Homework 1 Report

Pınar Yücel
CS 315-3
21802188

Dart

Sample Code

```
void main()
{
  // 1. Initialize
  print('1. Initialize');

  // I am trying to test if I can initialize a map. I will see in the following lines if it works.
  var namesToNumbers =
  {
    'James': 1,
    'Mary': 2,
    'John': 3,
    'Jennifer': 4,
    'Daniel': 5
  };

  // 2. Get the value for a given key
  print('\n2. Get the value for a given key ');

  // I am trying to get the value of the 'John' key into johnsNumber variable
  var johnsNumber = namesToNumbers['John'];

  // I am trying to print johnsNumber. If it prints 3, then I successfully got the value for the 'John' key into the johnsNumber
variable in the previous line.
  print("John\'s number is $johnsNumber");

  // 3. Add a new element
  print('\n3. Add a new element ');

  // I am trying to add a new key 'Sarah' and associating it with the value 6
  namesToNumbers['Sarah'] = 6;

  // I am trying to get the value of the 'Sarah' key into sarahsNumber variable
  var sarahsNumber = namesToNumbers['Sarah'];

  // I am trying to print sarahsNumber. If it prints 6, then I successfully added the key 'Sarah' with the value 6 into the map.
  print('Sarah\'s number is $sarahsNumber');

  // 4. Remove an element
  print('\n4. Remove an element');

  // I am trying to print the value of the 'Mary' key. If it prints 2, then it was previously initialized as expected and currently
exists in the map.
  var marysNumber = namesToNumbers['Mary'];
  print('Mary\'s number is $marysNumber');

  // I am trying to remove the 'Mary' key and its value with the remove function. I found this function in Dart API.
  namesToNumbers.remove('Mary');

  // I am trying to print the value of the 'Mary' key again. If it prints null, then I successfully removed 'Mary' key and its
value.
  marysNumber = namesToNumbers['Mary'];
  print('Mary\'s number is $marysNumber');
```

```
  // 5. Modify the value of an existing element
  print('\n5. Modify the value of an existing element');

  // I am trying to print the value of the 'Jennifer' key. If it prints 4, then it was previously initialized as expected and currently exists in the map.
  var jennifersNumber = namesToNumbers['Jennifer'];
  print('Jennifer\'s number is $jennifersNumber');

  // I found out from Dart API that there are more than one ways to modify the value of an existing element. I am trying to set the value of the 'Jennifer' key to value of 10.
  //namesToNumbers['Jennifer'] = 10;
  namesToNumbers.update('Jennifer', (e) => 10);

  // I am trying to get the updated value of the 'Jennifer' key.
  jennifersNumber = namesToNumbers['Jennifer'];

  // If it prints the updated value which is 10, then I successfully updated the value previously.
  print('Jennifer\'s new number is $jennifersNumber');

  // 6. Search for the existence of a key
  print('\n6. Search for the existence of a key');

  // I found from Dart API that there is a function called containsKey which searches for the existence of a key. I know that there is no key named 'Linda'. The function works if it prints the correct information.
  if(namesToNumbers.containsKey('Linda'))
  {
    print('\"Linda\" key exists');
  }
  else
  {
    print('\"Linda\" key does not exist');
  }

  // I know 'John' key exists. I should try for this case too to confirm that the containsKey function works. I will confirm that it works if it prints the correct information.
  if(namesToNumbers.containsKey('John'))
  {
    print('\"John\" key exists');
  }
  else
  {
    print('\"John\" key does not exist');
  }

  // I can also try to search for the existance of a key by checking if it is equal to null. If it prints the correct information then this way works too.
  if(namesToNumbers['Linda'] != null)
  {
    print('\"Linda\" key exists');
  }
  else
  {
    print('\"Linda\" key does not exist');
  }

  if(namesToNumbers['John'] != null)
  {
    print('\"John\" key exists');
```

```
  }
  else
  {
    print('\"John\" key does not exist');
  }

  // 7. Search for the existence of a value
  print('\n7. Search for the existence of a value');

  // I found from Dart API that there is a function called containsValue which searches for the existence of a value. I know that 1 as a value exists. The function works if it prints the correct information.
  if(namesToNumbers.containsValue(1))
  {
    print('1 value exists');
  }
  else
  {
    print('1 value does not exist');
  }

  // I know that 0 does not exist as a value. I should try for this case too to confirm that the containsValue function works. I will confirm that it works if it prints the correct information.
  if(namesToNumbers.containsValue(0))
  {
    print('0 value exists');
  }
  else
  {
    print('0 value does not exist');
  }

  // 8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
  print('\n8. Loop through an associative array');

  // I found from Dart API that there is already a function named toString that will help. I am trying to check what it prints. If it prints the key - value pairs, then it works.
  foo(map)
  {
    // Shortcut
    // print(map.toString());

    map.forEach((k, v)
    {
      print('$k - $v');
    });
  }

  foo(namesToNumbers);
}
```

