df = new SimpleDateFormat("dd-MM-yy HH:mm:SS z");

df.setTimeZone(TimeZone.getTimeZone(locale));

String date1 = df.format(today);

out.println("Date: "+today);%></body></html>

**Output:**

****

1. **Create a JSP page to calculate the percentage of student. Create different function to perform this job.**

**Code:**

<html>

<body><center>

<h1>Calculate percentage </h1>

<%! int per(int m1, int m2){

return (m1+m2)/2;

} %>

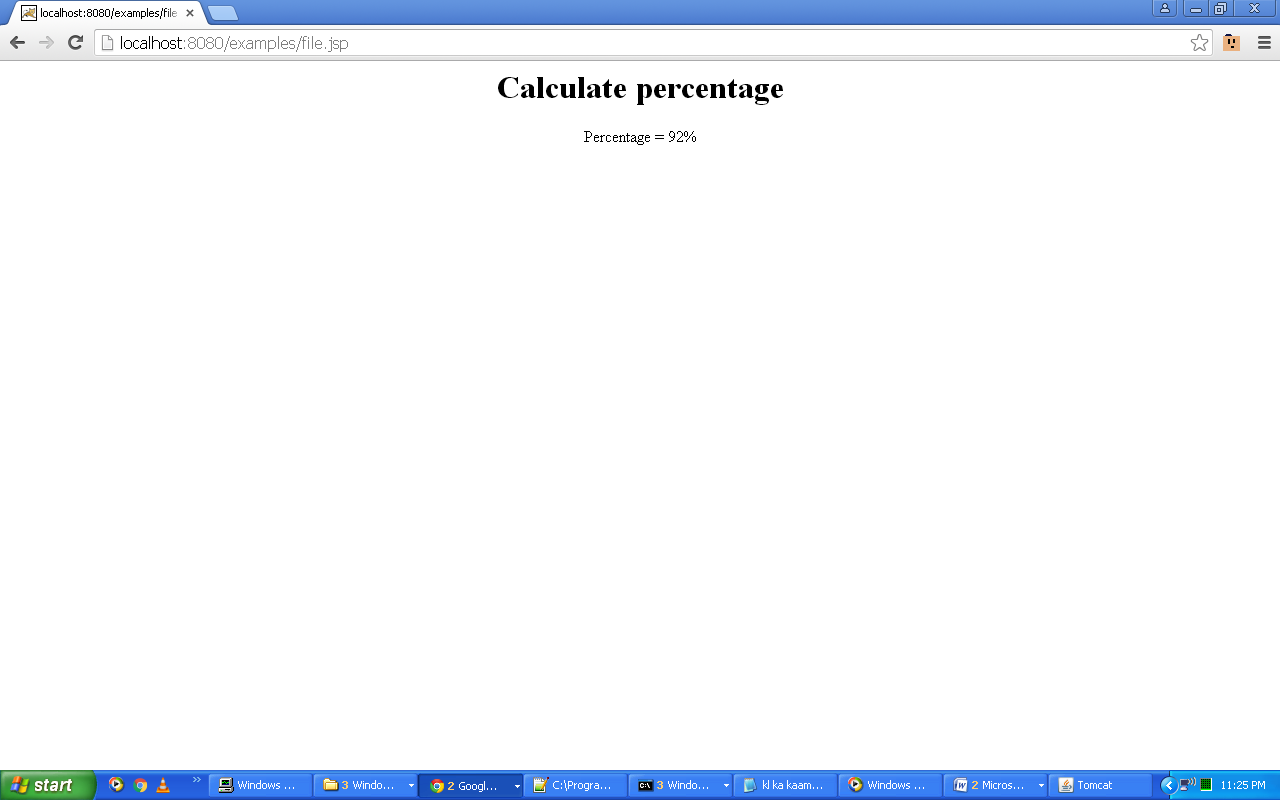
Percentage = <%=per(90,95)%>%

</center>

</body>

</html>

**Output:**

****

1. **Create a JSP page to perform division. Create different function to perform this job.**

**Code:**

**(File.jsp)**

<html><body><center><table><form action="file1.jsp"><tr><td>Enter first number : </td><td><input type="text" name=n1></td></tr><tr><td>Enter second number : </td><td><input type="text" name=n2></td></tr><tr><td colspan="2" align="center"><input type=submit value="Divide"></td></tr></form></table></body></html>

**(File1.jsp)**

<%@ page contentType="text/html" isThreadSafe="false" buffer="5kb" errorPage="exception.jsp" autoFlush="true" %>

<%String str1=request.getParameter("n1");

String str2=request.getParameter("n2");

double d1=Integer.parseInt(str1);

double d2=Integer.parseInt(str2);

double divide=d1/d2;

