MODULE PRACTICE

Go Map Deletions

In Go, key:value pairs can be removed with the delete syntax where a map and key are specified.

```
delete(yourMap, keyValueToDelete)
```

Go Map Creation

In Go, an empty map can be created using make. The type of keys and values are specified.

```
variableName := make(map[keyType]valueType)
```

Go Map Access

In Go, any value can be accessed by specifying the associated key. A second value can be returned to check if the key is in the map.

```
value := yourMap[key]
checkedValue,isFound := yourMap[key]
```

Go Adding to a Map

In Go, additional key:value pairs can be added to the map by specifying the new key and its value.

```
yourMap[newKey] = newValue
```

Go Maps

In Go, a map is an unordered collection of key:value pairs. Keys are used as a way to access a value. Maps are used to associate one piece of data with another.

Example use cases include:

- Counting the number of times unique names appear in a list.
- Mapping simple identifiers, such as an employee id, to related values.
- Anytime we need to associate any piece of data with another!

Go Updating a Map

In Go, existing values can be updated by specifying a key and an updated value.

```
yourMap[existingKey] = newValue
```

Go Map Initialization

In Go, a map can be intialized with key:value data.

```
variableName := map[keyType]valueType{
    key1, value1,
    key2, value2,
    key3, value3,
}
```