

CORE JAVA - Day 58

Agenda

- **size() method**



size() method: size() method of HashMap class is used to get the size of the map which refers to the number of the key-value pair or mappings in the Map.

Example: size() method in HashMap

```
import java.util.HashMap;

public class MapIntro {

    public static void main(String[] args) {
        //Create an empty HashMap
        HashMap<String, String> myDetails = new HashMap<String, String>();

        //Adding elements to the HashMap
        myDetails.put("FirstName", "Somanna");
        myDetails.put("Surname", "Somanna");
        myDetails.put("Phone number", "8965474562");
        myDetails.put("Password", "##90$$");
        myDetails.put("DOB", "20/02/1947");
        myDetails.put("Gender", "Male");

        //size() method returns the number of key value pairs
        System.out.println(myDetails.size());
    }
}
```

Output:

Command Prompt

```
$java MapIntro.java  
6
```

isEmpty() method: isEmpty() method of HashMap class is used to check for the emptiness of the map. The method returns true if no key-value pair or mapping is present in the map else false.

Example: isEmpty() method

```
import java.util.HashMap;  
  
public class MapIntro {  
  
    public static void main(String[] args) {  
        //Create an empty HashMap  
        HashMap<String, String> myDetails = new HashMap<String, String>();  
  
        System.out.println("Before Adding the elements");  
  
        //Check if HashMap is empty or not  
        System.out.println(myDetails.isEmpty());  
  
        //Adding elements to the HashMap  
        myDetails.put("FirstName", "Somanna");  
        myDetails.put("Surname", "Somanna");  
        myDetails.put("Phone number", "8965474562");  
        myDetails.put("Password", "##90$$");  
        myDetails.put("DOB", "20/02/1947");  
        myDetails.put("Gender", "Male");  
  
        System.out.println("After Adding the elements");  
  
        //Check if HashMap is empty or not  
        System.out.println(myDetails.isEmpty());  
    }  
}
```

Output:

Command Prompt

```
$java MapIntro.java  
Before Adding the elements  
true  
After Adding the elements  
false
```

In the above example, we have created a hashmap named myDetails. Here, we have used the isEmpty() method to check whether the hashmap contains any elements or not. Initially, the newly created hashmap does not contain any element. Hence, isEmpty() returns true. However, after we add some elements, the method returns false.

containsKey(key) method: containsKey() method is used to check whether a particular key is being mapped into the HashMap or not. It takes the key element as a parameter and returns True if that element is mapped in the map.

Example: containsKey(key) method

```
import java.util.HashMap;

public class MapIntro {


    public static void main(String[] args) {
        //Create an empty HashMap
        HashMap<String, String> myDetails = new HashMap<String, String>();

        //Adding elements to the HashMap
        myDetails.put("FirstName", "Somanna");
        myDetails.put("Surname", "Somanna");
        myDetails.put("Phone number", "8965474562");
        myDetails.put("Password", "##90$$");
        myDetails.put("DOB", "20/02/1947");
        myDetails.put("Gender", "Male");

        //checking for key element Surname
        System.out.println(myDetails.containsKey("Surname"));

        //checking for key element Lastname
        System.out.println(myDetails.containsKey("Lastname"));
    }
}
```

Output:

 Command Prompt

```
$java MapIntro.java
true
false
```

Here, the Hashmap contains a mapping for the key Surname and doesn't contains a key Lastaname. Hence, the containskey() method returns true first then it will return false.

containsValue(value) method: containsValue() method is used to check whether a particular value is being mapped by a single or more than one key in the HashMap. It takes the Value as a parameter and returns true if that value is mapped by any of the key in the map.

