

Nested Maps Solutions

- Write a simple program that creates a map instance, initializes it using a list initializer and prints out the elements
- Write a `using` declaration that creates a type alias for your map. Check that your program works with this type alias
- Create a nested map that uses this type for its value
- Write a program that populates this nested map with some elements and prints out their values

- Create a level map whose initial values are key 1, value "player" and key 10, value "door"
- Create another level map whose initial values are key 5, value "player" and key 10, value "monster"
- Create a nested game map whose values are the level maps you created above. Give the first level map a key of 1 and the second level map a key of 2

- Write a program which iterates over the game map and prints out all the key-value pairs for the level maps
- Amend your program so it also prints out the keys for the game map

- Update the game map by adding an element with pair value {3, "magic wand"} to the second level map you created
- Create another level map which has the initial pair values {7, "player"} and {8, "bomb"}. Add it to the game map with a key of 3
- Print out all the key-value pairs again

- Remove the level map with key 1 from the game map
- Update the game map by removing the element with key 10 from the level map which has key 2 in the game map
- Print out all the key-value pairs again

- The code below does not work as expected. Explain what the problem is and how it could be fixed

```
auto level = game_map.find(2);  
  
if (level != game_map.end()) {  
    auto lmap = level->second;  
    auto ten = lmap.find(10);  
    if (ten != lmap.end())  
        lmap.erase(ten);  
}
```

- The code below does not work as expected. Explain what the problem is and how it could be fixed
 - `lmap` is a copy of the second level
 - The `erase` call deletes the element with key 10 from this copy, but not the original
 - This could be fixed by using a reference instead
`auto& lmap = level->second;`