

# Java Home Assignment - Map Merger

Merging two maps is not a trivial task. Consider the following example:

```
Map<String, Integer> map1 = new HashMap<>();
Map<String, Integer> map2 = new HashMap<>();
map1.put("key1", 20);
map1.put("key2", 30);
map2.put("key3", 40);
map2.put("key1", 50);
```

the result of: `map1.putAll(map2)` will result with one value for "key1" overriding the other value.

Your task is to build a **generalized** map-merging method.

The method you create should accept two maps and "merging" behavior.

For example, you should be able to use the method to merge the above maps with summing behavior (i.e., key1 will hold 70 as a result).

You could also choose to use multiplying behavior (key1 will hold 1000).

If the map is of type: `Map<Integer, String>` the caller may choose String concatenation as the merging behavior.

## Implementation notes

- 1) Please make good use of generics (i.e., no hard-coded types like Integer or String).
- 2) It should be efficient (consider cases when one map is much bigger than the other one).
- 3) No third-parties for this task (only standard library).
- 4) Java8 lambdas are nice to have, but not a requirement.

## Bonus part

If we can now merge two maps, how about merging a list of maps. And, doing it in parallel. Design and implement a method to merge a list of maps (you can use Java8 streams API).

Note, consider carefully, what can be parallelized and in which conditions!