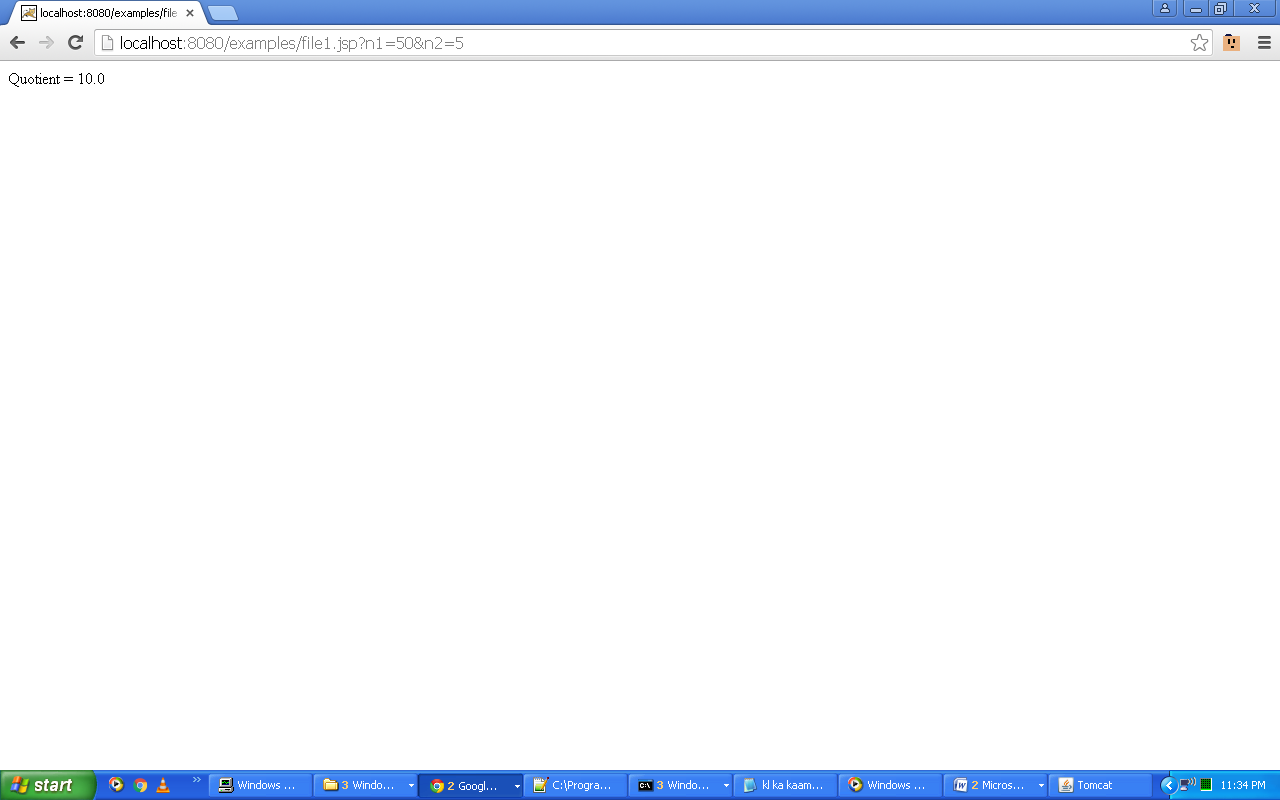
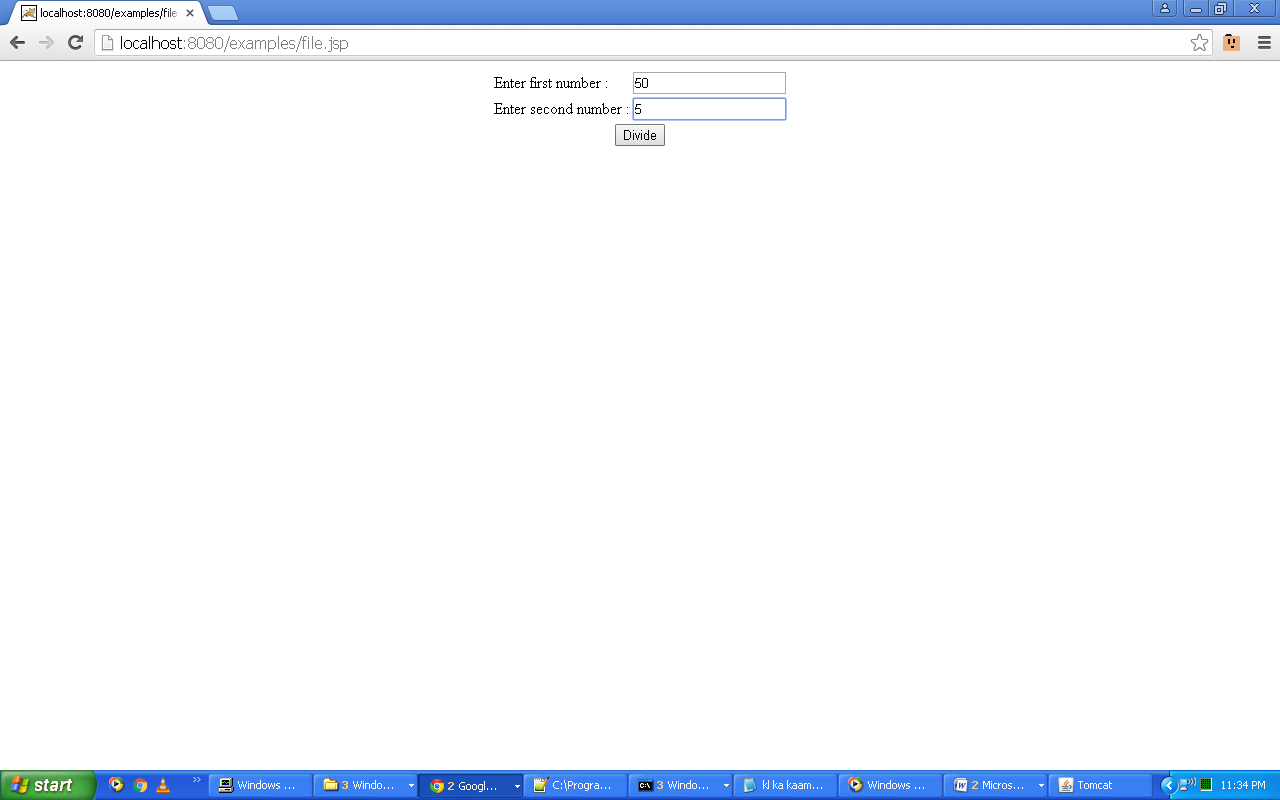
out.println("Quotient = " + divide);%>

**(File2.jsp)**

<%@ page isErrorPage="true" %>

There is an Exception: <%= exception %><br>Please Enter correct values

**Output:**

****

1. **Write a program to implement all the page directives.**

**Code:**

**(File.jsp)**

<html><body><center>

<%@ page import="java.io.\*"%>

<%@ page session="true"%>

<%@ page contentType="text/html"%>

<%@ page isThreadSafe="false"%>

<%@ page errorPage="file2.jsp"%>

<%@ page buffer="5kb"%>

<%@ page language="java"%>

<%

int a=10;int b=0;

int c=a/b;

out.println(c);

%>

</center></body></html>

**(File2.jsp)**

<html><body>

<%@ page isErrorPage="true"%>

OOPS!! There is an Error.