Result of the Execution

1. Initialize

2. Get the value for a given key
John's number is 3

3. Add a new element
Sarah's number is 6

4. Remove an element
Mary's number is 2
Mary's number is null

5. Modify the value of an existing element
Jennifer's number is 4
Jennifer's new number is 10

6. Search for the existence of a key
"Linda" key does not exist
"John" key exists
"Linda" key does not exist
"John" key exists

7. Search for the existence of a value
1 value exists
0 value does not exist

8. Loop through an associative array
James - 1
John - 3
Jennifer - 10
Daniel - 5
Sarah - 6

Javascript

Sample Code

```
<script>
// 1. Initialize
document.write('1. Initialize<br>')

// I am trying to initialize a map. I will see its correctness later in the code.
let namesToNumbers = new Map([
    ['James', 1],
    ['Mary', 2],
    ['John', 3],
    ['Jennifer', 4],
    ['Daniel', 5]
]);

// 2. Get the value for a given key
document.write('<br>2. Get the value for a given key<br>')

// I am trying to print the value of 'John' key with the get function. If it prints 3, then I successfully got the value with the get function.
document.write('John\'s number is ' + namesToNumbers.get('John') + '<br>');

// 3. Add a new element
document.write('<br>3. Add a new element<br>')

// I am trying to add a new key 'Sarah' and associate it with the value 6 with the set function.
```

```
namesToNumbers.set('Sarah', 6);
// I am trying to print the value of the 'Sarah' key. If it prints 6, then I successfully added the key 'Sarah' with the value 6 into the map.
document.write('Sarah\'s number is ' + namesToNumbers.get('Sarah') + '<br>');

// 4. Remove an element
document.write('<br>4. Remove an element<br>')

// I am trying to print the value of the 'Mary' key with the get function. If it prints 2, then it was previously initialized as expected and currently exists in the map.
document.write('Mary\'s number is ' + namesToNumbers.get('Mary') + '<br>');

// I am trying to delete the 'Mary' key and its value with the delete function.
namesToNumbers.delete('Mary');

// I am trying to print the value of the 'Mary' key again. If it prints undefined, then I successfully removed 'Mary' key and its value.
document.write('Mary\'s number is ' + namesToNumbers.get('Mary') + '<br>');

// 5. Modify the value of an existing element
document.write('<br>5. Modify the value of an existing element<br>')

// I am trying to print the value of the 'Jennifer' key with get function. If it prints 4, then it was previously initialized as expected and currently exists in the map.
document.write('Jennifer\'s number is ' + namesToNumbers.get('Jennifer') + '<br>');

// I am trying to set the value of the 'Jennifer' key to the value of 10 with the set function.
namesToNumbers.set('Jennifer', 10);

// I am trying to get the updated value of the 'Jennifer' key with the set function. If it prints the updated value which is 10, then I successfully updated the value previously.
document.write('Jennifer\'s number is ' + namesToNumbers.get('Jennifer') + '<br>');

// 6. Search for the existence of a key
document.write('<br>6. Search for the existence of a key<br>')

// I found that there is a function called has which searches for the existence of a key. I know that there is no key named 'Linda'. The function works if it prints the correct information.
if(namesToNumbers.has('Linda'))
{
   document.write('\"Linda\" key exists <br>');
}
else
{
   document.write('\"Linda\" key does not exist <br>');
}

// I know 'John' key exists. I should try for this case too to confirm that the has function works. I will confirm that it works if it prints the correct information.
if(namesToNumbers.has('John'))
{
   document.write('\"John\" key exists <br>');
}
else
{
   document.write('\"John\" key does not exist <br>');
}
```

```
// 7. Search for the existence of a value
document.write('<br>7. Search for the existence of a value<br>')

// I found that there is no shortcut to search for the existence of a value. I have to check all of the values to see if 1 as a value exists using values function which returns a new Iterator object that contains the values for each element in the map.I know that 1 as a value exists. The function works if it prints the correct information.
for (let number of namesToNumbers.values())
{
   if(number === 1)
   {
      document.write('1 value exists <br>');
   }
}

// I know that 0 does not exist as a value. I should try for this case too to confirm that I know how to search for the existence of a value. I will confirm that it works if it does not print anything.
for (let number of namesToNumbers.values())
{
   if(number === 0)
   {
      document.write('\"0\" value exists <br>');
   }
}

// 8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
document.write('<br>8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair<br>')

// I found that there is no shortcut to print the elements of a map. I will try to print the elements with a for...of. If it prints then it works.
function foo(map)
{
   for (let [key, value] of map)
   {
      document.write(key + "-" + value + '<br>');
   }

   // Alternative
   //map.forEach(function(value, key) {
   //       document.write(key + ' - ' + value + '<br>')
            //})
}

foo(namesToNumbers);

</script>
```

