Java HashMap.computeIfAbsent() – Syntax & Examples

Java HashMap.computeIfAbsent() – Examples

In this tutorial, we will learn about the Java HashMap.computeIfAbsent() function, and learn how to use this function to compute a value for given key only if this key is not present in the HashMap, with the help of examples.

computeIfAbsent()

HashMap.computeIfAbsent() computes the value for given key, using the given mapping function and puts this entry into this HashMap. computeIfAbsent() does this only if the specified key is not already present in the HashMap.

If the key is already present, computeIfAbsent() does nothing.

Syntax

The syntax of computeIfAbsent() function is

```
computeIfAbsent(K key, Function<? super K,? extends V> mappingFunction)
```

where

Parameter	Description
key	The key with which the specified value is to be associated.
mappingFunction	The mapping function to compute a value for the specified key.

Returns

The function returns value.

Example 1 – computeIfAbsent() – Key is Absent

In this example, we will initialize a HashMap with mappings from String to Integer. Using computeIfAbsent() function, we will compute a value for key `"M"`. Since the key `"M"` is not already present in the HashMap, this new mapping will be inserted to HashMap.

Java Program

```
import java.util.HashMap;

public class Example{
    public static void main(String[] args) {
        HashMap<String,Integer> hashMap = new HashMap<>();
        hashMap.put("A",1);
        hashMap.put("B",2);
        hashMap.put("C",3);
        hashMap.put("D",4);
        System.out.println("Before compute : " + hashMap);

        hashMap.computeIfAbsent("M", k -> 0 );
        System.out.println("After  compute : " + hashMap);
    }
}
```

Output

```
Before compute : {A=1, B=2, C=3, D=4}
After  compute : {A=1, B=2, C=3, D=4, M=0}
```

Example 2 – computeIfAbsent() – Key is Present Already

In this example, we will initialize a HashMap with mappings from String to Integer. Using computeIfAbsent() function, we will compute a value for key `"C"`. Since the key `"C"` is already present in the HashMap, computeIfAbsent() does nothing, and the HashMap is unaffected.

Java Program

```
import java.util.HashMap;

public class Example{
    public static void main(String[] args) {
        HashMap<String,Integer> hashMap = new HashMap<>();
        hashMap.put("A",1);
        hashMap.put("B",2);
        hashMap.put("C",3);
        hashMap.put("D",4);
        System.out.println("Before compute : " + hashMap);

        hashMap.computeIfAbsent("C", k -> 0 );
        System.out.println("After  compute : " + hashMap);
    }
}
```

```
}
```

Output

```
Before compute : {A=1, B=2, C=3, D=4}
After  compute : {A=1, B=2, C=3, D=4}
```

Example 3 – computeIfAbsent() – Return Value

In this example, we will store the value returned by computeIfAbsent(), and print to console.

The datatype of return value is the datatype of values of mappings in the HashMap. In this example, the datatype of the the return value is Integer.

Java Program

```
import java.util.HashMap;

public class Example{
    public static void main(String[] args) {
        HashMap<String,Integer> hashMap = new HashMap<>();
        hashMap.put("A",1);
        hashMap.put("B",2);
        hashMap.put("C",3);
        hashMap.put("D",4);

        int value = hashMap.computeIfAbsent("M", k->8);
        System.out.println("Return Value : " + value);
    }
}
```

Output

```
Return Value : 8
```

Example 4 – computeIfAbsent()

In this example, we will initialize a HashMap with mappings from String to Integer. Using computeIfAbsent() function, we will compute a value for key `"M"` and provide a `null` value for the **mappingFunction** parameter. computeIfAbsent() throws java.lang.NullPointerException.

Java Program

```
import java.util.HashMap;

public class Example{
    public static void main(String[] args) {
        HashMap<String,Integer> hashMap = new HashMap<>();
```

```
        hashMap.put("A",1);
        hashMap.put("B",2);
        hashMap.put("C",3);
        hashMap.put("D",4);
        System.out.println("Before compute : " + hashMap);

        hashMap.computeIfAbsent("M", null);
        System.out.println("After  compute : " + hashMap);
    }
}
```

Output

```
Before compute : {A=1, B=2, C=3, D=4}
Exception in thread "main" java.lang.NullPointerException
    at java.base/java.util.HashMap.computeIfAbsent(Unknown Source)
    at Example.main(Example.java:12)
```

Conclusion

In this Java Tutorial, we have learnt the syntax of Java HashMap.computeIfAbsent() function, and also learnt how to use this function with the help of examples.

- Java String
- Java Exception Handling
- Java File Operations
- Java Date & Time
- Java MySQL
- Java Random
- Java Math