</body></html>

**Output:**

****

1. **Write a program to implement i) include directive ii) include directive with parameter.**
2. **Code:**

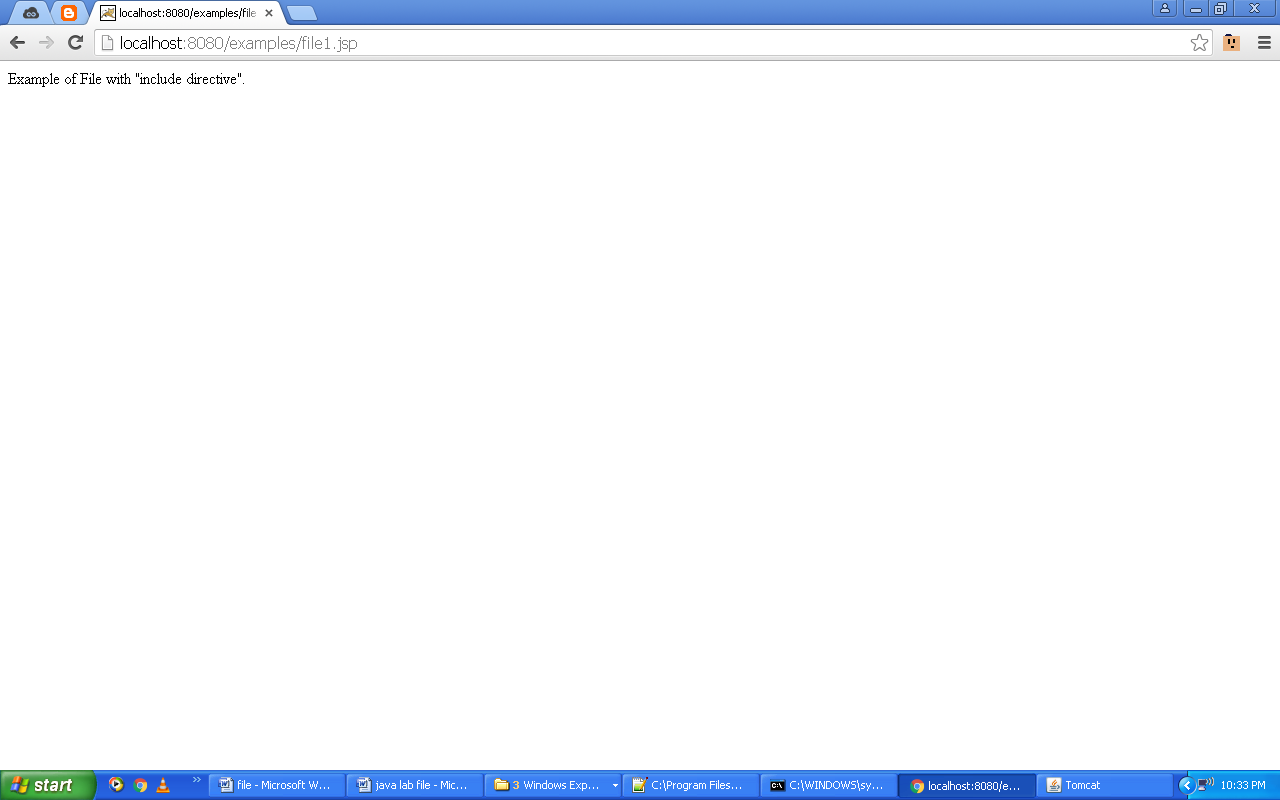
**(File1.jsp)**

<html><body><%@ include file="file2.jsp"%></body></html>

**(File2.jsp)**

<html><body>Example of File with "include directive".</body></html>

**Output:**

****

1. **Code:**

**(File1.jsp)**

<html><body><jsp:include page="file2.jsp">

<jsp:param name="first" value="Sonali"/><jsp:param name="last" value="Chawla"/>

</jsp:include></body></html>

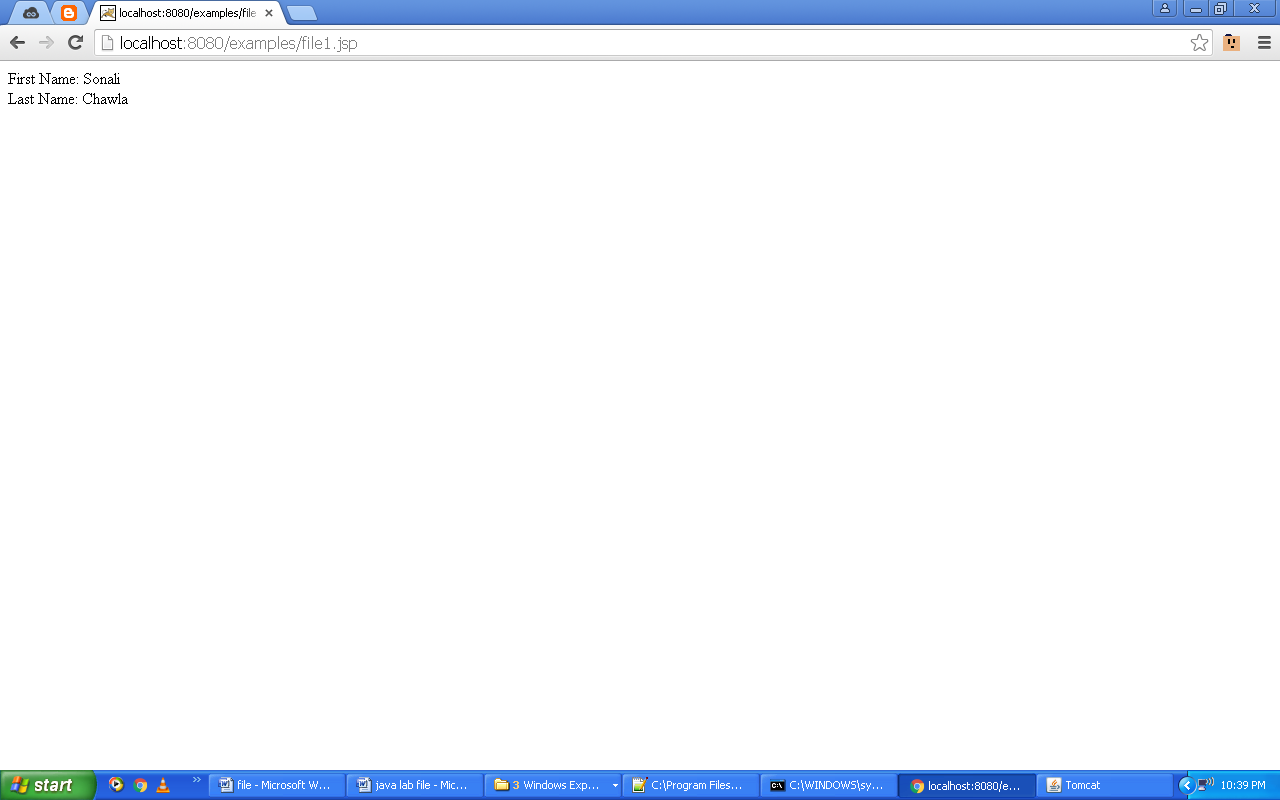
**(File2.jsp)**

<html><body>

First Name: <%= request.getParameter("first") %><br>Last Name: <%= request.getParameter("last") %>