Result of the Execution

1. Initialize

2. Get the value for a given key
John's number is 3

3. Add a new element
Sarah's number is 6

4. Remove an element
Mary's number is 2
Mary's number is undefined

5. Modify the value of an existing element
Jennifer's number is 4
Jennifer's number is 10

6. Search for the existence of a key
"Linda" key does not exist
"John" key exists

7. Search for the existence of a value
1 value exists

8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
James-1
John-3
Jennifer-10
Daniel-5
Sarah-6

Lua

Sample Code

```
-- 1. Initialize
io.write("1. Initialize\n\n")

--I am trying to initialize a map.
namesToNumbers = {["James"]= 1,
                  ["Mary"] = 2,
                  ["John"] = 3,
                  ["Jennifer"] = 4,
                  ["Daniel"] = 5 }

-- 2. Get the value for a given key
io.write("2. Get the value for a given key\n")

-- I am trying to print the value of 'John' key . If it prints 3, then I successfully got the value.
io.write("John\'s number is " , namesToNumbers["John"], "\n")

-- 3. Add a new element
io.write("\n3. Add a new element\n")

-- I am trying to add a new key 'Sarah' and associate it with the value 6
namesToNumbers["Sarah"] = 6

-- I am trying to print the value of the 'Sarah' key. If it prints 6, then I successfully added the key 'Sarah' with the value 6 into the map.
io.write("Sarah\'s number is " , namesToNumbers["Sarah"], "\n")

-- 4. Remove an element
io.write("\n4. Remove an element\n")

-- I am trying to print the value of the 'Mary' key. If it prints 2, then it was previously initialized as expected and currently exists in the map.
```

```
io.write("Mary\'s number is " , namesToNumbers["Mary"], "\n")

-- I am trying to delete the 'Mary' key and its value from the map.
namesToNumbers["Mary"] = nil

-- I am trying to print the value of the 'Mary' key again. If it prints nil, then I successfully removed 'Mary' key and its value.
io.write("Mary\'s number is ")
print(namesToNumbers["Mary"])

-- 5. Modify the value of an existing element
io.write("\n5. Modify the value of an existing element\n")

-- I am trying to print the value of the 'Jennifer' key. If it prints 4, then it was previously initialized as expected and currently exists in the map
io.write("Jennifer\'s number is " , namesToNumbers["Jennifer"], "\n")

-- I am trying to set the value of the 'Jennifer' key to the value of 10.
namesToNumbers["Jennifer"] = 10

-- I am trying to get the updated value of the 'Jennifer' key. If it prints the updated value which is 10, then I successfully updated the value previously.
io.write("Jennifer\'s number is " , namesToNumbers["Jennifer"], "\n")


-- 6. Search for the existence of a key
io.write("\n6. Search for the existence of a key\n")

--I know that there is no key named "Linda". The function works if it prints the correct information.
if namesToNumbers["Linda"] ~= nil then
   print("\"Linda\" key exists")
else
   print("\"Linda\" key does not exist")
end

-- I know 'John' key exists. I should try for this case too to confirm that I have learned how to search for the existence of a key. I will confirm that I have learnt it if it prints the correct information.
if namesToNumbers["John"] ~= nil then
   print("\"John\" key exists")
else
   print("\"John\" key does not exist")
end

-- 7. Search for the existence of a value
io.write("\n7. Search for the existence of a value\n")

-- I found that there is no shortcut to search for the existence of a value. I have to check all of the values to see if 1 as a value exists.I know that 1 as a value exists. This approach works if it prints the correct information.
for key,value in pairs(namesToNumbers) do
   if value == 1 then
      print("1 value exists")
   end
end

-- I know that 0 does not exist as a value. I should try for this case too to confirm that I know how to search for the existence of a value. I will confirm that it works if it does not print anything.
for key,value in pairs(namesToNumbers) do
   if value == 0 then
      print("0 value exists")
```

```
    end
end

-- 8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
io.write("\n8. Loop through an associative array\n")

-- I found that there is no shortcut to print the elements of a map. I will try to print the elements with a for. If it prints then it works.

function foo(t)
    for key,value in pairs(namesToNumbers) do
        io.write(key, "-", value, "\n")
    end
end

foo(namesToNumbers)
```

Result of the Execution

```
1. Initialize

2. Get the value for a given key
John's number is 3

3. Add a new element
Sarah's number is 6

4. Remove an element
Mary's number is 2
Mary's number is nil

5. Modify the value of an existing element
Jennifer's number is 4
Jennifer's number is 10

6. Search for the existence of a key
"Linda" key does not exist
"John" key exists

7. Search for the existence of a value
1 value exists

8. Loop through an associative array
James-1
Sarah-6
Jennifer-10
Daniel-5
John-3
```

