```java
/**
 *  Implements a non-generic hash table.
 *  Note: not full functionality or commented.
 *  @author Nick Howe, Sara Mathieson
 *  @version April 8, 2020
 */
public class Table {

    private TableRow[] rows; // key/value pairs

    /**
     *  Constructor creates a table with
     *  the specified number of slots
     */
    public Table(int tableSize) {
        rows = new TableRow[tableSize];
    }

    /**
     *  Return the default position (index)
     *  where this key is stored
     */
    private int hash(int key) {
        return key % rows.length;
    }

    /**
     *  Locates the position (index) where the
     *  specified key can be found, or where it
     *  should be inserted if it is not already
     *  in the table
     */
    private int locate(int key) {
        int pos = hash(key);

        // this is the linear probing part
        while (rows[pos] != null && rows[pos].key != key) {
            pos = (pos + 1) %rows.length;
        }
        return pos;
    }

    /**
     *  Put the specified value in the table under the
```

```
     *  specified key, returning any previous value
     */
    public String insert(int key, String value) {
        int pos = locate(key);

        String prev = null;
        if (rows[pos] != null)
            prev = rows[pos].value;

        // replace or insert data
        rows[pos] = new TableRow(key, value);
        return prev;
    }

    /**
     *  Retrieve the value associated with the given
     *  key, or null if it doesn't exist
     */
    public String lookup(int key) {
        int pos = locate(key);

        String result = null;
        if (rows[pos] != null)
            result = rows[pos].value;

        return result;
    }

    /**
     *  Class designed to hold the data
     *  in one row of the table
     */
    private class TableRow {

        int key;
        String value;

        /** Constructor to fill in data */
        TableRow(int key, String value) {
            this.key = key;
            this.value = value;
        }
    }
}
```