</body></html>

**Output:**

****

1. **Write a program to implement exceptional handling in JSP.**

**Code:**

<html>

<head><title></title></head>

<body>

<%

try{

int a,b,c;

a=10;

b=0;

c=a/b;

out.println(c);

}catch(Exception e){

out.println("There is an Exception: "+e);

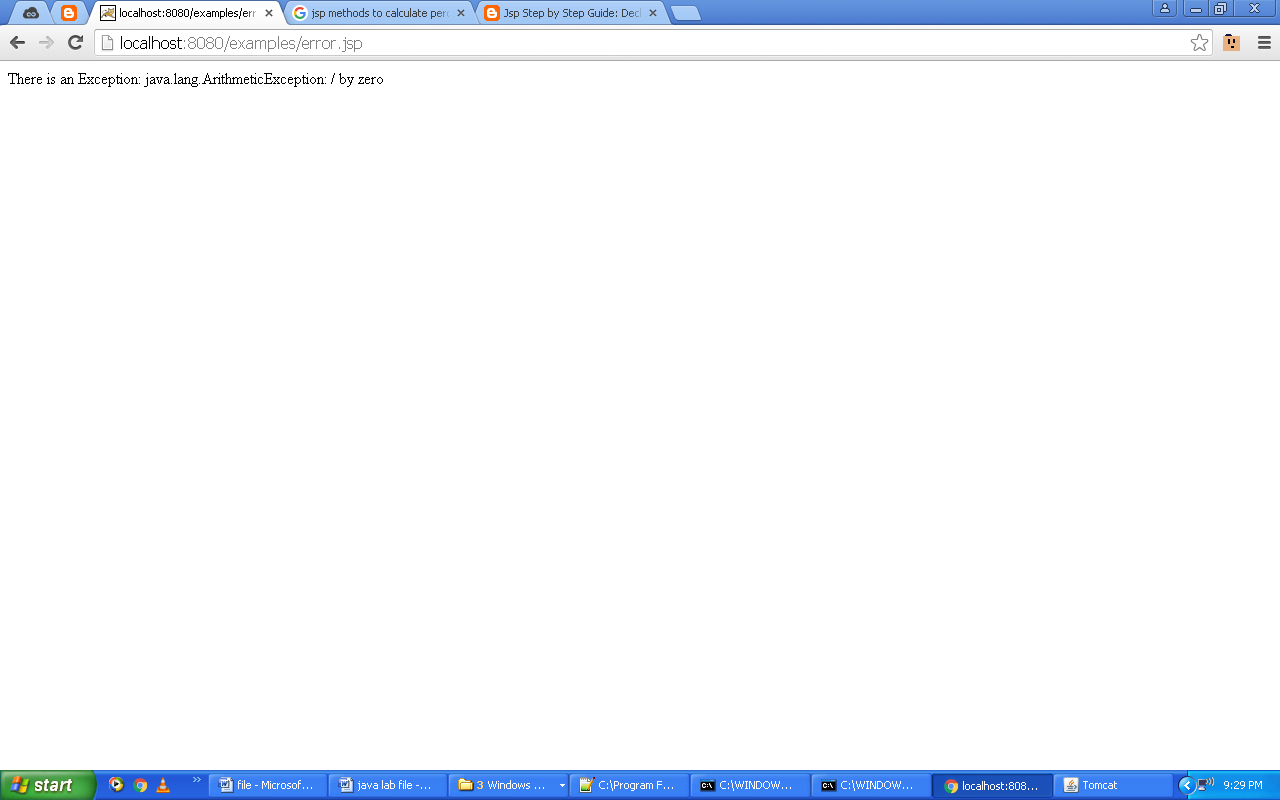
}

%>

</body>

</html>

**Output:**

****

1. **Write a program to implement action tags in JSP.**

**Code:**

**(File.jsp)**

<html><body>

<jsp:include page="file1.jsp"/>

</body></html>

**(File1.jsp)**

<html><body>

<jsp:forward page="file2.jsp">

<jsp:param name="first" value="Sonali"/>

<jsp:param name="last" value="Chawla"/>

</jsp:forward>

</body></html>

**(File2.jsp)**

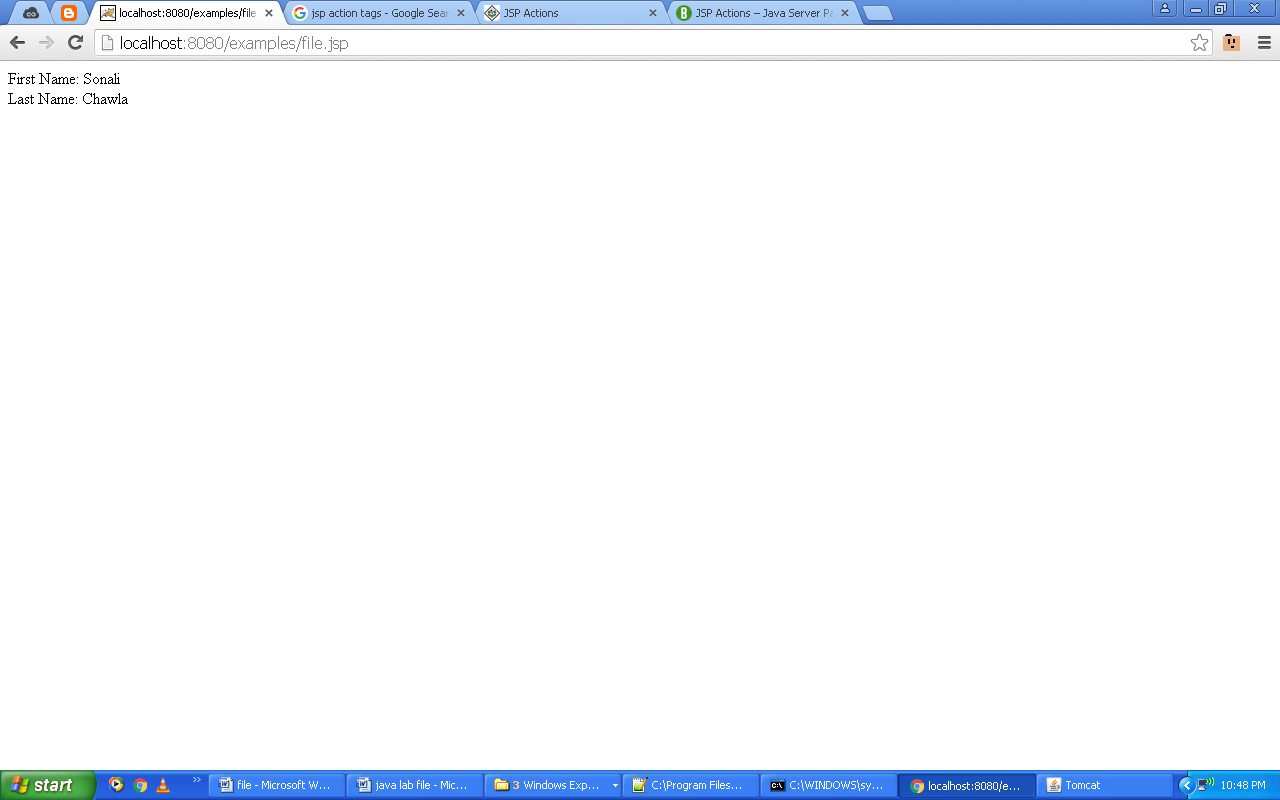
<html><body>

First Name: <%= request.getParameter("first") %><br>

Last Name: <%= request.getParameter("last") %>

</body></html>

**Output:**

****

1. **Write a program to implement jsp: useBean, jsp: setProperty, jsp: getProperty action tags.**

**Code:**

**(File.jsp)**

<html><body><center><table border="0">

<form action="file1.jsp" method="post">

<tr><td>User Name: </td><td><input type="text" name="username"></td></tr>

<tr><td>User Password: </td><td><input type="password" name="password"></td></tr>

<tr><td>User Age: </td><td><input type="text" name="age"></td></tr>