PHP

Sample Code

```
<!DOCTYPE html>
<html>
<body>
```

```php
<?php

// 1. Initialize
echo '1. Initialize<br>';

// I am trying to initialize a map. I will see its correctness later in the code.
$namesToNumbers = ['James' => 1,
                                        'Mary' => 2,
                'John' => 3,
                'Jennifer' => 4,
                'Daniel' => 5];

// 2. Get the value for a given key
echo '<br>2. Get the value for a given key<br>';

// I am trying to print the value of 'John' key. If it prints 3, then I successfully got the value.
echo "John's number is " . $namesToNumbers['James'] . "<br>";

// 3. Add a new element
echo '<br>3. Add a new element<br>';

// I am trying to add a new key 'Sarah' and associate it with the value 6. I found that there are two ways to do that. I will test each of them with echo later.
//$namesToNumbers['Sarah'] = 6;
$namesToNumbers += ['Sarah' => 6];

// I am trying to print the value of the 'Sarah' key. If it prints 6, then I successfully added the key 'Sarah' with the value 6 into the map.
echo "Sarah's number is " . $namesToNumbers['Sarah'] . "<br>";

// 4. Remove an element
echo '<br>4. Remove an element<br>';

// I am trying to print the value of the 'Mary' key. If it prints 2, then it was previously initialized as expected and currently exists in the map.
echo "Mary's number is " . $namesToNumbers['Mary'] . "<br>";

// I am trying to delete the 'Mary' key and its value with the unset function.
unset($namesToNumbers['Mary']);

// I am trying to print the value of the 'Mary' key again. If it prints undefined, then I successfully removed 'Mary' key and its value.
echo "Mary's number is " . $namesToNumbers['Mary'] . "<br>";

// 5. Modify the value of an existing element
echo '<br>5. Modify the value of an existing element<br>';

// I am trying to print the value of the 'Jennifer' key. If it prints 4, then it was previously initialized as expected and currently exists in the map.
echo "Jennifer's number is " . $namesToNumbers['Jennifer'] . "<br>";

// I am trying to set the value of the 'Jennifer' key to the value of 10.
$namesToNumbers['Jennifer'] = 10;

// I am trying to get the updated value of the 'Jennifer' key. If it prints the updated value which is 10, then I successfully updated the value previously.
echo "Jennifer's number is " . $namesToNumbers['Jennifer'] . "<br>";
```

```
// 6. Search for the existence of a key
echo '<br>6. Search for the existence of a key<br>';

// I found that there is a function called array_key_exists which searches for the existence of a key. I know that there is no key named 'Linda'. The function works if it prints the correct information.
if(array_key_exists('Linda', $namesToNumbers))
{
   echo '"Linda" key exists <br>';
}
else
{
   echo '"Linda" key does not exist <br>';
}

// I know 'John' key exists. I should try for this case too to confirm that the array_key_exists function works. I will confirm that it works if it prints the correct information.
if(array_key_exists('John', $namesToNumbers))
{
   echo '"John" key exists <br>';
}
else
{
   echo '"John" key does not exist <br>';
}

// 7. Search for the existence of a value
echo '<br>7. Search for the existence of a value<br>';

// I found that there is a function called in_array to search for the existence of a value. I know that 1 as a value exists in the map. It works if it prints the correct information.
if(in_array(1, $namesToNumbers, true))
{
   echo '1 value exists <br>';
}
else
{
   echo '1 value does not exist <br>';
}

// I know 0 as a value does not exist in the map. I should try for this case too to confirm that the in_array function works. I will confirm that it works if it prints the correct information.
if(in_array(0, $namesToNumbers, true))
{
   echo '0 value exists <br>';
}
else
{
   echo '0 value does not exist <br>';
}

// 8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
echo '<br>8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair<br>';

// I found that there is no shortcut to print the elements of a map. I will try to print the elements with foreach. If it prints then it works.
function foo($map)
{
   foreach ($map as $k=>$v)
```

```
    {
        echo "$k-$v <br>";
    }

    // Alternative way. It doesnt look very nice when printed like this.
    // print_r($namesToNumbers);
}

foo($namesToNumbers);

?>

</body>
</html>
```

Result of the Execution

1. Initialize

2. Get the value for a given key
John's number is 1

3. Add a new element
Sarah's number is 6

4. Remove an element
Mary's number is 2
Mary's number is

5. Modify the value of an existing element
Jennifer's number is 4
Jennifer's number is 10

6. Search for the existence of a key
"Linda" key does not exist
"John" key exists

7. Search for the existence of a value
1 value exists
0 value does not exist

8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
James-1
John-3
Jennifer-10
Daniel-5
Sarah-6