Example: containsValue(value) method

```
import java.util.HashMap;

public class MapIntro {

    public static void main(String[] args) {
        //Create an empty HashMap
        HashMap<String, String> myDetails = new HashMap<String, String>();

        //Adding elements to the HashMap
        myDetails.put("FirstName", "Somanna");
        myDetails.put("Surname", "Somanna");
        myDetails.put("Phone number", "8965474562");
        myDetails.put("Password", "##90$$");
        myDetails.put("DOB", "20/02/1947");
        myDetails.put("Gender", "Male");

        //checking for value Somanna
        System.out.println(myDetails.containsValue("Somanna"));

        //checking for value Alex
        System.out.println(myDetails.containsKey("Alex"));
    }
}
```

Output:

Command Prompt

```
$java MapIntro.java
true
false
```

Here, the Hashmap contains a mapping for the value Somanna and doesn't contains a key Alex. Hence, the containsValue() method returns true first then it will return false.

remove(key) method with key: remove() is an inbuilt method of HashMap class and is used to remove the mapping of any particular key from the map. It basically removes the values for any particular key in the Map.

Example: remove() method

```
import java.util.HashMap;

public class MapIntro {

    public static void main(String[] args) {
        //Create an empty HashMap
        HashMap<String, String> myDetails = new HashMap<String, String>();

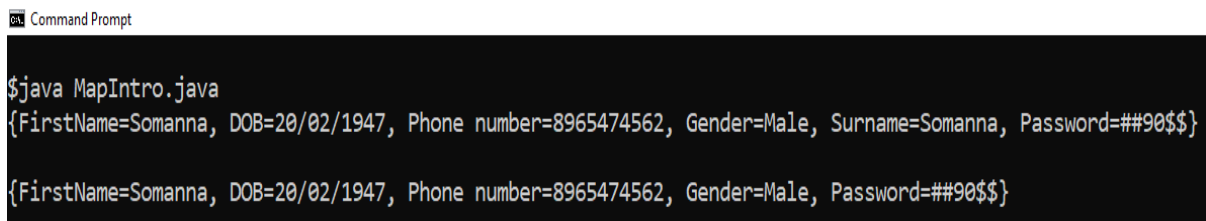
        //Adding elements to the HashMap
        myDetails.put("FirstName", "Somanna");
        myDetails.put("Surname", "Somanna");
        myDetails.put("Phone number", "8965474562");
        myDetails.put("Password", "##90$$");
        myDetails.put("DOB", "20/02/1947");
        myDetails.put("Gender", "Male");

        System.out.println(myDetails);

        //removing the mapping with the key
        myDetails.remove("Surname");

        System.out.println();
        System.out.println(myDetails);
    }
}
```

Output:



```
Command Prompt

$java MapIntro.java
{FirstName=Somanna, DOB=20/02/1947, Phone number=8965474562, Gender=Male, Surname=Somanna, Password=##90$$}

{FirstName=Somanna, DOB=20/02/1947, Phone number=8965474562, Gender=Male, Password=##90$$}
```

Here, if you observe the output remove() method have removed surname and it's respective value from the map. remove() method not only remove the map it also return the value as shown in below example.

Example:

```
import java.util.HashMap;

public class MapIntro {


    public static void main(String[] args) {
        //Create an empty HashMap
        HashMap<String, String> myDetails = new HashMap<String, String>();

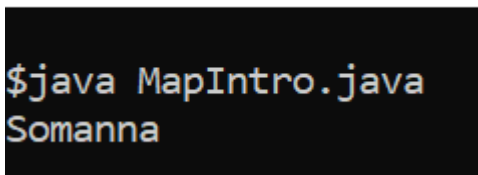
        //Adding elements to the HashMap
        myDetails.put("FirstName", "Somanna");
        myDetails.put("Surname", "Somanna");
        myDetails.put("Phone number", "8965474562");
        myDetails.put("Password", "##90$$");
        myDetails.put("DOB", "20/02/1947");
        myDetails.put("Gender", "Male");

        //removing the mapping with the key
        String value = myDetails.remove("Surname");

        System.out.println(value);
    }
}
```

Output:

 Command Prompt



```
$java MapIntro.java
Somanna
```

Example:

```
import java.util.HashMap;

public class MapIntro {

    public static void main(String[] args) {
        //Create an empty HashMap
        HashMap<String, String> myDetails = new HashMap<String, String>();

        //Adding elements to the HashMap
        myDetails.put("FirstName", "Somanna");
        myDetails.put("Surname", "Somanna");
        myDetails.put("Phone number", "8965474562");
        myDetails.put("Password", "##90$$");
        myDetails.put("DOB", "20/02/1947");
        myDetails.put("Gender", "Male");

        //removing the mapping with the key
        String value = myDetails.remove("Lastname");

        System.out.println(value);
    }
}
```

Output:

Command Prompt

```
$java MapIntro.java  
null
```

Here, we are getting null as the output because the specified key i.e Lastname is not present in the map so it returns null.

remove(key, value) method with key and value: Removes the entry for the specified key only if it is currently mapped to the specified value.

Example:

```
import java.util.HashMap;  
  
public class MapIntro {  
  
    public static void main(String[] args) {  
        //Create an empty HashMap  
        HashMap<String, String> myDetails = new HashMap<String, String>();  
  
        //Adding elements to the HashMap  
        myDetails.put("FirstName", "Somanna");  
        myDetails.put("Surname", "Somanna");  
        myDetails.put("Phone number", "8965474562");  
        myDetails.put("Password", "##90$$");  
        myDetails.put("DOB", "20/02/1947");  
        myDetails.put("Gender", "Male");  
  
        System.out.println(myDetails);  
  
        //removing the mapping with the key and value  
        myDetails.remove("Surname", "mg");  
        System.out.println();  
  
        System.out.println(myDetails);  
    }  
}
```

Output:

Command Prompt

```
$java MapIntro.java  
{FirstName=Somanna, DOB=20/02/1947, Phone number=8965474562, Gender=Male, Surname=Somanna, Password=##90$$}  
{FirstName=Somanna, DOB=20/02/1947, Phone number=8965474562, Gender=Male, Surname=Somanna, Password=##90$$}
```

Here, remove(key, value) method will remove key and value pair if the specified key value pair present inside the map.

Example:

```
import java.util.HashMap;

public class MapIntro {

    public static void main(String[] args) {
        //Create an empty HashMap
        HashMap<String, String> myDetails = new HashMap<String, String>();

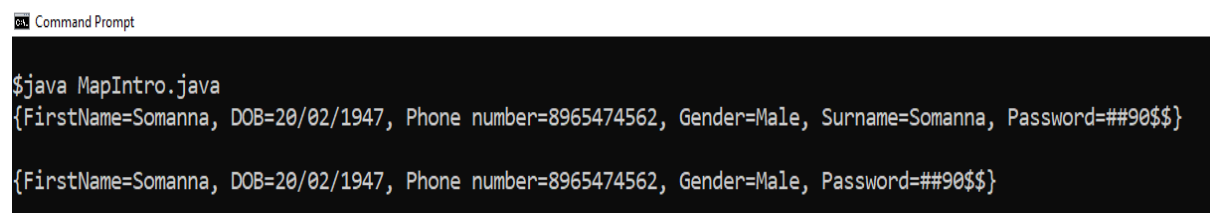
        //Adding elements to the HashMap
        myDetails.put("FirstName", "Somanna");
        myDetails.put("Surname", "Somanna");
        myDetails.put("Phone number", "8965474562");
        myDetails.put("Password", "##90$$");
        myDetails.put("DOB", "20/02/1947");
        myDetails.put("Gender", "Male");

        System.out.println(myDetails);

        //removing the mapping with the key and value
        myDetails.remove("Surname", "Somanna");
        System.out.println();

        System.out.println(myDetails);
    }
}
```

Output:



```
Command Prompt

$java MapIntro.java
{FirstName=Somanna, DOB=20/02/1947, Phone number=8965474562, Gender=Male, Surname=Somanna, Password=##90$$}

{FirstName=Somanna, DOB=20/02/1947, Phone number=8965474562, Gender=Male, Password=##90$$}
```

Here, remove(key, value) method will remove Surname and Somanna key-value pair in the HashMap.