Sakhi Patel
Student_id: wz7995

Symbol Table

```
package Symbol_Table;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Iterator; import
java.util.ArrayList; import
java.util.Hashtable; import
java.util.Scanner; import
java.util.Set; import
java.util.Stack;


public class Symbol_Table
       {
        //hold current block elements
          static ArrayList<Hashtable<String, String>> mainHTable = new
ArrayList<Hashtable<String, String>>();


       //hold stored block elements after }
          static ArrayList<Hashtable<String, String>> old_mainHTable = new
ArrayList<Hashtable<String, String>>();

          /* Insert Function*/
    public static Hashtable<String, String>Insert(Hashtable<String, String>
newBlockk,String key)
         {
              newBlockk.put(key, key);
              return newBlockk;
          }



    /* Find in current Block finds the key in current block i.e. mainHtable*/
          public static String find_in_current(String finding_key,int curr_block)
         {
              Hashtable<String, String> scanBlock = mainHTable.get(curr_block);


       if(scanBlock.containsKey(finding_key))
              {
                  return scanBlock.get(finding_key);
              }
        else
              return null;
          }
```

```
    /* Find in All Block finds key in all the blocks before mainHtable which is the
current block */

     public static String find_in_All(String finding_key)
          {

              Iterator<Hashtable<String,String>> it = old_mainHTable.iterator();
                          while(it.hasNext())
              {

                Hashtable<String,String> scanBlock = it.next();
                  if (scanBlock.containsKey(finding_key))
                  {
                       return scanBlock.get(finding_key);
                  }
             }
                       return null;
          }


          /*DISPLAY */
          public static void Display()
          {
                 for(Integer i=0;i<mainHTable.size();i++)
                 {
                        Hashtable<String,String> temp = mainHTable.get(i);
                            Set<String> value = temp.keySet();//returns keys
                     contained in mainHTable
                        System.out.print("scope"+i+" has:");
                        for(String token:value)
                        {
                               System.out.println(token+"");
                        }
                        System.out.println(" ");
                 }
          }
```

```
                    //MAIN METHOD
        public static void main(String[] args) throws FileNotFoundException

                {
                       Hashtable<String,String> newBlock = null;
                       Stack<Integer> scope_noStack = new Stack<Integer>();

                    String str1 = "C:\\Users\\sakhi\\OneDrive\\Documents\\East
Bay\\Compilers\\sakhi.txt";

                    int count=-1;


                Scanner scan = new Scanner(new File(str1));
                boolean keyexist;
                while(scan.hasNext())
                    {
                     String current_key1 = scan.next();


                if(current_key1.equals("{"))
                      {
                              newBlock = new Hashtable<String, String>();
                              mainHTable.add(newBlock) ;
                              count= count+1;
                              scope_noStack.add(count);
                      }



                   else if(current_key1.equals("}"))
                     {
                              scope_noStack.pop();
                              old_mainHTable.add(newBlock);
                              count= count-1;
                      }


                else

                {
                String token=find_in_current(current_key1,scope_noStack.peek());

                          if(token != null)
                            {

                             System.out.println(token+ "is present in current
                             scope");
                            keyexist = true;
                            }
                           else if(keyexist= false)
```

```
                    token = find_in_All(current_key1);

                    else
                    newBlock = Insert(newBlock,current_key1);

                    }
                }
                Display();
            }
        }
```

Text File:sakhi.txt

```
{ a  A  b  B a

      {   }

      { a  A

      { b  B  c}

 }}
```

Output:

```
a is present in current scope
scope0 has:  b A a B
scope1 has:
scope2 has:  A a
scope3 has:  b
B c
```

Text File:sakhi.txt

```
{   z  x  y x
      {  x y  }
      { a  A
      { b  B  c}
      { ab ba }
 }}
```

Output:

```
x is present in current scope
scope0 has:  z x y
scope1 has:
x y
scope2 has:  a A
scope3 has:  b
B c
scope4 has: ab ba
```