Python

Sample Code

```
# 1. Initialize
print('1. Initialize')

# I am trying to test if I can initialize a map. I will see in the following lines if it works.
namesToNumbers = {
```

```
    'James': 1,
    'Mary': 2,
    'John': 3,
    'Jennifer': 4,
    'Daniel': 5
}

# 2. Get the value for a given key
print('\n2. Get the value for a given key ')

# I am trying to print the value of the "John" key with the get function. If it prints 3, then I successfully got the value for the 'John' key.
print('John\'s number is ', end = "")
print(namesToNumbers.get("John"))

# I am trying if this works too. It works if it prints 3.
# print('John\'s number is', namesToNumbers["John"])

# 3. Add a new element
print('\n3. Add a new element ')

# I am trying to add a new key "Sarah" and associating it with the value 6 with the update function.
namesToNumbers.update({'Sarah': 6})

# I am trying an alternative way.
# namesToNumbers['Sarah'] = 6

# I am trying to print the value of they "Sarah" key. If it prints 6, then I successfully added the key "Sarah" with the value 6 into the map.
print('Sarah\'s number is', namesToNumbers["Sarah"])

# 4. Remove an element
print('\n4. Remove an element')

# I am trying to print the value of the 'Mary' key. If it prints 2, then it was previously initialized as expected and currently exists in the map.
print('Mary\'s number is', namesToNumbers["Mary"])

# I am trying to remove the "Mary" key and its value with the pop function.
namesToNumbers.pop("Mary")

# I am trying to print the value of the "Mary" key again. If it is error, then I successfully removed 'Mary' key and its value.
# print('Mary\'s number is', namesToNumbers["Mary"])

# 5. Modify the value of an existing element
print('\n5. Modify the value of an existing element');

# I am trying to print the value of the 'Jennifer' key. If it prints 4, then it was previously initialized as expected and currently exists in the map.
print('Jennifer\'s number is', namesToNumbers["Jennifer"])

# I am trying to modify the value of the 'Jennifer' key
namesToNumbers['Jennifer'] = 10

# Alternative way
# namesToNumbers.update({'Jennifer': 10})

# If it prints 10, then I modified its value
```

```
print('Jennifer\'s number is', namesToNumbers["Jennifer"])

# 6. Search for the existence of a key
print('\n6. Search for the existence of a key');

# I am trying to search for a key that I know that does not exist. I am using a function named keys() which returns a list containing the dictionary's keys.
if "Linda" in namesToNumbers.keys():
	print('\"Linda\" key exists')
else:
	print('\"Linda\" key does not exist')

# I know 'John' key exists. I should try for this case too to confirm that the keys function works. I will confirm that it works if it prints the correct information.
if "John" in namesToNumbers.keys():
	print('\"John\" key exists')
else:
	print('\"John\" key does not exist')

# 7. Search for the existence of a value
print('\n7. Search for the existence of a value');

# I am trying to search for a value that I know that does exist. I am using a function named values which returns a list containing the dictionary's values.
if 1 in namesToNumbers.values():
	print('1 value exists')
else:
	print('1 value does not exist')

# I know that 0 does not exist as a value. I should try for this case too to confirm that the values function works. I will confirm that it works if it prints the correct information.
if 0 in namesToNumbers.values():
	print('0 value exists')
else:
	print('0 value does not exist')

# 8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
print('\n8. Loop through an associative array');

# I am trying to print the elements with the help of the items function with a for loop. If it prints then it works.
def foo(d):
	for k, v in d.items():
		print(k, "-", v)

foo(namesToNumbers)
```

