# Using a HashMap (see documentation)

1. Which method is used to add a key/value pair to a HashMap?
2. Which method is used to retrieve a value if you know the key?
3. How do you find out whether a given key is mapped to a value?
4. What happens when you add a key/value pair and the key is already mapped to a different value?
5. What happens when you add an entry to a map with a value that already exists in the map?
6. What happens when you try to retrieve the value associated with a given key BEFORE that key has been added to the map?

# Lookup table

The `StudentMap` class will be used to associate names with SUN numbers:

- Write the definition of a field of type `HashMap` that can be used to associate student SUN numbers (as a `String`) with student names (also `String`).
- Write the constructor for the `StudentMap` class that initialises your field.

# Mapping SUN numbers to names

The `StudentMap` class will be used to associate names with SUN numbers:

```
1  public class StudentMap
2  {
3
4    private HashMap<String><String> students;
5
6    /**
7     * Constructor for objects of class StudentMap
8     */
9    public StudentMap()
10   {
11     students = new HashMap<String><String>();
12   }
13 }
```

# Adding methods to the `StudentMap` class

Write a method called `addStudent` that has two parameters; a SUN number and a name, both of type `String`, where the SUN number is used as the key.

# Adding a student

```
1  public void addStudent(String sun, String name)
2  {
3    students.put(sun, name);
4  }
```

# Adding methods to the `StudentMap` class

Write a method called `lookupStudent` that has one parameter, a SUN number, and returns the name associated with that name.

# Looking up a student

```
1  public String lookupStudent(String sun)
2  {
3    return students.get(sun);
4  }
```

# Adding methods to the `StudentMap` class

Write a method called `sunExists` which takes one parameter, a SUN number, and returns `true` if there is a name associated with that SUN number and `false` otherwise. name.

# Checking if a student exists

```
1  public boolean sunExists(String sun)
2  {
3    return students.containsKey(sun);
4  }
```

# Adding methods to the `StudentMap` class

Write a method called `numberOfStudents` that returns the number of students that have been added to the `StudentMap`.

# Returning the number of students

```
1  public int numberOfStudents()
2  {
3    return students.size();
4  }
```

# Adding many students

- Write a method called `addAll` that has one parameter, an array of strings in which each string is of the form `"sun,name"`. Use the `split` method of the `String` class to extract each sun/name pair from the array and add that entry. Assume there are no errors in the input.
- The `split` method takes a `String` parameter which is the text to use as the boundary between tokens. It returns the array of strings computed by splitting this string around matches of the given text. For example: `"a:b:c".split(":")` returns an array containing three strings, "a", "b", and "c".

# Processing an array of entries

```
1    public void addAll(String[] data)
2    {
3      for (int i=0; i<data.length; i++) {
4        String[] entry = data[i].split(",");
5        addStudent(entry[0], entry[1]);
6      }
7    }
```