

```
import java.util.HashMap;
import java.util.Map;

public class KeyValueStoreExample {
    public static void main(String... args) {
        // Keep track of age based on name
        Map<String, Integer> age_by_name = new HashMap<>();

        // Store some entries
        age_by_name.put("wilfried", 34);
        age_by_name.put("seppe", 30);
        age_by_name.put("bart", 46);
        age_by_name.put("jeanne", 19);

        // Get an entry
        int age_of_wilfried = age_by_name.get("wilfried");
        System.out.println("Wilfried's age: " + age_of_wilfried);

        // Keys are unique
        age_by_name.put("seppe", 50); // Overrides previous entry
    }
}
```

```

import java.util.ArrayList;
import java.util.List;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.MemcachedClient;

public class MemCachedExample {
    public static void main(String[] args) throws Exception {
        List<String> serverList = new ArrayList<String>() {
            {
                this.add("memcachedserver1.servers:11211");
                this.add("memcachedserver2.servers:11211");
                this.add("memcachedserver3.servers:11211");
            }
        };
        MemcachedClient memcachedClient = new MemcachedClient(
            AddrUtil.getAddresses(serverList));

        // ADD adds an entry and does nothing if the key already exists
        // Think of it as an INSERT
        // The second parameter (0) indicates the expiration - 0 means no expiry
        memcachedClient.add("marc", 0, 34);
        memcachedClient.add("seppe", 0, 32);
        memcachedClient.add("bart", 0, 66);
        memcachedClient.add("jeanne", 0, 19);

        // It is possible to set expiration (the following expires in 3 sec)
        memcachedClient.add("short_lived_name", 3, 19);
        // Sleep 5 sec to make sure our short-lived name is expired
        Thread.sleep(1000 * 5);

        // SET sets an entry regardless of whether it exists
        // Think of it as an UPDATE-OR-INSERT
        memcachedClient.add("marc", 0, 1111); // <- ADD will have no effect
        memcachedClient.set("jeanne", 0, 12); // <- But SET will

        // REPLACE replaces an entry and does nothing if the key does not exist
        // Think of it as an UPDATE
        memcachedClient.replace("not_existing_name", 0, 12); // <- no effect
        memcachedClient.replace("jeanne", 0, 10);

        // DELETE deletes an entry, similar to an SQL DELETE statement
        memcachedClient.delete("seppe");

        // GET retrieves an entry
        Integer age_of_marc = (Integer) memcachedClient.get("marc");
        Integer age_of_short_lived = (Integer) memcachedClient.get(
            "short_lived_name");
        Integer age_of_not_existing = (Integer) memcachedClient.get(
            "not_existing_name");
        Integer age_of_seppe = (Integer) memcachedClient.get("seppe");
        System.out.println("Age of Marc: " + age_of_marc);
        System.out.println("Age of Seppe (deleted): " + age_of_seppe);
        System.out.println("Age of not existing name: " + age_of_not_existing);
        System.out.println(
            "Age of short lived name (expired): " + age_of_short_lived);

        memcachedClient.shutdown();
    }
}

```