Result of the Execution

1. Initialize

2. Get the value for a given key
John's number is 3

3. Add a new element
Sarah's number is 6

4. Remove an element
Mary's number is 2

5. Modify the value of an existing element
Jennifer's number is 4
Jennifer's number is 10

6. Search for the existence of a key
"Linda" key does not exist
"John" key exists

7. Search for the existence of a value
1 value exists
0 value does not exist

8. Loop through an associative array
James - 1
John - 3
Jennifer - 10
Daniel - 5
Sarah - 6

Ruby

Sample Code

```
# 1. Initialize
print '1. Initialize', "\n"

# I am trying to test if I can initialize a map. I will see in the following lines if it works.
namesToNumbers =
{
   'James'=> 1,
   'Mary'=> 2,
   'John'=> 3,
   'Jennifer'=> 4,
   'Daniel'=> 5
};

# 2. Get the value for a given key
print "\n", '2. Get the value for a given key ', "\n"

# I am trying to print the value of the "John" key. If it prints 3, then I successfully got the value for the 'John' key.
print 'John\'s number is ', namesToNumbers["John"], "\n"

# I am trying the same thing with the fetch function. If it prints 3, then I successfully got the value for the 'John' key.
# print 'John\'s number is ', namesToNumbers.fetch("John"), "\n"

# 3. Add a new element
print "\n", '3. Add a new element ', "\n"

# I am trying to add a new key "Sarah" and associating it with the value 6.
namesToNumbers['Sarah'] = 6

# I am trying to print the value of they "Sarah" key. If it prints 6, then I successfully added the key "Sarah" with the value 6
into the map.
print 'Sarah\'s number is ', namesToNumbers.fetch("Sarah"), "\n"
```

```
# 4. Remove an element
print "\n", '4. Remove an element ', "\n"

# I am trying to print the value of the 'Mary' key. If it prints 2, then it was previously initialized as expected and currently exists in the map.
print 'Mary\'s number is ', namesToNumbers["Mary"], "\n"

# I am trying to remove the "Mary" key and its value with the delete function.
namesToNumbers.delete("Mary")

# I am trying to print the value of the "Mary" key again. If it prints nothing, then I successfully removed 'Mary' key and its value.
print 'Mary\'s number is ', namesToNumbers["Mary"], "\n"

# 5. Modify the value of an existing element
print "\n", '5. Modify the value of an existing element ', "\n"

# I am trying to print the value of the 'Jennifer' key. If it prints 4, then it was previously initialized as expected and currently exists in the map.
print 'Jennifer\'s number is ', namesToNumbers["Jennifer"], "\n"

# I am trying to modify the value of the 'Jennifer' key
namesToNumbers['Jennifer'] = 10

# If it prints 10, then I modified its value
print 'Jennifer\'s number is ', namesToNumbers["Jennifer"], "\n"

# 6. Search for the existence of a key
print "\n", '6. Search for the existence of a key ', "\n"

# I am trying to search for a key that I know that does not exist. I am using a function named key?. If it prints the correct information then it works.
if namesToNumbers.key?("Linda")
        print '"Linda" key exists', "\n"
else
        print '"Linda" key does not exist', "\n"
end

# I am trying to search for a key that I know that does exist. If it prints the correct information then it works.
if namesToNumbers.key?("John")
        print '"John" key exists', "\n"
else
        print '"John" key does not exist', "\n"
end

# 7. Search for the existence of a value
print "\n", '7. Search for the existence of a value ', "\n"

# I am trying to search for a value that I know that does exist. I am using a function named value?. If it prints the correct information then it works.
if namesToNumbers.value?(1)
        print '1 value exists', "\n"
else
        print '1 value does not exist', "\n"
end

# I am trying to search for a key that I know that does not exist. If it prints the correct information then it works.
```

```
if namesToNumbers.value?(0)
        print '0 value exists', "\n"
else
        print '0 value does not exist', "\n"
end

# 8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
print "\n", '8. Loop through an associative array ', "\n"

# I am trying to print the elements with the help of the inspect function
def foo(map)
   # Shortcut
   #print map.inspect, "\n"

    map.each {|k, x| puts "#{k} - #{x}"}
end

foo(namesToNumbers)
```

Result of the Execution

```
1. Initialize

2. Get the value for a given key
John's number is 3

3. Add a new element
Sarah's number is 6

4. Remove an element
Mary's number is 2
Mary's number is

5. Modify the value of an existing element
Jennifer's number is 4
Jennifer's number is 10

6. Search for the existence of a key
"Linda" key does not exist
"John" key exists

7. Search for the existence of a value
1 value exists
0 value does not exist

8. Loop through an associative array
James - 1
John - 3
Jennifer - 10
Daniel - 5
Sarah - 6
```