<tr><td colspan="2" align="center"><input type="submit" value="Register"></td></tr>

</form></table></center></body></html>

**(File1.jsp)**

<jsp:useBean id="userinfo" class="file2.file2"></jsp:useBean>

<jsp:setProperty property="\*" name="userinfo"/>

You have enterted below details:<br>

<jsp:getProperty property="username" name="userinfo"/><br>

<jsp:getProperty property="password" name="userinfo"/><br>

<jsp:getProperty property="age" name="userinfo" /><br>

**(file2.java)**

package file2;

public class file2 {

public file2() {}

private String username;

private int age;

private String password;

public String getUsername() {

return username; }

public void setUsername(String username) {

this.username = username; }

public int getAge() {

return age; }

public void setAge(int age) {

this.age = age; }

public String getPassword() {

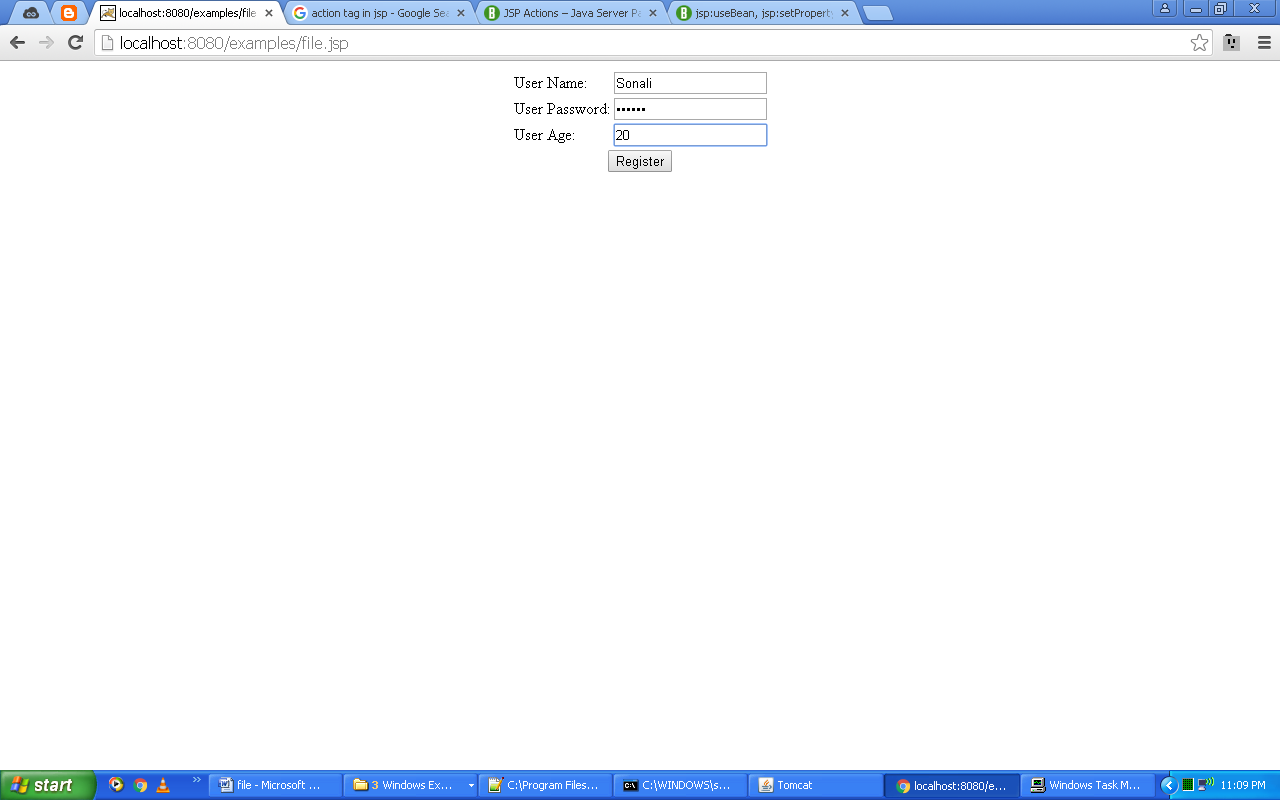
return password; }

public void setPassword(String password) {

this.password = password; }

}

**Output:**

****

1. **Write a program to implement all the methods of request implicit object.**

**Code:**

**(File.jsp)**

<html><body><center><table border="0">

<form action="file2.jsp" method="post">

<tr><td>User Name: </td><td><input type="text" name="username"></td></tr>

<tr><td>User Password: </td><td><input type="password" name="password"></td></tr>

