Here are the **newly added methods in Java 8’s Map interface** (java.util.Map):

| **Method** | **Description** |
| --- | --- |
| forEach(BiConsumer<K,V>) | Performs action for each entry |
| getOrDefault(Object key, V defaultValue) | Returns value or default if key not found |
| putIfAbsent(K key, V value) | Adds entry if key is absent |
| remove(Object key, Object value) | Removes entry only if key maps to value |
| replace(K key, V value) | Replaces value for key if present |
| replace(K key, V oldValue, V newValue) | Replaces value only if currently mapped to oldValue |
| replaceAll(BiFunction<K,V,V>) | Replaces all values using function |
| compute(K key, BiFunction<K,V,V>) | Computes new value for key |
| computeIfAbsent(K key, Function<K,V>) | Computes value if key is absent |
| computeIfPresent(K key, BiFunction<K,V,V>) | Computes value if key is present |
| merge(K key, V value, BiFunction<V,V,V>) | Merges value with existing value for key |

**Example:**

map.putIfAbsent("A", 1);

map.computeIfAbsent("B", k -> 2);

map.forEach((k, v) -> System.out.println(k + ":" + v));

These methods make working with maps more functional and concise in Java 8.