Rust

Sample Code

```
fn main()
{
  // 1. Initialize
  println!("1. Initialize");

  use std::collections::HashMap;

  // I am trying to declare a map.
  let mut names_to_numbers = HashMap::new();

  // I am trying to initialize the map. I will see if it works later in the code.
  names_to_numbers.insert("James", 1);
  names_to_numbers.insert("Mary", 2);
  names_to_numbers.insert("John", 3);
  names_to_numbers.insert("Jennifer", 4);
  names_to_numbers.insert("Daniel", 5);

  // 2. Get the value for a given key
  println!("\n2. Get the value for a given key");

  // I am trying to get the value of the John key. It works if it prints 3.
  println!("John's number is {}", names_to_numbers["John"]);

  // 3. Add a new element
  println!("\n3. Add a new element");

  // This is error
  //names_to_numbers["Sarah"] = 6;

  // I am trying to insert a new key with the insert function.
  names_to_numbers.insert("Sarah", 6);

  // It means that the new key was successfully inserted if it prints 6.
  println!("Sarah's number is {}", names_to_numbers["Sarah"]);

  // 4. Remove an element
  println!("\n4. Remove an element");

  // I am first checking if the key was added successfuly on initialization.
  println!("Mary's number is {}", names_to_numbers["Mary"]);

  // I am trying to remove the element with the remove function.
  names_to_numbers.remove("Mary");

  // I am trying to get the value of the key again to see what happened. If it gives error or something like that, then the
removal was successful.
  //println!("Mary's number is {}", names_to_numbers["Mary"]);

  // 5. Modify the value of an existing element
  println!("\n5. Modify the value of an existing element");

  // I am trying to check if the key was added successfully before. It was if it prints 4.
  println!("Jennifer's number is {}", names_to_numbers["Jennifer"]);

  // I am trying to modify.
  names_to_numbers.insert("Jennifer", 10);

  // If it prints 10 then I successfully modified.
```

```
println!("Jennifer's number is {}", names_to_numbers["Jennifer"]);

// 6. Search for the existence of a key
println!("\n6. Search for the existence of a key");

// I am trying to search for a key that I know that does not exist. I am using a function named contains_key. It works if it prints the correct info.
if names_to_numbers.contains_key("Linda")
{
  println!("Linda key exists");
}
else
{
  println!("Linda key does not exist");
}

// Now, I am trying the same thing with a key I know exists. It works if it prints the correct info.
if names_to_numbers.contains_key("John")
{
  println!("John key exists");
}
else
{
  println!("John key does not exist");
}

// 7. Search for the existence of a value
println!("\n7. Search for the existence of a value");

// I am trying to check if 1 as a value exists.
let does_contain_1 = names_to_numbers.values().any(|&val| val == 1);

// I am trying to check if 0 as a value exists.
let does_contain_0 = names_to_numbers.values().any(|&val| val == 0);

// It works if it prints the correct info.
if does_contain_1
{
  println!("1 value exists");
}
else
{
  println!("1 value does not exist");
}

// It works if it prints the correct info.
if does_contain_0
{
  println!("0 value exists");
}
else
{
  println!("0 value does not exist");
}

// 8. Loop through an associative array, apply a function, called foo, which simply prints the key-value pair
println!("\n8. Loop through an associative array");
```

```
 // I am tring to print the elements. It works if it prints the pairs.
 fn foo(h: HashMap<&str, i32>)
 {
  // Shortcut
  //println!("{:?}", h);

  // Alternative way
  for (key, value) in &h {
     println!("{} - {}", key, value);
  }
 }

 foo(names_to_numbers);

}
```

Result of the Execution

1. Initialize

2. Get the value for a given key
John's number is 3

3. Add a new element
Sarah's number is 6

4. Remove an element
Mary's number is 2

5. Modify the value of an existing element
Jennifer's number is 4
Jennifer's number is 10

6. Search for the existence of a key
Linda key does not exist
John key exists

7. Search for the existence of a value
1 value exists
0 value does not exist

8. Loop through an associative array
Jennifer - 10
Sarah - 6
James - 1
John - 3
Daniel - 5

Evaluation of the Languages

Evaluation of Dart

Initialization of a map in Dart language is readable and writeable because its simple. Getting the value for a given key is also simple and error-free and it can be done according to array conventions and

therefore readable and writeable. There is no function like add(key, value) to add a new element into a map. Adding can be done as if modifying an element and therefore not really readable. One can accidentally add a new element instead of modifying an element easily. This makes it less writeable. It is also hard to read whether it is a modify or add. Since there is a function called remove to remove an element, it is both readable because of the simplicity and writeable because error-free. There are different ways to modify an element which makes it less readable because you have to know them both if both are used to understand the code. Same applies to writeability because there is more to be learned. For checking the existance of a key or value, functions are present. They are easy to use and read and understand and therefore contribute to both readability and writeability. One can also try to search for the existance of a key by checking if it is equal to null which is a less readable and writeable way. To print the elemets, for each can be used which simple so readable. There is also a shortcut for printing.

Evaluation of Javascript

Initialization of a map in Javascript language is readable and writeable because its simple. Getting and setting the value of a spesific key can be made by get and set function which makes the language less error prone and simple and therefore readable and writeable. However, there is no function like add(key, value) to add a new element into a map. Adding is done as if modifying an element and therefore not really readable. One can accidentally add a new element instead of modifying an element easily. This makes it less writeable. It is also hard to read whether it is a modify or add. Since there is a function called delete to remove an element, it is both readable because of the simplicity and writeable because error-free. For searching for the existence of a key, there is a function named has which has an intuitive use and simple and therefore contribute to the readability and writeability. However, there is no simple function to search for the existence of a value. It makes it less writeable because it would be less error prone if there were simple functions for these operations. A simple for loop can be used to print the elements which is readable.