<tr><td>User Age: </td><td><input type="text" name="age"></td></tr>

<tr><td colspan="2" align="center"><input type="submit" value="Register"></td></tr>

</form></table></center></body></html>

**(File2.jsp)**

<html><body>

<%

String name=request.getParameter("username");

String password=request.getParameter("passwprd");

String age=request.getParameter("age");

out.println("Name: "+name);

out.println("<br>");

out.println("Password: "+password);

out.println("<br>");

out.println("Age: "+age);

out.println("<br>");

out.println("<br>");

out.println("Setting and Getting an attribute");

request.setAttribute("user","JSP");

out.println("<br>");

out.println("Username: "+request.getAttribute("user"));

out.println("<br>");

out.println("<br>");

out.println("Cookies:");

Cookie[] ch=request.getCookies();

for(int i=0;i<ch.length;i++){

out.println(ch[i].getName()+": "+ch[i].getValue());

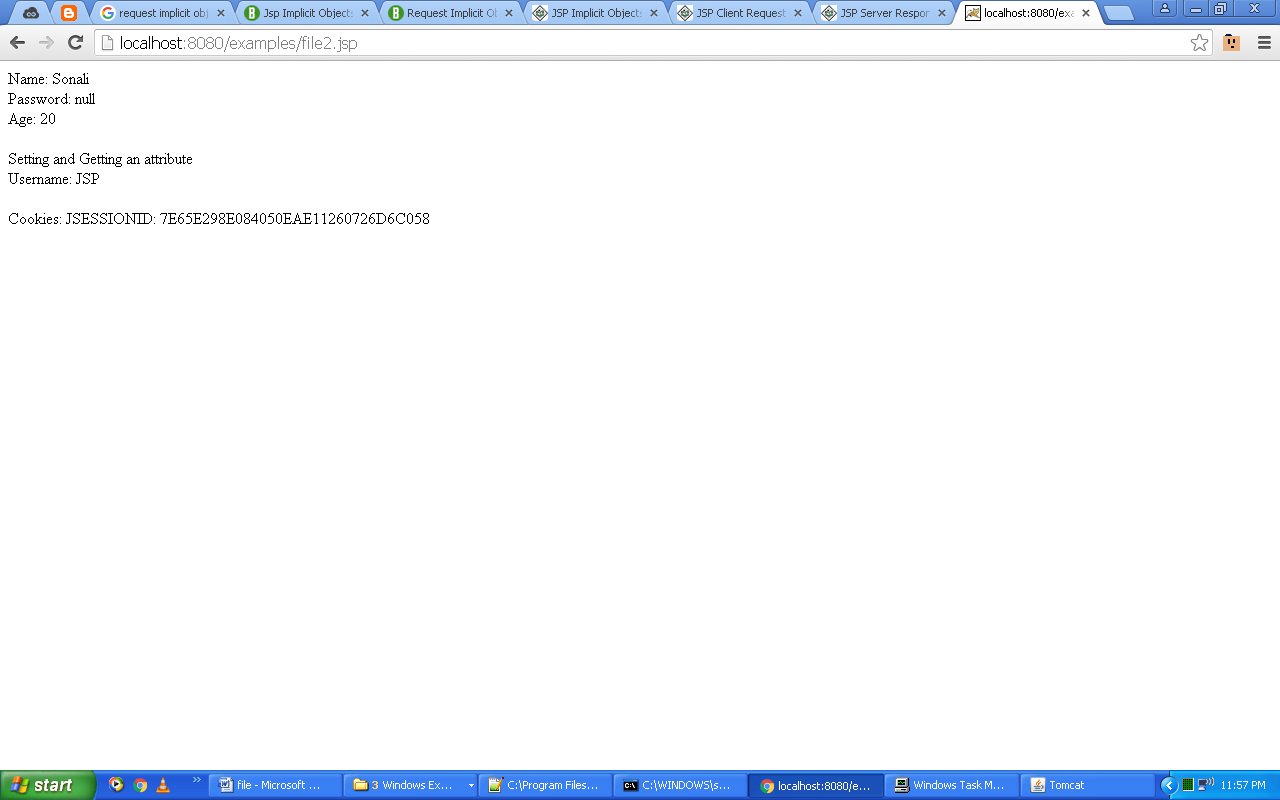
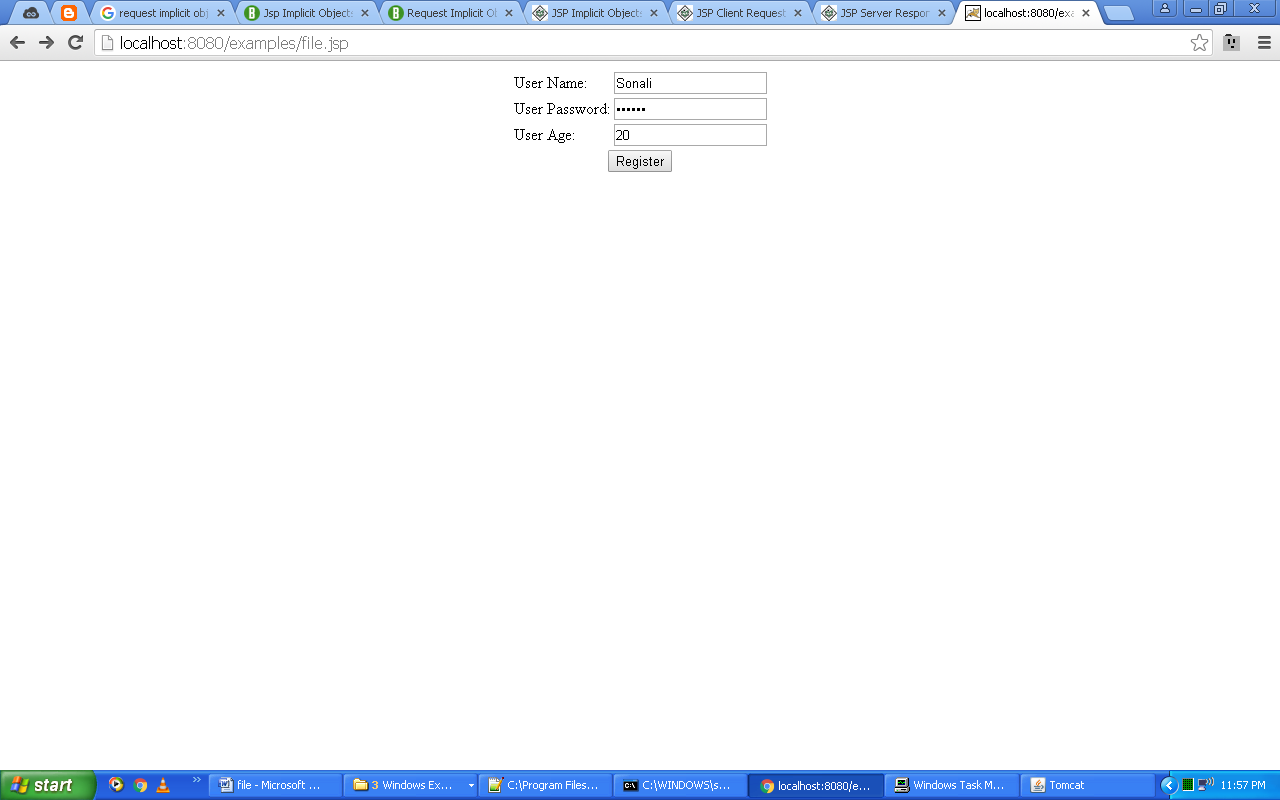
out.println("<br>");