Evaluation of Lua

Initialization of a map in Lua language is readable and writeable because its simple. Getting the value for a given key is also simple and error-free and is done as according to array conventions and therefore readable and writeable. There is no function like add(key, value) to add a new element into a map. Adding can be done as if modifying an element and therefore not really readable. One can accidentally add a new element instead of modifying an element easily. This makes it less writeable. It is also hard to read whether it is a modify or add. To remove an element, one needs to set it equal to nil. This error-prone and non simple because it both not meaningless and requires you to memorize which makes it less readable and writeable. To search for the existence of a key you need to check if its value is equal to nil. This is again the same situation as in removing. You have to iterate through every value to search for the existence of a value. This is not simple and error prone so minus to writeability and readability. To print every key value pair, you can use a simpe for loop.

Evaluation of PHP

Initialization of a map in PHP is kind of not simple because of the use of the non simple symbol “=>”. This is a minus for both readability and also writeability because people can accidentally use “:” or “=”. Getting the value for a given key is simple and error-free and is done as according to array conventions and therefore readable and writeable. There are more than one ways to add an element into a map in

PHP. The use of the first way is the same as modifying an element which may cause errors and make the code less readable as the same format is used in modifying. Although the other way is different than as if modifying, it is not very readable (see code) and error prone. There is a function to remove an element but its name is unset which I find unintuitive. People have to memorize its name to use it and because of this, it makes the language less readable and writeable because its error-prone. There are functions for both searching for the existence of a key and value which makes the language more readable and writeable in terms of associative arrays. In terms of printing, you can use print_r but the thing it prints it not very readable. Foreach can also be used, but since it is not a shortcut, it makes it less writeable because its error prone.

Evaluation of Python

Initialization of a map in Python language is readable and writeable because its simple. There are more than one ways to get the value of a key. You can use the get function or get as conventionally with [] for example. Both of these ways are intuitive and easy to write and understand and simple and therefore contribute to writeability and readability. You can use a function named update to add a new element or add with [] (see code). It is not intuitive to have function named update to add a new element so I think its a minus for writeability and readability. Same applies for the [] way of adding. Removing can be done with pop function which is understandable and simple. One can modify the value of a key with both [] operator and update function which is readable because its simple and writeable because its error prone. There are functions called keys and values with which keys and values can be iterated to search for their existences. I think since they are simple to use they are readable and writeable. You can use a for loop to print the elements.

Evaluation of Ruby

Initialization of a map in Ruby is kind of not simple because of the use of the non simple symbol “=>”. This is a minus for both readability and also writeability because people can accidentally use “:” or “=”. Getting the value of a key can be done either by the use of [] or fetch function. First one is the conventional way hence readable and writeable because people are less likely to have an error regarding that. Second one is intuitive and simple so readable but I think is not writeable because people can misremember “fetch” and get errors so its error prone. Adding an element can be made as if modifying so it can be confusing and error prone hence non readable and non writeable. Delete function can be used to delete an element which is readable and also writeable as its simple and error prone. There are functions for searching for a spesific key and value with meaningful names and hence they contribute to both writeability and readability. Hash can easily be printed with .inspect. Each function can also be used simplty (see code).

Evaluation of Rust

Initialization is readable because you understand easily in Rust. However, it is not easy to write and therefore non writeable. Value of a key can be gotten according to the conventional array mechanics, this is a plus for both readability and writeability. You can both use insert to modify and add an element which may cause confusion and therefore bad for readability. Removing can be made with a function named remove which is intuitive and simple therefore contributes to both readability and writeability. There is a function for checking the existence of a key whose name is simple and intuitive and therefore it contributes to readability and also writeability because its error-free. There is no shortcut

for checking for the existence of a value which is a minus for both readability and writeability. Printing is easy with a for loop so its good for writeability. There is also a shortcut to print (see code).

Best Language

When I consider every evaluation I have made, none of the languages are perfect in terms of writeability and readability. Many of the languages share some common problems such as modifying and adding in the same way. However, the best language seems to be Dart for associative array operations. First of all, its initialization is simple. Removal and updating is intuitive. It has shortcut functions to search for the existence of a key or value. Printing is simple to read and write with foreach as well. Overall, it is the best language in terms of readability and writeability in the context of associative arrays for me.

Learning Strategy

I used Google and Youtube to learn how associative arrays are implemented in the specified languages. I tried to use the websites that are the offical websites of the languages. I reported every experiment I performed as comments in the code. I did not continue experimenting without completely understanding an operation. I did not try anything random to see if it would work. I followed the information that I found on the internet. Since online compilers/interpreters were allowed, I just tested my code on the online compilers/interpreters whose link can be found below. According to those, outputs are as they should be.

Online Compilers/Interpreters
Dart : https://dartpad.dev/
Javascript : https://js.do/
Lua : https://www.jdoodle.com/execute-lua-online/
PHP : https://www.w3schools.com/php/phptryit.asp?filename=tryphp_compiler
Python : https://www.programiz.com/python-programming/online-compiler/
Ruby : https://www.onlinegdb.com/online_ruby_compiler
Rust : https://play.rust-lang.org/