}

%>

</body></html>

**Output:**

****

1. **Write a program to implement all response implicit objects.**

**Code:**

**(File.jsp)**

<html><body><center><table border="0"><form action="file1.jsp" method="post"> <tr><td>User Name: </td><td><input type="text" name="username"></td></tr><tr><td>User Password: </td><td><input type="password" name="password"></td></tr><tr><td colspan="2" align="center"><input type="submit" value="Register"></td></tr></form></table></center> </body></html>

**(File1.jsp)**

<html><body><%

String name=request.getParameter("username");String pass=request.getParameter("password");

out.println(name);out.println(pass);if(name.equals("Sonali") && pass.equals("sonali"))

response.sendRedirect("file2.jsp");

else

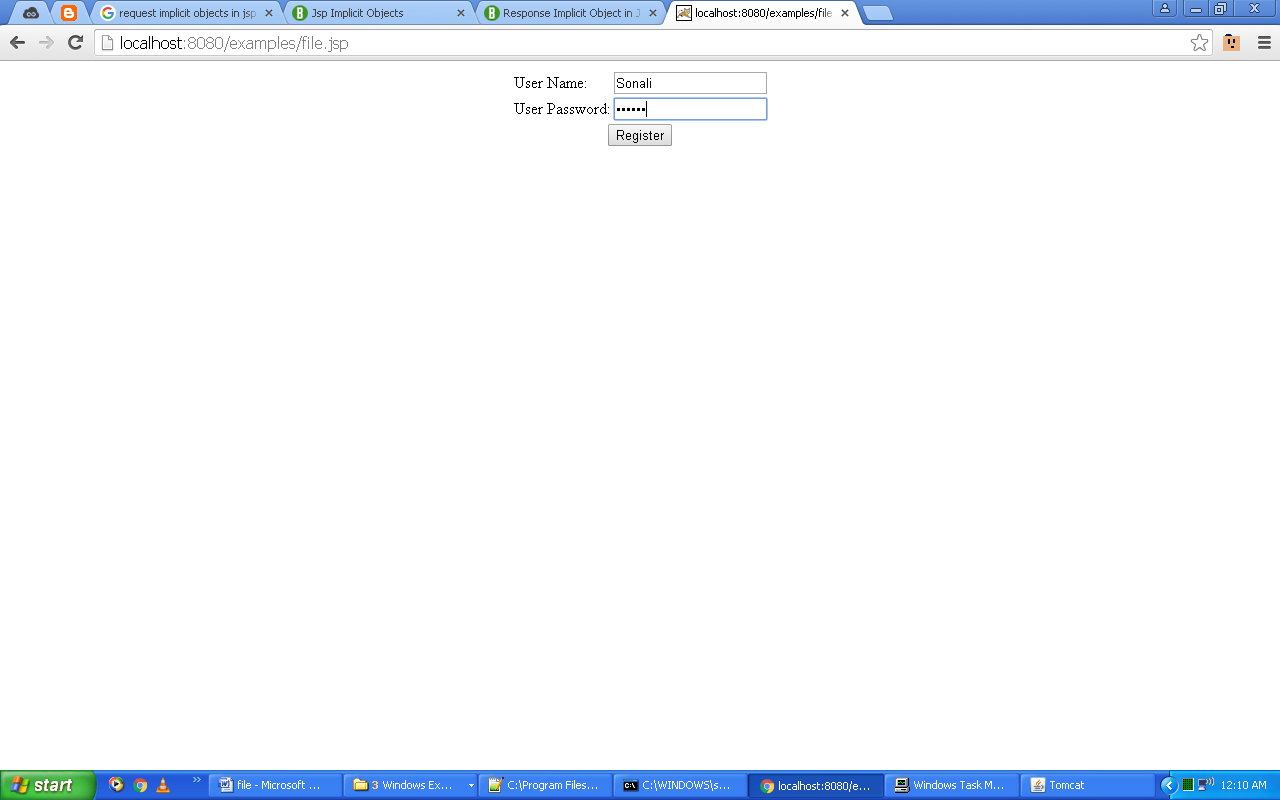
out.println("Login Fail");

%></body></html>

**(File2.jsp)**

<html><body><%response.setContentType("text/html");out.println("Login Successful.");%></body></html>

**Output:**

****

1. **Write a program to implement all methods of application object.**

**Code:**

<html>

<body><center>

<%

Integer counter= (Integer)application.getAttribute("numberOfVisits");

if( counter ==null || counter == 0 ){

counter = 1;

}else{

counter = counter+ 1;

}

application.setAttribute("numberOfVisits", counter);

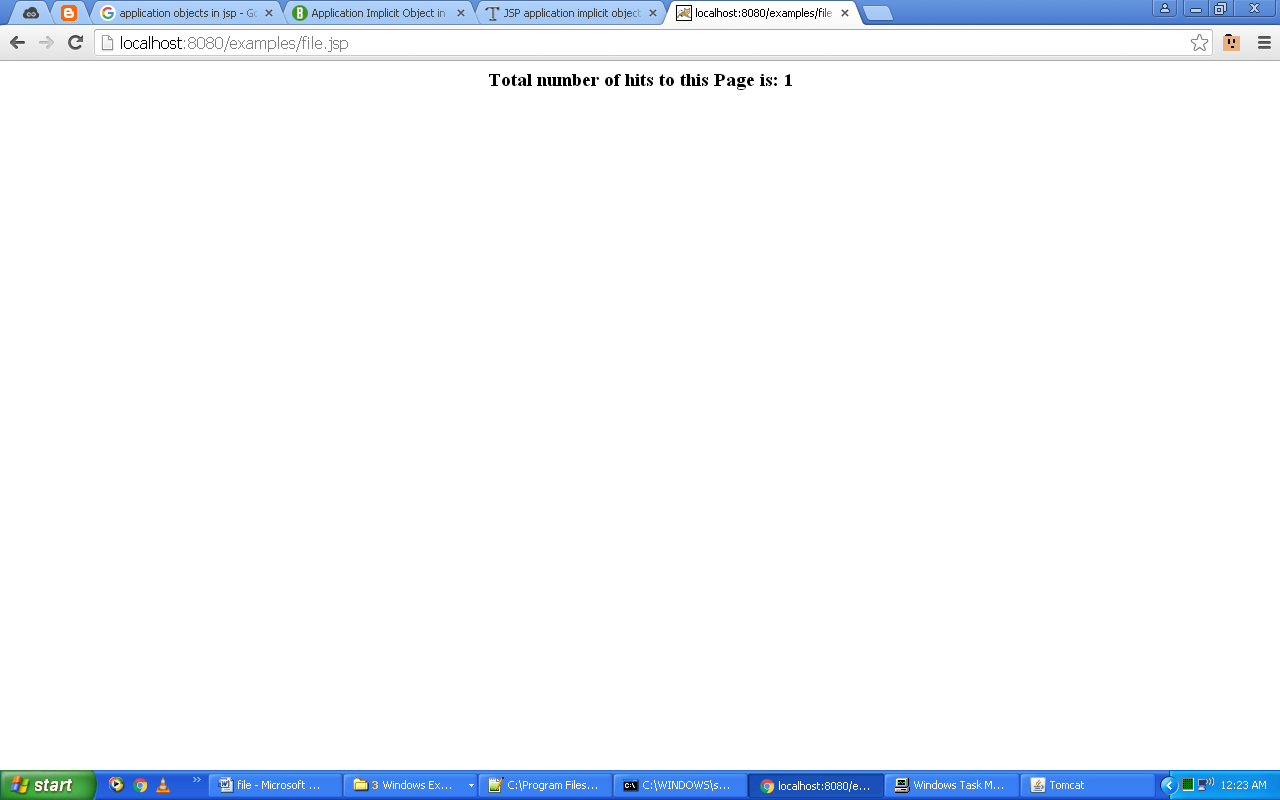
%>

<h3>Total number of hits to this Page is: <%= counter%></h3></center>

</body>

</html>

**Output:**

****

1. **Write a program to implement all methods of GenericServlet class**

**Code:**

package chat1;

import java.io.\*;

import javax.servlet.\*;

public class q30 extends GenericServlet{

public void service(ServletRequest request,ServletResponse response) throws ServletException, IOException{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

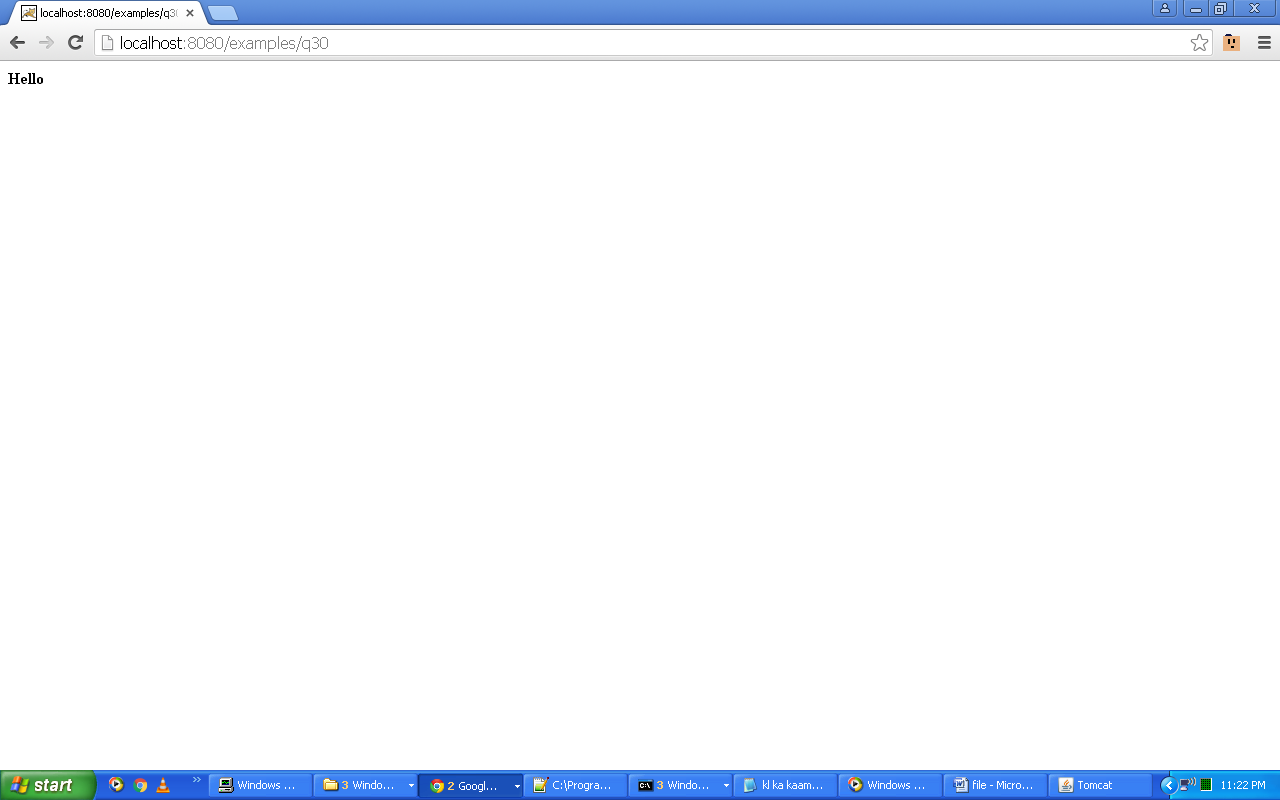
out.println("<B>Hello");

out.close();

}

}

**Output:**

****